Climbing and Walking Robots

Manuel Armada Pablo González de Santos

Climbing and Walking Robots

Proceedings of the 7th International Conference CLAWAR 2004

Springer

Dr. Manuel A. Armada
Industrial Automation Institute – CSIC
Carretera de Campo Real, Km. 0, 200
28500 La Poveda (Madrid), Spain
E-mail: armada@iai.csic.es

Dr. Pablo González de Santos
Industrial Automation Institute – CSIC
Carretera de Campo Real, Km. 0, 200
28500 La Poveda (Madrid), Spain
E-mail: pgds@iai.csic.es

Library of Congress Control Number: 2005927314

ISBN-10 3-540-22992-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-22992-6 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springeronline.com
© Springer-Verlag Berlin Heidelberg 2005
Printed in The Netherlands

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: by the authors and TechBooks using a Springer LaTeX macro package

Cover design: *design & production* GmbH, Heidelberg

Printed on acid-free paper SPIN: 10972984 89/TechBooks 5 4 3 2 1 0

Conference Committees

ORGANISING COMMITTEE

General Chair

Manuel Armada

Co-Chairs

Yvan Baudoin — Gurvinder S. Virk
Karsten Berns — Philippe Bidaud
Giovanni Muscato

International Organising Committee
Chair: Pablo González de Santos

G Abba
T Akinfiev
J R Alique
A T de Almeida
R Aracil
C Balaguer
A Baños
J Billingsley
G Bolmsjo
M Buehler
R Caballero
D G Caldwell
R Carelli
C Chevallerau
H Cruse
X Dai
R Dillmann
B Espiau
G. Fernández
J G Fontaine
L Fortuna
KV Frolov
T Fukuda
M Garcia-Alegre
E Garcia
Z Gong
V Gradetsky
J C Grieco
A Halme
N Heyes
S Hirose
D Howard
O Kaynak
R Kelly
H Kimura
P Kiriazov
P Kopacek
K R Kozlowski
V Larin
D Lefeber
J Lopez-Coronado
O Matsumoto
M Maza
R Molfino
L Montano
N K M'Sirdi
A Nishi
J H Oh
D E Okhotsimsky
F Pfeiffer
A Preumont
E von Puttkamer
R Quinn
M Ribeiro
M A Salichs
V Sanchez
J Santos Victor
A Semerano
P Skelton
L Steinicke
S Tachi
K Tanie
T J Tarn
M O Tokhi
S Tzafestas
A Vitko
T Wadden
K J Waldron
R Walker
H A Warren
T White
W Yan
S Ylönen
M Xie

National Organising Committee
Chair: Salvador Ros

J. A. Cobano	T. Guardabrazo	M. Prieto
J. Estremera	J. L. López	L. Pedraza
R. Fernández	C. Manzano	A. Ramırez
V. Garcıa	H. Montes	C. Salinas
O. Garcıa	S. Nabulsi	J. Sarria

Proceedings of the
Seventh International Conference on

Climbing and Walking Robots

CLAWAR 2004

Consejo Superior de Investigaciones Científicas,
Madrid, Spain

22–44 September 2004

Edited by

Dr M. A. Armada

and

Dr P. González de Santos

Instituto de Automatica Industrial
Consejo Superior de Investigaciones Científicas
Madrid, Spain

Springer

About the Editors

Manuel Armada received his Ph. D. in Physics from the University of Valladolid (Spain) in 1979. He has been involved since 1976 in research activities related to Automatic Control (singular perturbations and aggregation, bilinear systems, adaptive and non-linear control, multivariable systems in the frequency domain, and digital control) and Robotics (kinematics, dynamics, tele-operation). He has been working in more than forty RTD projects (including international ones like EUREKA, ESPRIT, BRITE/EURAM, GROWTH, and others abroad the EU, especially with Latin America (CYTED) and Russia. Dr. Armada owns several patents, and has published over 200 papers (including contributions to several books, monographs, journals, international congresses and workshops). He is currently the Head of the Automatic Control Department at the Instituto de Automatica Industrial (IAI-CSIC), and Member of Russian Academy of Natural Sciences, being his main research direction concentrated in robot design and control, with especial emphasis in new fields like flexible robots and on walking and climbing machines.

Pablo González de Santos is a research scientist at the Spanish Council for Scientific Research (CSIC). He received his Ph.D. degree from the University of Valladolid (Spain). Since 1981 he has been involved actively in the design and development of industrial robots and also in special robotic systems. His work during last ten years has been focused on walking machines. He worked on the AMBLER project as a visiting scientist at the Robotics Institute of Carnegie Mellon University. Since then, he has been leading the development of several walking robots such as the RIMHO robot, designed for intervention on hazardous environments, the ROWER walking machine developed for welding tasks in ship erection processes and the SILO4 walking robot intended for educational and basic research purposes. He has also participated in the development of other legged robot such as the REST climbing robot and the TRACMINER. Dr. González de Santos is now studying how to apply walking machines to the field of humanitarian demining and has designed the new SILO6.

Preface

These Proceedings are a collection of selected papers presented at the Seventh International Conference organised by the EC GROWTH Thematic Network on Climbing and Walking Robots (CLAWAR). This conference, that was held in Madrid (September 2004), builds upon the highly successful previous Conferences held in Brussels (November 1998), Portsmouth (September 1999), Madrid (October 2000), Karlsruhe (September 2001), Paris (September 2002), and Catania (September 2003), and precedes the next CALWAR'05 to be held in London.

The interest in climbing and walking robots has remarkably augmented over recent years. Novel solutions for complex and very diverse application fields (exploration/intervention in severe environments, personal services, emergency rescue operations, transportation, entertainment, medical, etc.), has been anticipated by means of a large progress in this area of robotics. Moreover, the amalgamation of original ideas and related innovations, the search for new potential applications and the use of state of the art supporting technologies permit to foresee an important step forward and a significant socio-economic impact of advanced robot technology in the forthcoming years. In response to the technical challenges in the development of these sophisticated machines, a significant research and development effort has yet to be undertaken. It concerns embedded technologies (for power sources, actuators, sensors, information systems), new design methods, adapted control techniques for highly redundant systems, as well as operational and decisional autonomy and human/robot co-existence.

The European Commission is funding the CLAWAR Network. However, our aim as scientists and industrialists in this exciting field of robotics, is not only to promote the knowledge and applications of the complex mechatronic devices under development inside the EU and to show how them can contribute to a competitive and sustainable growth in our countries, but to disseminate our technology outside Europe and to be very receptive of what is being done all around the world. Confirming this situation greatest awareness has been received to CLAWAR'04, and after a careful reviewing procedure the

conference finally accommodate 115 papers of high quality, where the number of authors goes over 270. Papers and Members of International Organising Committee account 26 countries, pointing out the high level of international activity in this field, that is continuously growing and where many new research groups are being set up throughout the world.

As a summary of the conference, it can be said that included both state of the art and more practical presentations dealing with implementation problems, support technologies and future applications. A growing interest in passive locomotion is reflected by a very interesting number of contributions, collected at the special session organised by Prof. Martijn Wisse (Delft University of Technology). Some outstanding new climbing and walking robots were also included, and the conference was going together with a robot exhibition and the traditional student climbing robot competition. Best Paper Award was offered by Emerald: Industrial Robot: An international Journal, and special issues of some of the foremost scientific journals on robotics will be published after the conference. Of major relevance is the commitment with our new editors, Springer Verlag, that are in charge of this Conference Proceedings edition.

In addition to the submitted papers, several keynote speakers made plenary presentations. These included: Professor R. Dillmann (Fakultät für Informatik, Institut für Prozessrechentechnik und Robotik, University of Karlsruhe, Karlsruhe, Germany) on *Biologically motivated control of walking machines*; Prof. R. McNeil Alexander (The Faculty of Biological Sciences, University of Leeds, Leeds, UK) on *Problems of scale for walking and climbing animals*; Prof. D. A. Winter (Department of Kinesiology, University of Waterloo, Waterloo, Ontario, Canada) on *What bipedal human locomotion can teach us about motor control synergies for safe robotic locomotion;* Prof. A. L. Ruina (The Cornell BioRobotics and Locomotion Lab, Cornell University, Ithaca, New York, USA) on *Some mechanics perspectives on robot locomotion;* and, Prof. M. Xie (Nanyang Technological University School of Mechanical & Production Engineering Singapore) on *Robot vision: a holistic view.*

We would like to take this opportunity to thank all those involved in organising CLAWAR'04. To the Co-Chairmen (Yvan Baudoin, Gurvinder Singh Virk, Karsten Berns, Philippe Bidaud, Giovanni Muscato), to the Plenary Speakers, to the International Organising Committee, to the National Organising Committee, Chaired by Dr. Salvador Ros, Director of the Industrial Automation Institute, and to the Consejo Superior de Investigaciones Científicas, who host the Conference, our acknowledgement for their invaluable help and kind assistance. Special thanks are for Miss Anna Kochan and Dr. Clive Loughlin from Industrial Robot, that apart from offering the Award joined us at the conference. CLAWAR Network partners have been very supportive with their extensive work in the backstage, promoting, widening and rationalising the CLAWAR technology, filling existing gaps, defining new concepts

and expanding the applications horizon of climbing and walking robots. Their work has been fundamental for the preparation of this 7th Conference.

Particularly, thanks are extensive to the IAI-CSIC colleagues, to its Technical and Administrative Staff and to the members of the Automatic Control Department with special mention to: Teodor Akinfiev, Samir Nabulsi, who was "riding" ROBOCLIMBER outdoor CSIC premises, Hector Montes, Javier Sarria, Roberto Ponticelli, Manuel Prieto, Elena García, Joaquín Estremera, Luis Pedraza, Tomás Guardabrazo, Octavio García, Carlota Salinas, Mauricio Alba, Alvaro Lou, Gabriel Bacallado, and Jorge López, because without their invaluable assistance CLAWAR'04 would never been a sound reality.

June, 2005

Manuel A. Armada
Pablo González de Santos

Contents

Part I Plenary Sessions

Robot Vision: A Holistic View
Ming Xie 3

What Bipedal Human Locomotion Can Teach Us About Motor Control Synergies for Safe Robotic Locomotion
D.A. Winter 39

Problems of Scale for Walking and Climbing Animals
R. McNeill Alexander 47

Biologically Motivated Control of Walking Machines
R. Dillmann, J. Albiez, B. Gaßmann, T. Kerscher 55

Part II Control

Integer vs. Fractional Order Control of a Hexapod Robot
Manuel F. Silva, J. A. Tenreiro Machado, António M. Lopes 73

Synchronous Landing Control of a Rotating 4-Legged Robot, PEOPLER, for Stable Direction Change
T. Okada, Y. Hirokawa, T. Sakai, K. Shibuya 85

Neuro-Controllers for Walking Machines – An Evolutionary Approach to Robust Behavior
Joern Fischer, Frank Pasemann, Poramate Manoonpong 97

Decentralized Dynamic Force Distribution for Multi-legged Locomotion
T. Odashima, Z.W. Luo 103

An Outdoor Vehicle Control Method Based Body Configuration Information
Daisuke Chugo, Kuniaki Kawabata, Hayato Kaetsu, Hajime Asama, Taketoshi Mishima 113

Physically Variable Compliance in Running
Jonathan W. Hurst, Alfred A. Rizzi 123

Implementation of a Driver Level with Odometry for the LAURON III Hexapod Robot
J.L. Albarral, E. Celaya 135

Local Positive Velocity Feedback (LPVF): Generating Compliant Motions in a Multi-Joint Limb
Axel Schneider, Holk Cruse, Josef Schmitz 143

Motion Calculation for Human Lower Extremities Based on EMG-Signal-Processing and Simple Biomechanical Model
Christian Fleischer, Konstantin Kondak, Christian Reinicke, Günter Hommel 153

Bifurcating Recursive Processing Elements in Neural Architectures for Applications in Multi-Dimensional Motor Control and Sensory Fusion in Noisy/Uncertain Environments
E. Del Moral, E. Akira 163

The Effectiveness of Energy Conversion Elements in the Control of Powered Orthoses
S.C. Gharooni, M.O. Tokhi, G.S. Virk 171

Kinematical Behavior Analysis and Walking Pattern Generation of a Five Degrees of Freedom Pneumatic Robotic Leg
G. Muscato, G. Spampinato 181

Artificial Potential Based Control for a Large Scale Formation of Mobile Robots
Wojciech Kowalczyk, Krzysztof Kozlowski 191

Mobile Mini Robots for MAS
M.W. Han, P. Kopacek, B. Putz, E. Sshierer, M. Würzl 201

Two Neural Approaches for Solving Reaching Tasks with Redundant Robots
J. Molina-Vilaplana, J.L. Pedreño-Molina, J. López-Coronado 211

Design and Implementation of Force Sensor for ROBOCLIMBER
H. Montes, S. Nabulsi, M. Armada 219

Detecting Zero-Moment Point in Legged Robot
H. Montes, S. Nabulsi, M. Armada 229

Vision Feedback in Control of a Group of Mobile Robots
Piotr Dutkiewicz, Marcin Kiełczewski 237

Part III Design

Kinematics of a New Staircase Climbing Wheelchair
R. Morales, V. Feliu, A. González, P. Pintado 249

Open Modular Design for Robotic Systems
I. Chochlidakis, Y. Gatsoulis, G.S. Virk 265

Mechanical Design Optimization of a Walking Robot Leg Using Genetic Algorithm
C. Reyes, F. Gonzalez 275

Design Toolset for Realising Robotic Systems
Yiannis Gatsoulis, Ioannis Chochlidakis, Gurvinder S. Virk 285

Design, Dynamic Simulation and Experimental Tests of Leg Mechanism and Driving System for a Hexapod Walking Robot
J. Roca, M. Nogués, S. Cardona 295

Limb-Mechanism Robot with Winch Mechanism
N. Fujiki, Y. Mae, T. Umetani, T. Arai, T. Takubo, K. Inoue 305

Embodiment in Two Dimensions
Christian R. Linder 313

Legged Robot with Articulated Body in Locomotion Over Complex Terrain
Palis, Rusin, Schumucker, Schneider, Zavgorodniy 321

WALKIE 6.4: A New Improved Version of a Rigid Frames Hexapod Rover
N. Amati, B. Bona, S. Carabelli, M. Chiaberge, G. Genta, M. Padovani, R. Volpe 329

WallWalker: Proposal of Locomotion Mechanism Cleaning Even at the Corner
T. Miyake, H. Ishihara 341

Behaviour Networks for Walking Machines – A Design Method
Jan Albiez, Rudiger Dillmann 349

Biological Inspired Walking – How Much Nature Do We Need?
Jan Albiez, Karstan Berns 357

Part IV Sensors, Teleoperation and Telepresence

Results of Applying Sensor Fusion to a Control System Using Optic Flow
G. Martinez, V.M. Becerra 367

Novel Method for Virtual Image Generation for Teleoperation
R. Chellali, C. Maaoui, J.G. Fontaine 377

Intelligent Technical Audition and Vision Sensors for Walking Robot Realizing Telepresence Functions
V.E. Pavlovsky, S.A. Polivtseev, T.S. Khashan 387

Learning About the Environment by Analyzing Acoustic Information. How to Achieve Predictability in Unknown Environments? (Part II)
M. Deutscher, M. Katz, S. Krüger, M. Bajbouj 399

Ultrasound Sensor System with Fuzzy Data Processing
J.A. Morgado de Gois, M. Hiller 411

Finding Odours Across Large Search Spaces: A Particle Swarm-Based Approach
Lino Marques, A.T. de Almeida 419

Vision Feedback in Control of a Group of Mobile Robots
Piotr Dutkiewicz, Marcin Kiełczewski 427

Vehicle Teleoperation with a Multisensory Driving Interface
Mario Maza, Santiago Baselga, Jesús Ortiz 437

Approaches to the Generation of Whole Body Motion Sensation in Teleoperation
Mario Maza, Santiago Baselga, Jesús Ortiz 447

Virtual Platform for Land-Mine Detection Based on Walking Robots
A. Ramirez, E. Garcia, P. Gonzalez de Santos 457

Vision Computer Tool to Improve the Dependability of Mobile Robots for Human Environments
C. Salinas, L. Pedraza, M. Armada 467

Part V Efficiency and Actuation

Toward Springy Robot Walk Using Strand-Muscle Actuators
Masakazu Suzuki, Azumi Ichikawa 479

Actuator Sizes in Bio-Robotic Walking Orthoses
S.C. Gharooni, G.S. Virk, M.O. Tokhi 487

The Design and Simulated Performance of an Energy Efficient Hydraulic Legged Robot
Salim Al-Kharusi, David Howard 495

The Modularity of Super Embedded Robotic PC
A. Basile, N. Abbate, C. Guastella, M. Lo Presti, G. Macina 503

Mass Distribution Influence on Power Consumption in Walking Robots
T.A. Guardabrazo, P. Gonzalez de Santos 511

Design of Dual Actuator for Walking Robots
Teodor Akinfiev, Roemi Fernandez, Manuel Armada 519

Part VI Hopping, Biped and Humanoid Robots

Control of a 3-D Hopping Apparatus
V.B. Larin 531

Learning of the Dynamic Walk of an Under-Actuated Bipedal Robot: Improvement of the Robustness by Using CMAC Neural Networks
Christophe Sabourin, Olivier Bruneau, Jean-Guy Fontaine 543

Dynamic Stabilization of an Under-Actuated Robot Using Inertia of the Transfer Leg
A. David, O. Bruneau, J.G. Fontaine 551

Kinematic and Dynamic Analyses of a Pantograph-Leg for a Biped Walking Machine
E. Ottaviano, M. Ceccarelli, C. Tavolieri 561

Humanoid Robot Kinematics Modeling Using Lie Groups
J.M. Pardos, C. Balaguer 569

GA Optimisation of the PD Coefficients for the LMBC of a Planar Biped
D. Harvey, G.S. Virk, D. Azzi 577

Three-Dimensional Running is Unstable but Easily Stabilized
Justin E. Seipel, Philip J. Holmes 585

A Biomimetic Approach for the Stability of Biped Robots
Javier de Lope, Darío Maravall 593

Parallel Manipulator Hip Joint for a Bipedal Robot
J. Hofschulte, M. Seebode, W. Gerth 601

Gaits Stabilization for Planar Biped Robots Using Energetic Regulation
N.K. M'Sirdi, N. Khraief, O. Licer 611

Force Feedback Control Implementation for SMART Non-Linear Actuator
H. Montes, L. Pedraza, M. Armada, T. Akinfiev 625

User Friendly Graphical Environment for Gait Optimization of the Humanoid Robot Rh-0
M. Arbulú, I. Prieto, D. Gutiérrez, L. Cabas, P. Staroverov, C. Balaguer . . 633

Development of the Light-Weight Human Size Humanoid Robot RH-0
L. Cabas, S. de Torre, M. Arbulu, C. Balaguer 643

Human Machine Interface for Humanoid Robot RH-0
I. Prieto, C. Pérez, C. Balaguer 655

Trajectory Planning for the Walking Biped "Lucy"
Jimmy Vermeulen, Dirk Lefeber, Björn Verrelst, Bram Vanderborght . . . 665

Height Control of a Resonance Hopping Robot
Roemi Fernández, Teodor Akinfiev, Manuel Armada 677

Zero Moment Point Modeling Using Harmonic Balance
R. Caballero, M. Armada 689

An Introductory Revision to Humanoid Robot Hands
D. Alba, M. Armada, R. Ponticelli 701

Control Architecture of LUCY, a Biped with Pneumatic Artificial Muscles
B. Vanderborght, B. Verrelst, R. Van Ham, J. Vermeulen, J. Naudet, D. Lefeber 713

Part VII Passive Walking Locomotion

Stable Walking and Running Robots Without Feedback
Katja D. Mombaur, H. Georg Bock, Johannes P. Schlöder, Richard W. Longman 725

From Passive to Active Dynamic 3D Bipedal Walking – An Evolutionary Approach
Steffen Wischmann, Frank Pasemann 737

First Steps in Passive Dynamic Walking
Martijn Wisse, Arend L. Schwab 745

Controlling Walking Period of a Pneumatic Muscle Walker
Takashi Takuma, Koh Hosoda, Masaki Ogino, Minoru Asada 757

Evolutionary Design of an Adaptive Dynamic Walker
Joachim Haß, J. Michael Herrmann, Theo Geisel 765

The Passivity Paradigm in the Control of Bipedal Robots
Mark W. Spong 775

Ankle Joints and Flat Feet in Dynamic Walking
Daan G.E. Hobbelen, Martijn Wisse 787

Robotic Walking Aids for Disabled Persons
G.S. Virk, S.K. Bag, S.C. Gharooni, M.O. Tokhi, R.I. Tylor, S. Bradshaw, F. Jamil, I.D. Swain, P.H. Chapple, R.A. Allen 801

The Tango of a Load Balancing Biped
Eric D. Vaughan, Ezequiel Di Paolo, Inman R. Harvey 813

Locomotion Modes of an Hybrid Wheel-Legged Robot
G. Besseron, Ch. Grand, F. Ben Amar, F. Plumet, Ph. Bidaud 825

Stabilizing Dynamic Walking with Physical Tricks
Norbert M. Mayer, Amir A. Forough-Nassiraei, Ziton Hsu, Ferenc Farkas, Thomas Christaller 835

Stability of a Spherical Pendulum Walker
J. Seipel 843

A CLAWAR That Benefits From Abstracted Cockroach Locomotion Principles
T.E. Wei, R.D. Quinn, R.E. Ritzmann 849

***iSprawl*: Autonomy, and the Effects of Power Transmission**
Sangbae Kim, Jonathan E. Clark, Mark R. Cutkosky 859

Locomotion of a Modular Worm-like Robot Using a FPGA-based Embedded MicroBlaze Soft-processor
J. Gonzalez-Gomez, E. Aguayo, E. Boemo 869

Evolutionary Synthesis of Structure and Control for Locomotion Systems
O. Chocron, N. Brener, Ph. Bidaud, F. B. Amar 879

Kinematic Model and Absolute Gait Simulation of a Six-Legged Walking Robot
G. Figliolini, V. Ripa 889

Part VIII Climbing and Navigation

Simulation of Climbing Robots Using Underpressure for Adhesion
C. Hillenbrand, J. Wettach, K. Berns 899

Technique for a Six-Legged Walker Climbing a High Shelf by Using a Vertical Column
Yu. F. Golubev, V.V. Korianov 909

Inverse Kinematic/Dynamic Analysis of a New 4-DOF Hybrid (Serial-Parallel) Pole Climbing Robot Manipulator
M.R. Zakerzadeh, M. Tavakoli, G.R. Vossoughi, S. Bagheri 919

Climbing Without a Vacuum Pump
Werner Brockmann, Florian Mösch 935

ROBOCLIMBER: Control System Architecture
S. Nabulsi, H. Montes, M. Armada 943

Navigation of Walking Robots: Localisation by Odometry
B. Gaßmann, J.M. Zöllner, R. Dillmann 953

Towards Penetration-based Clawed Climbing
William R. Provancher, Jonathan E. Clark, Bill Geisler, Mark R. Cutkosky 961

Motion Planning for a Legged Vehicle Based on Optical Sensor Information
Richard Bade, André Herms, Thomas Ihme 971

Developing Climbing Robots for Education
K. Berns, T. Braun, C. Hillenbrand, T. Luksch 981

Robust Localisation System for a Climbing Robot
André Martins, Lino Marques, A.T. de Almeida 989

Roboclimber: Proposal for Online Gait Planning
M. Moronti, M. Sanguineti, M. Zoppi, R. Molfino 997

Adhesion Control for the Alicia^3 Climbing Robot
D. Longo, G. Muscato 1005

Part IX Applications

In Service Inspection Robotized Tool for Tanks Filled with Hazardous Liquids – Robtank Inspec
A. Correia Cruz, M. Silva Ribeiro 1019

SIRIUSc – Facade Cleaning Robot for a High-Rise Building in Munich, Germany
N. Elkmann, D. Kunst, T. Krueger, M. Lucke, T. Böhme, T. Felsch, T. Stürze 1033

In-Pipe Microrobot with Inertial Mood of Motion
G. G. Rizzoto, M. Velkenko, P. Amato, V. Gradetsky, S. Babkirov, M. Knyazkov, V. Solovtov 1041

The Layer-Crossing Strategy of Curved Wall Cleaning Robot
R. Liu, J. Heng, G. Zong 1053

Pneumatic Climbing Robots for Glass Wall Cleaning
Houxiang Zhang, Jianwei Zhang, Rong Liu, Wei Wang, Guanghua Zong 1061

Design and Prototyping of a Hybrid Pole Climbing and Manipulating Robot with Minimum DOFs for Construction and Service Applications
M. Tavakoli, M.R. Zakerzadeh, G.R. Vossoughi, S. Bagheri 1071

Part X Innovative Systems

Design and Control of a Manipulator for Landmine Detection
E. Garcia, P. Gonzalez de Santos 1083

Interactions Between Human and Robot – Case Study: WorkPartner-Robot in the ISR 2004 Exhibition
S. Ylönen, M. Heikkilä, P. Virekoski 1091

Robust Platform for Humanitarian Demining
Lino Marques, Svetlana Larionova, A.T. de Almeida 1097

Co-Operative Smell-Based Navigation for Mobile Robots
C. Lytridis, G.S. Virk, E.E. Kadar 1107

A Localization Algorithm for Outdoor Trajectory Tracking with Legged Robots
J.A. Cobano, J. Estremera, P. Gonzalez de Santos 1119

Sit to Stand Transfer Assisting by an Intelligent Walking-Aid
P. Médèric, V. Pasqui, F. Plumet, P. Bidaud 1127

Part XI CLAWAR Network Workpackages

CLAWAR Modularity – Design Tools
G.S. Virk 1139

Interaction Space Analysis for CLAWAR WP5 Societal Needs
M. Armada, M. Prieto 1147

CLAWAR WP3 Applications: Natural/Outdoor and Underwater Robots
D. Longo, G. Muscato 1159

CLAWAR 2 Work Package 6 (WP6) – Assessment Year 2, May 2004. Economic Prospects for Mobile Robotic Systems – Exploitation and Risk
H.A. Warren, N.J. Heyes 1171

Index 1191

Part I

Plenary Sessions

Robot Vision: A Holistic View

Ming Xie

School of Mechanical & Aerospace Engineering, Nanyang Technological University, Singapore 639798
mmxie@ntu.edu.sg

Abstract. It is well understood that artificial vision enables a wide range of applications from visual inspection, visual measurement, visual recognition, visual surveillance, to visual guidance of robot systems in real-time and real environment. However, in the literature, there is no definite answer to what artificial (or robot) vision should be, or how it should be. This dilemma is largely due to the fact that artificial (or robot) vision is being actively pursued by scientists of various backgrounds in both social and natural sciences. In this paper, the intention is to present a new way of re-organizing various concepts, principles and algorithms of artificial vision. In particular, we propose a function-centric view comprising these five coherent categorizations, namely: (a) instrumental vision, (b) behavior-based vision, (c) reconstructive vision, (d) model-based vision, and (e) cognitive vision. This function-centric view abandons the long-standing notions of low-, intermediate- and high-level vision, as they are more illusive than insightful. The contribution of this article is two-fold. First, it is time to assess and consolidate the current achievements in artificial (or robot) vision. Secondly, it is important to objectively state the remaining challenges in order to guide the future investigations in artificial (or robot) vision.

Key words: Instrumental Vision, Behavior-based Vision, Reconstructive Vision, Model-based Vision, Cognitive Vision.

1 Introduction

Today's robot is an automated machine. The robot of tomorrow will certainly become an intelligent and autonomous creature of human being. In order to reach a certain level of intelligence and autonomy, the ability to perceive and understand external environment is indispensable. This is especially true for humanoid robots because they are supposed to perform useful tasks in a human environment. Among the different modalities of perception, vision is admittedly the most important one. In particular, vision is the only efficient way for an agent, such as a human or robot, to incrementally gain the visual awareness which is an important part of the consciousness [4].

Human vision is marvellously powerful. We are able to visually perceive and understand a complex scene both instantaneously and effortlessly. However, despite the study on human vision for decades, not much is understood about the inner processes of human vision. We know what human vision can do, but not how it does the complex task of visual perception. [1–3, 6–8] For instance, human vision is binocular in nature, but there is no common answer to how human vision solves the difficult problem of stereo correspondence in an effortless and perfect manner. [1] There is no doubt that human acquires internalized representation of the external world through vision. [5, 6] Yet, there is no definite answer to how human vision exactly transforms visual sensory signals into cognitive understanding and representation of meanings. [7, 8] Human vision is qualitative (i.e. not to do metric measurement in a precise manner) and cognitive (i.e. to derive meanings from images). But, there is no definite scientific answer to how human vision achieves the robustness through qualitative vision, and the understanding through cognitive vision. [2–5].

In this article, robot vision will be the focus of our discussions. Here, we distinguish vision (i.e. visual perception) from imaging (i.e. visual sensory system). And, we assume that the ultimate goal of robot vision is to enable a machine, such as a robot or computer, to see (i.e. to gain the visual awareness) for various purposes related to decision-making and action-taking. As robot vision falls into the category of invention-centric research, we are not restricted to follow the unknown principles (i.e. mental processes) of human vision. In fact, we do not know any definite answer yet to the question of how human vision works, despite the tremendous progress toward the understanding of the neural aspects of eyes and brain. [4, 5] Hence, we turn our attention toward the question of what the engineering principles and solutions underlying robot vision should be.

Before going into the detail of what robot vision should be, it is interesting to highlight that vision research was not originally intended to develop engineering principles and solutions for machines, such as computers or robots, to see. In fact, vision research was originated from the desire of understanding human vision as part of the science of the brain, which could be dated back a few centuries. Only the invention of digital computers and software in the middle of the last century has made it possible to capture, digitize, process and interpret images by computer (or artificial brain). Indeed, the advent of re-programmable computers has marked the beginning of the scientific investigation of how to make computer, or robot, to see. Most importantly, the advance of today's vision research has been benefited from various achievements made in very different fields such as psychology, cognitive science, neural science, neurophysiology, computer science, and engineering science. This fact implies that artificial (or robot) vision is multidisciplinary in nature.

Because of the multidisciplinary nature of robot vision, there is a diverging view with regard to what are really fundamentals for the constituent topics in robot vision. Therefore, we feel that it is timely now to adopt a holistic view on robot vision for the sake of unifying various methodologies into a

coherent and integrative platform. This thought has motivated us to abandon the illusive notions of low-, intermediate- and high-level vision. In contrast, we advocate a new function-centric view, which categorizes the various concepts, principles and algorithms of robot vision into five related, but distinct, areas, namely: (a) instrumental vision, (b) behavior-based vision, (c) reconstructive vision, (d) model-based vision, and (e) cognitive vision.

This paper is organized as follows: In Sect. 2, we briefly outline the concepts of visual scenes and objects. Section 3 details the fundamentals of instrumental vision. The basic principles of behavior-based vision are presented in Sect. 4. And, the main methodologies of reconstructive vision are discussed in Sect. 5. In Sect. 6, we detail the major aspects of model-based vision. And, future challenges of cognitive vision are highlighted in Sect. 7. Finally, we conclude this paper in Sect. 8.

2 Concepts of Visually Perceivable Scene and Object

Vision appears so natural to human beings. Because of this, it seems difficult to give a commonly acceptable answer to the question of what vision is all about. It is easy to say that humans (or animals) have vision because of the need of survival in finding foods, avoiding dangers, adapting to environment, and interacting in groups for intellectual, social and economic gains, etc. Indeed, vision do provide a sufficient condition for the survival. But, the real question is: what is the unique cause which makes vision a necessity? In other words, what makes vision a necessary condition which a physical agent or system must satisfy?

Without a clear and common answer to the question of what robot vision should be in order to effectively play the intended essential role as a necessity, we will face the trouble of losing the real focus of vision research. Here, we advocate the claim that the ultimate goal of vision is for an agent, such as human or robot, to incrementally gain the visual awareness of the external world for various purposes related to decision-making and action-taking. In engineering terms, it is to say that vision is a process for an agent to acquire the meanings of a scene through its images or videos, in order to be effective, autonomous, and even conscious. Hence, we propose to define vision as follows:

Definition 1. *Vision is a process, which derives the meanings of a scene through its images or videos.*

Usually, a scene will consist of entities called *objects*, which evolve and interact in space and time. This evolution and interaction in space and time will result in the concepts of *configuration*, *event* and *episode*. It is clear that a scene will have no meaning, if there is no object inside it. Therefore, the meanings of a scene depend on the meanings of its constituent objects. And, it is important to first define the meaning of an object.

In general, the meaning of an entity can be modelled by a combination of four elements in the conceptual and physical worlds. They are: (a) identity, (b) category, (c) property and (d) constraint. The first two elements are conceptual, while the last two are physical. If we follow this general idea about meaning, the meaning of an object can be stated as follows:

Definition 2. *The identity, categories, properties and constraints of an object constitute its meaning.*

We all know that vision relies on lights, optics and electronic devices to sense signals which carry the meaning of an object or a scene. Since lights are not able to transduce all properties and constraints into visual signals, it is natural to raise the question of what the visually-perceivable meaning of an object (or a scene) is. In other words, it is important to define the terms of *visual scene* and *visual object* in robot vision. Since a visual scene consists of visual objects, it is easy to define the term *visual scene*, as follows:

Definition 3. *A visual scene is a reality, which consists of visually-perceivable objects that evolve and interact in space and time.*

Now, we come to the question of how to define the term *visual object*. We know that light is a physical quantity in the physical world. Therefore, light only affects the perception of physical elements of a meaning. For instance, in terms of properties, lights allow the perception of appearance such as shape, color or texture. In terms of constraints, lights enable the perception of an object's structure and mechanism, which materialize the kinematic and dynamic constraints imposed to the object. Thus, a simple definition of visual objects can be stated as follows:

Definition 4. *A visual object is a reality, which consists of appearance, structure and mechanism, that could self-evolve or interact with other visual objects in space and time. A visual object carries its own identity and belongs to hierarchically-organized categories.*

In the following sections, we are going to detail the various aspects of robot vision under the new function-centric view.

3 Instrumental Vision

Instrumental vision studies the aspect of using robot vision to serve as an instrument of measurement. We know that human vision is not metric. For instance, no matter how long we perceive a new scene or object, we are not able to tell its exact geometry. In contrast, robot vision can serve as an instrument to measure the geometry of a scene or object in a relatively precise manner [9].

3.1 Measuring 2-D Geometry

Refer to Fig. 1. A 2-D plane in a 3-D space is related to the 2-D image-plane of a camera by the so-called *homography transform*, which is described as follows:

$$k \bullet \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \bullet \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{1}$$

where (u, v) are the index coordinates of a point in the 2-D image-plane, k a scalar, and (X, Y) the coordinates of a corresponding point in the 2-D plane. If the relative geometry between the two planes remains unchanged, the 3×3 matrix is constant, which can be determined by a simple calibration.

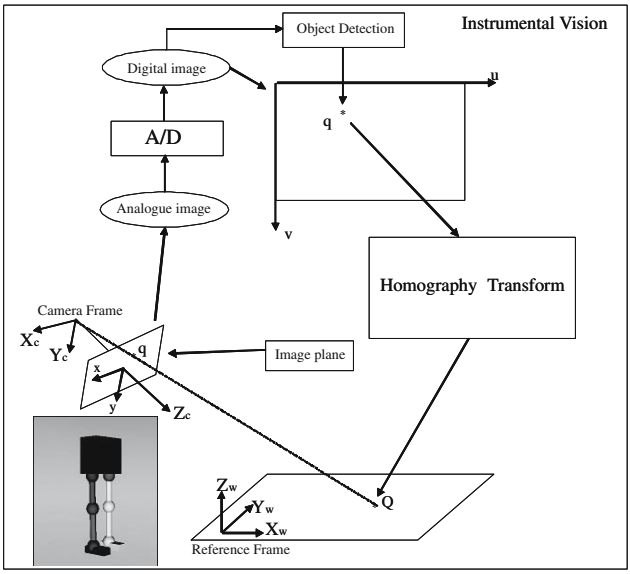


Fig. 1. An illustration of instrumental vision when measuring 2-D geometry. In this case, a 2-D plane in a 3-D space is geometrically related to the 2-D image-plane by a 3×3 matrix called homography transform. And, there is a one-to-one mapping between the coordinates in the 2-D plane and the index coordinates in the 2-D image-plane. As long as the geometry of an object such as the road can be detected in the image plane, its geometry in the 2-D plane can be computed by the homography transform

Equation (1) tells that the geometry of an object in a 2-D plane can be precisely measured from its image. One typical application of measuring 2-D geometry is the estimation of the geometry of a planar road-surface for autonomous walking or driving.

3.2 Measuring 3-D Geometry

A practical way of measuring the 3-D geometry of static objects is to use the so-called *3-D scanner*. There are different approaches to develop a 3-D scanner. One simple method is to apply the homography transform described in (1). A 3-D scanner based on the homography transform will consist of a camera, a laser projector, and a rotating table, as shown in Fig. 2.

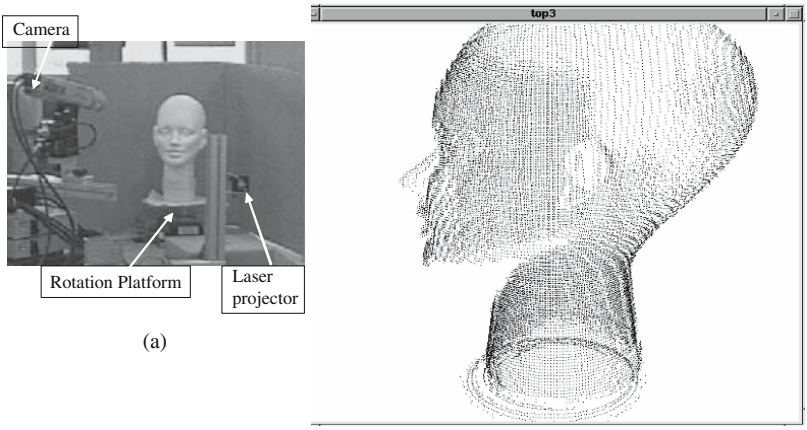


Fig. 2. An illustration of instrumental vision when measuring 3-D geometry. One can create a 2-D plane with a laser projector. When the camera and the laser projector are fixed together as shown in (**a**), the 2-D image-plane of the camera is related to the laser's emitting plane by a homography transform. When a 3-D object is placed on a rotating table, and is scanned through the laser's emitting plane, the geometries of the intersection lines at various time instants could be measured, and be put together to form the 3-D geometry of the object, as shown in (**b**)

When the camera and the laser projector are fixed on a same rigid structure, the relationship between the camera's image-plane and the laser projector's emitting-plane is governed by a homography transform (i.e. (1)). When the laser projector's emitting-plane intersects an object, it creates an intersection line. This intersection line can be detected from the image of the camera, and be measured by applying (1).

In order to measure the complete 3-D geometry of an object, the laser projector's emitting-plane must scan the surface of the object. This can be done in two ways: (a) to rotate the emitting plane while keeping the object stand still, and (b) to rotate the object while keeping the emitting plane stand still. After the full scan (i.e. the coverage of 360 degrees), the data are combined together to form the full description of the 3-D geometry of an object under scan.

3.3 Future Challenge

With the use of a single camera, one can measure the geometry in a 2-D plane. If a projector, which emits a specific light pattern (e.g. a plane), is used, one can measure the geometry of 3-D objects in a controlled setting. As human vision is binocular, we may raise this question of how to quantitatively infer 3-D geometry of a scene or object using binocular vision in real-time. This poses a big challenge, which is still actively pursued under the topic of reconstructive vision.

4 Behavior-based Vision

Behavior-based vision studies the aspect of using vision to provide sensory feedback to a robot's control system. We know that vision is an effective process to acquire signal, information and knowledge about a scene or object. And, the ultimate goal of sensing and understanding a scene or object is to enable decision-making and action-taking. In other words, one important purpose of vision is to guide actions. As vision is closely related to behaviors, it is crucial to understand the essence of behavior-based vision.

We have already mentioned that vision is able to output signal, information, and knowledge about a scene or object. However, at the level of signal, there are two interesting phenomena:

- First, the motion of a camera in an operational space could be manifested in the form of velocity, or displacement, vector field in the camera's image space. In the terminology of control engineering, this phenomenon means that the desired motion in an operational space could be mapped into the corresponding desired motion in an image space. In other words, we can purposefully control the camera's motion in order to achieve the desired outcome specified in an image space. This idea forms the basis of one type of visual servoing, which is called here "inner-loop" visual servoing.
- Secondly, if a binocular vision system is used, there is a simple correlation between a path specified in a task space and the corresponding paths in the image spaces. In the terminology of control engineering, this means that we could automate the process of planning the desired motion first, followed by motion execution. This idea forms the basis of another type of visual servoing, which is called here "outer-loop" visual servoing.

4.1 Inner-loop Visual Servoing

The control in the inner-loop visual servoing aims at minimizing an error function defined in the image space. As an image is the projection of a 3-D task space into a 2-D image space, the inner-loop visual servoing is a special case of task-space control in robotics.

As shown in Fig. 3, one way to construct an inner-loop visual servoing system is to mount a camera at the end-effector of a robot arm (or neck). The ultimate goal of inner-loop visual servoing is to regulate the joint motions (or joint torques) so as to bring the camera to, or maintain the camera at, a reference pose where the current image-features (e.g. a set of points or lines) equals the desired image-features [12].

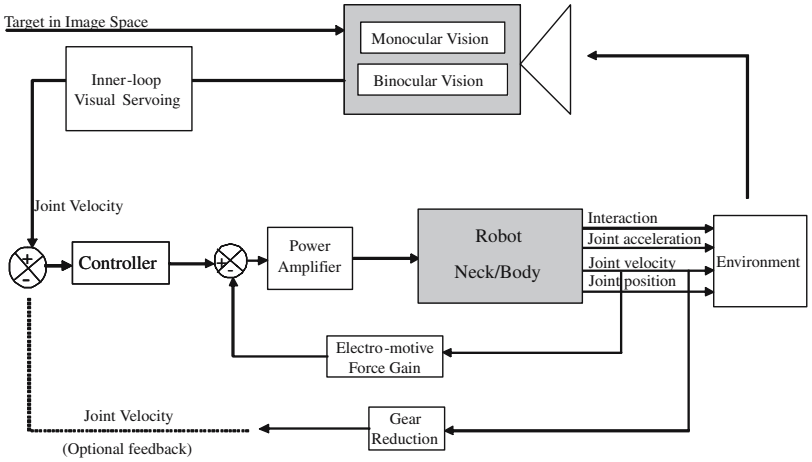


Fig. 3. Behavior-based vision using inner-loop visual servoing. The vision system is typically mounted on a neck/body mechanism. The observation of variations in image space is mapped to the joint velocity vector $\dot{q}$, which is the feedback to the robot's control system

If one chooses to explicitly regulate the joint torques in visual servoing, the complex system dynamics must be established, which relates the variation of image-features to the variation of joint torques. This approach will make the design of control laws and the stability analysis very difficult [14, 30].

A relatively simpler approach is to explicitly regulate the joint motions (e.g. joint velocities). In this way, the visual feedback loop serves the purpose of specifying the desired joint velocities for the optional velocity feedback loop, as shown in Fig. 3. In order to apprehend the challenge faced by approaches of inner-loop visual servoing, we briefly develop here the equation which relates the variation of image-features (e.g. a set of points) to the variation of joint variables (i.e. joint velocities).

Differential Model of Pin-hole Camera

Figure 4 shows the frame assignment in a typical setting of inner-loop visual servoing. If $({}^cX,\ {}^cY,\ {}^cZ)$ is the coordinates of an arbitrary point, denoted

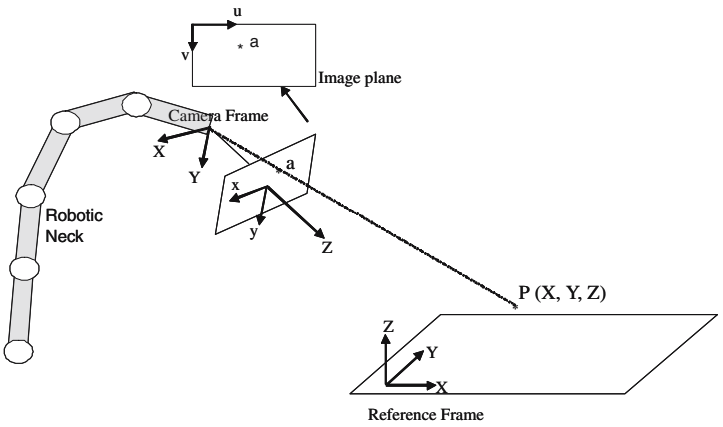


Fig. 4. Frame assignment in a typical setting of inner-loop visual servoing

by P, with respect to camera frame c, the index coordinates (u, v) of its corresponding image point will be:

$$\begin{cases} u = u_0 + f_x \frac{^cX}{^cZ} \\ \\ v = v_0 + f_y \frac{^cY}{^cZ} \end{cases} \tag{2}$$

where (u_0, v_0) are the coordinates of the image center, f_x the focal length scaled by the pixel size in X direction, and f_y the focal length scaled by the pixel size in Y direction.

The differentiation of (2), with respect to time, yields:

$$\begin{cases} \dot{u} = f_x \frac{^c\dot{X}}{^cZ} - f_x \frac{^cX}{^cZ^2} \, ^c\dot{Z} \\ \\ \dot{v} = f_y \frac{^c\dot{Y}}{^cZ} - f_y \frac{^cY}{^cZ^2} \, ^c\dot{Z} \, . \end{cases} \tag{3}$$

Equation (3) can be re-written in a matrix form, as follows:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = J_c \bullet \begin{pmatrix} ^c\dot{X} \\ ^c\dot{Y} \\ ^c\dot{Z} \end{pmatrix} \tag{4}$$

where

$$J_c = \begin{pmatrix} \frac{f_x}{^cZ} & 0 & -f_x \frac{^cX}{^cZ^2} \\ 0 & \frac{f_y}{^cZ} & -f_y \frac{^cY}{^cZ^2} \end{pmatrix} \, . \tag{5}$$

By using (2) to eliminate $(^cX, \, ^cY)$, (5) can be simplified to:

$$J_c = \begin{pmatrix} \frac{f_x}{^cZ} & 0 & \frac{u_0 - u}{^cZ} \\ 0 & \frac{f_y}{^cZ} & \frac{v_0 - v}{^cZ} \end{pmatrix} \, . \tag{6}$$

It is clear that matrix J_c depends on ${}^cZ$, which is the depth of a 3-D point in camera frame c.

Velocity of Camera Frame

Now, we assume that the variation of 3-D coordinates, with respect to camera frame c, is due to the ego-motion of the camera, which can be generally described by:

(1) ${}^w\dot{O}_c$: the linear velocity of camera frame c's origin, with respect to reference frame w.
(2) ${}^w\omega_c$: the angular velocity of camera frame c, with respect to reference frame w.

As the camera is mounted on the end-effector of a robot arm (or neck), the velocity of camera frame c is caused by the velocity of joint variables, denoted by the vector $\dot{q}$. And, the relationship between these two variations is governed by the differential kinematics, as follows:

$$\begin{cases} {}^w\dot{O}_c = J_o \bullet \dot{q} \\ {}^w\omega_c = J_\omega \bullet \dot{q} \end{cases} \tag{7}$$

where J_o and J_ω are the Jacobian matrices for linear and angular velocities, respectively.

If ${}^cR_w$ denotes the rotational transformation from reference frame w to camera frame c, the ego-motion of the camera with respect to its own frame c will be

$$\begin{cases} {}^c\dot{O}_c = {}^cR_w \bullet {}^w\dot{O}_c \\ {}^c\omega_c = {}^cR_w \bullet {}^w\omega_c \ . \end{cases} \tag{8}$$

Image Jacobian of Inner-loop Visual Servoing

As motion is always relative, a 3-D point will appear to undergo: (a) a linear motion described by $\{-{}^c\dot{O}_c\}$, and (b) an angular motion described by $\{-{}^c\omega_c\}$, with respect to camera frame c. As a result, the derivatives of a 3-D point's coordinates will be

$$\begin{pmatrix} {}^c\dot{X} \\ {}^c\dot{Y} \\ {}^c\dot{Z} \end{pmatrix} = -\left\{ {}^c\dot{O}_c + {}^c\omega_c \times \begin{pmatrix} {}^cX \\ {}^cY \\ {}^cZ \end{pmatrix} \right\} . \tag{9}$$

Then, the substitution of (7), (8) and (9) into (4) yields

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = -J_c \bullet \left\{ {}^cR_w \bullet J_o \bullet \dot{q} + \left[({}^cR_w \bullet J_w \bullet \dot{q}) \times \begin{pmatrix} {}^cX \\ {}^cY \\ {}^cZ \end{pmatrix} \right] \right\} . \tag{10}$$

Now, let S_x denote the skew-symmetrical matrix corresponding to the vector $({}^cX,\ {}^cY,\ {}^cZ)$. Finally, (10) can be expressed in the following compact form:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = J_{uv} \bullet \dot{q} \tag{11}$$

where

$$J_{uv} = -J_c \bullet \{{}^cR_w \bullet J_o - S_x \bullet {}^c R_w \bullet J_\omega\} \ . \tag{12}$$

In (11), J_{uv} is called the image Jacobian of inner-loop visual servoing. From (11) and (12), we can draw the following remarks:

- Matrix J_{uv} relates the variation of image feature(s) to the variation of joint variables of a robot arm (or neck).
- Matrix J_{uv} cannot be constant, because it depends on: (a) the configuration of the robot arm (or neck), and (b) the depth of a 3-D point in camera frame c (i.e. ${}^cZ$).
- Given a desired trajectory of q in joint space, it is possible to predict the trajectory of (u, v) in image space. However, the prediction requires the knowledge about J_{uv} which is not constant.
- Given a desired trajectory of (u, v) in image space, it is possible to predict the trajectory of q in joint space. Again, the prediction requires the knowledge about J_{uv} which is not constant.

J_{uv} being a time-varying matrix poses a major challenge to the scheme of inner-loop visual servoing [15].

4.2 Outer-loop Visual Servoing

With the use of binocular vision, we can obtain a constant matrix which relates the variation of image feature(s) to the variation of an end-effector frame's pose, which could be further related to the variation of joint variables through the pseudo-inverse of differential kinematics.

In practice, binocular vision can be mounted on the neck of a humanoid robot as shown in Fig. 5. Therefore, the visual feedback from the binocular vision can serve the purpose of guiding the positioning of either robot head (i.e. head-eye coordination), or robot hand (i.e. hand-eye coordination) towards a target-object in a 3-D space. Interestingly enough, the working principle of visual servoing with binocular vision will be the same for both head-eye and hand-eye coordinations [16].

In order to apprehend the simplicity of the scheme of outer-loop visual servoing, we briefly derive the so-called image-to-task mapping matrix, which is constant.

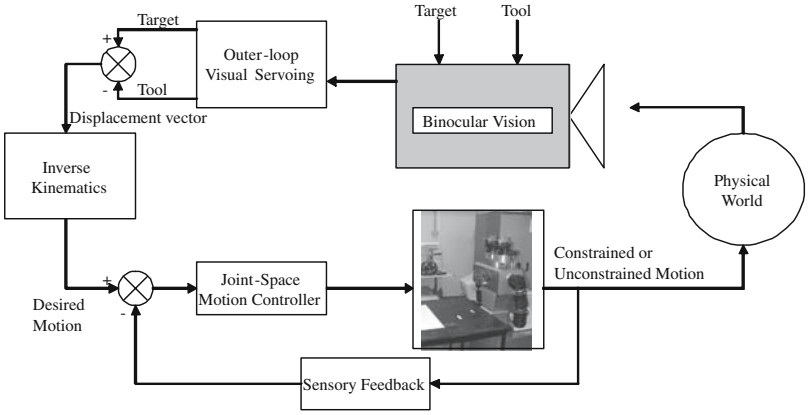


Fig. 5. Behavior-based vision using outer-loop visual servoing. The observation of variations in image space is mapped to the corresponding displacement vector in task space, which is then mapped to the desired output specified to the inner motion control loop

Qualitative Binocular Vision

Figure 6 shows the frame assignment in a typical setting of outer-loop visual servoing. Let $({}^{w}X,\ {}^{w}Y,\ {}^{w}Z)$ be the coordinates of an arbitrary point, denoted by P, in reference frame w at time $t = 0$. The reference frame can be any user-specified coordinate system. If $({}^{l}u,\ {}^{l}v)$ and $({}^{r}u,\ {}^{r}v)$ are the coordinates of the projections of point P, onto the two image-planes of a robot's binocular vision, we have

$$s_1 \bullet \begin{pmatrix} {}^{l}u \\ {}^{l}v \\ 1 \end{pmatrix} = H_1 \bullet \begin{pmatrix} {}^{w}X \\ {}^{w}Y \\ {}^{w}Z \\ 1 \end{pmatrix} \tag{13}$$

and

$$s_2 \bullet \begin{pmatrix} {}^{r}u \\ {}^{r}v \\ 1 \end{pmatrix} = H_2 \bullet \begin{pmatrix} {}^{w}X \\ {}^{w}Y \\ {}^{w}Z \\ 1 \end{pmatrix} \tag{14}$$

where s_1 and s_2 are the unknown scaling factors, and H_1 and H_2 the calibration matrices of the left and right cameras.

In (13), we purposely drop out the unknown scaling factor s_1. As a result, we obtain:

$$\begin{pmatrix} {}^{l}u \\ {}^{l}v \\ 1 \end{pmatrix} = H_1 \bullet \begin{pmatrix} {}^{w}X \\ {}^{w}Y \\ {}^{w}Z \\ 1 \end{pmatrix} . \tag{15}$$

The removal of the third row in (15) yields

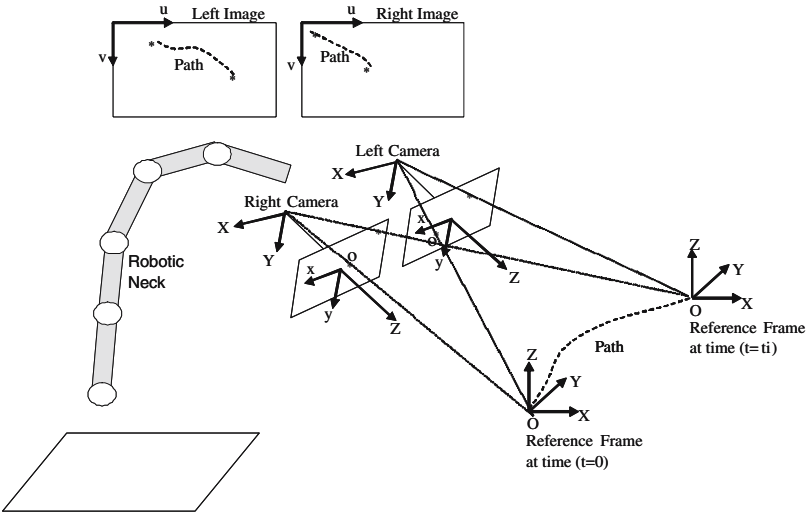


Fig. 6. Frame assignment in a typical setting of outer-loop visual servoing

$$\begin{pmatrix} {}^{l}u \\ {}^{l}v \end{pmatrix} = B_1 \bullet \begin{pmatrix} {}^{w}X \\ {}^{w}Y \\ {}^{w}Z \\ 1 \end{pmatrix} \tag{16}$$

with

$$B_1 = \begin{pmatrix} {}^{l}b_{11} & {}^{l}b_{12} & {}^{l}b_{13} & {}^{l}b_{14} \\ {}^{l}b_{21} & {}^{l}b_{22} & {}^{l}b_{23} & {}^{l}b_{24} \end{pmatrix}$$

Equation (16) describes the so-called *affine transformation*. But here, we call it the *qualitative projective mapping* by camera.

In a similar way, dropping out the scaling factor s_2 in (14) will result in the following qualitative projective mapping:

$$\begin{pmatrix} {}^{r}u \\ {}^{r}v \end{pmatrix} = B_2 \bullet \begin{pmatrix} {}^{w}X \\ {}^{w}Y \\ {}^{w}Z \\ 1 \end{pmatrix} \tag{17}$$

with

$$B_2 = \begin{pmatrix} {}^{r}b_{11} & {}^{r}b_{12} & {}^{r}b_{13} & {}^{r}b_{14} \\ {}^{r}b_{21} & {}^{r}b_{22} & {}^{r}b_{23} & {}^{r}b_{24} \end{pmatrix}$$

Binocular vision, governed by the qualitative projective mappings described by (16) and (17), is called *qualitative binocular vision*.

Qualitative Projective Mappings

Matrices (B_1, B_2) in (16) and (17) are the *calibration matrices* in a robot's qualitative binocular vision. Interestingly, the following properties can be easily verified:

- Property 1: The first column vectors in both B_1 and B_2 are the projections of the unit vector $(1, 0, 0)^t$ in a reference frame, onto the image planes of the two cameras in a robot's binocular vision.
- Property 2: The second column vectors in both B_1 and B_2 are the projections of the unit vector $(0, 1, 0)^t$ in a reference frame, onto the image planes of the two cameras in a robot's binocular vision.
- Property 3: The third column vectors in both B_1 and B_2 are the projections of the unit vector $(0, 0, 1)^t$ in a reference frame, onto the image planes of the two cameras in a robot's binocular vision.
- Property 4: The fourth column vectors in both B_1 and B_2 are the coordinates of the projection of the origin $(0, 0, 0)^t$ of a reference frame, onto the image planes of the two cameras in a robot's binocular vision.

We can advantageously exploit the above properties to easily determine the matrices B_1 and B_2. For instance, we can choose an orthogonal coordinate system, and project it onto the two image-planes of a robot's binocular vision. In practice, two possible solutions are as follows:

- If a robot's binocular vision has been calibrated in advance, one can define a virtual coordinate system (e.g. the reference frame in Fig. 6), and projects it onto the image-planes by using (H_1, H_2), which are the calibration matrices of binocular vision.
- If a robot's binocular vision is not calibrated, one can move a robot's end-effector into three orthogonal directions to define an orthogonal coordinate system (e.g. the reference frame in Fig. 6). This orthogonal coordinate system will be seen by the actual (un-calibrated) cameras of the binocular vision. This, in turn, supports the claim that a robot's qualitative binocular vision does not require the calibration matrices (H_1, H_2), if a robotic hand-arm is present.

Let's consider the base-vectors and the origin of a randomly chosen coordinate system, which is visible in a robot's binocular vision. Then, the images of the points

$$\{(1, 0, 0),\ (0, 1, 0),\ (0, 0, 1)\}$$

and the origin $(0, 0, 0)$, seen by the actual cameras, directly form the matrices (B_1, B_2), as follows:

- If $({}^l u_x,\ {}^l v_x)$, $({}^l u_y,\ {}^l v_y)$ and $({}^l u_z,\ {}^l v_z)$ are the coordinates of the points

  $$\{(1, 0, 0),\ (0, 1, 0),\ (0, 0, 1)\}$$

  projected onto the left-camera's image plane, then

$$B_1 = \begin{pmatrix} {}^{l}u_x - {}^{l}u_o & {}^{l}u_y - {}^{l}u_o & {}^{l}u_z - {}^{l}u_o & {}^{l}u_o \\ {}^{l}v_x - {}^{l}v_o & {}^{l}v_y - {}^{l}v_o & {}^{l}v_z - {}^{l}v_o & {}^{l}v_o \end{pmatrix} \tag{18}$$

where $({}^{l}u_o,\ {}^{l}v_o)$ are the coordinates of the origin $(0,0,0)$ projected onto the left-camera's image plane.

- If $({}^{r}u_x,\ {}^{r}v_x)$, $({}^{r}u_y,\ {}^{r}v_y)$ and $({}^{r}u_z,\ {}^{r}v_z)$ are the coordinates of the points

$$\{(1,0,0),\ (0,1,0),\ (0,0,1)\}$$

projected onto the right-camera's image plane, then

$$B_2 = \begin{pmatrix} {}^{r}u_x - {}^{r}u_o & {}^{r}u_y - {}^{r}u_o & {}^{r}u_z - {}^{r}u_o & {}^{r}u_o \\ {}^{r}v_x - {}^{r}v_o & {}^{r}v_y - {}^{r}v_o & {}^{r}v_z - {}^{r}v_o & {}^{r}v_o \end{pmatrix} \tag{19}$$

where $({}^{r}u_o,\ {}^{r}v_o)$ are the coordinates of the origin $(0,0,0)$ projected onto the right-camera's image plane.

Image-to-Task Mapping

In qualitative binocular vision, (B_1, B_2) fully describe the projections of a configuration (coordinate system) onto the image-planes. Thus, motion planning in task space becomes equivalent to motion planning in image space, if inverse projective mapping is available.

In image space, we can determine the displacement vectors $(\triangle^{l}u,\ \triangle^{l}v)$ and $(\triangle^{r}u,\ \triangle^{r}v)$, which move towards the target. Then, the question is: What should be the corresponding displacement vector $(\triangle^{w}X, \triangle^{w}Y, \triangle^{w}Z)$, in task space, which moves a robot's hand from its initial configuration towards the final configuration?

Because B_1 is a constant matrix, the differentiation of (16) will yield the following difference equation:

$$\begin{pmatrix} \triangle^{l}u \\ \triangle^{l}v \end{pmatrix} = C_1 \bullet \begin{pmatrix} \triangle^{w}X \\ \triangle^{w}Y \\ \triangle^{w}Z \end{pmatrix} \tag{20}$$

with

$$C_1 = \begin{pmatrix} {}^{l}u_x - {}^{l}u_o & {}^{l}u_y - {}^{l}u_o & {}^{l}u_z - {}^{l}u_o \\ {}^{l}v_x - {}^{l}v_o & {}^{l}v_y - {}^{l}v_o & {}^{l}v_z - {}^{l}v_o \end{pmatrix}$$

Similarly, by differentiating (17), we will obtain the difference equation corresponding to the right camera, as follows:

$$\begin{pmatrix} \triangle^{r}u \\ \triangle^{r}v \end{pmatrix} = C_2 \bullet \begin{pmatrix} \triangle^{w}X \\ \triangle^{w}Y \\ \triangle^{w}Z \end{pmatrix} \tag{21}$$

with

$$C_2 = \begin{pmatrix} {}^r u_x - {}^r u_o & {}^r u_y - {}^r u_o & {}^r u_z - {}^r u_o \\ {}^r v_x - {}^r v_o & {}^r v_y - {}^r v_o & {}^r v_z - {}^r v_o \end{pmatrix}$$

Equations (20) and (21) impose four constraints on these three unknowns:

$$(\triangle^w X, \triangle^w Y, \triangle^w Z) \ .$$

If we define

$$\triangle I = \left(\triangle^l u, \triangle^l v, \triangle^r u, \triangle^r v\right)^t \tag{22}$$

the displacement vector in image space, and

$$\triangle P = (\triangle^w X, \triangle^w Y, \triangle^w Z)^t \tag{23}$$

the displacement vector in task space, the least-square estimation of $\triangle P$ will be

$$\triangle P = A \bullet \triangle I \tag{24}$$

with

$$A = (C^t \bullet C)^{-1} \bullet C^t \tag{25}$$

and

$$C = \begin{pmatrix} {}^l u_x - {}^l u_o & {}^l u_y - {}^l u_o & {}^l u_z - {}^l u_o \\ {}^l v_x - {}^l v_o & {}^l v_y - {}^l v_o & {}^l v_z - {}^l v_o \\ {}^r u_x - {}^r u_o & {}^r u_y - {}^r u_o & {}^r u_z - {}^r u_o \\ {}^r v_x - {}^r v_o & {}^r v_y - {}^r v_o & {}^r v_z - {}^r v_o \end{pmatrix} \ . \tag{26}$$

Because C is a constant matrix, A will also be a constant matrix. This indicates that the mapping from image space to task space can be done by a constant matrix. Clearly, this is the simplest result that one could obtain. Here, we call matrix A the *image-to-task mapping matrix*.

If we consider $\triangle I$ the error signal to a robot's action controller, A will be the proportional control gain matrix and $\triangle P$ the control action, which acts on a robot's motion controller as shown in Fig. 5. In practice, we can introduce one extra proportional gain g $(0 < g < 1)$, which is a scalar, so that the transient response of a robot's action controller can be manually tuned. Accordingly, (24) becomes

$$\triangle P = g \bullet A \bullet \triangle I \ . \tag{27}$$

Alternatively, one can also choose g to be a gain matrix, as follows:

$$g = \begin{pmatrix} g_x & 0 & 0 \\ 0 & g_y & 0 \\ 0 & 0 & g_z \end{pmatrix} .$$

If the calibration matrices (H_1, H_2) are known, (27) can serve as the so-called *image-guided motion planning function*. Otherwise, it can serve as the control law in an image-guided action controller.

Example of Hand-Eye Coordination

Figure 7 shows two experiments of robotic hand-eye coordination. The objective here is to automatically generate a feasible path (i.e. a set of spatial locations) in task space, which brings a robot hand from an initial configuration to a final configuration. We can see that the application of (27) produces stable results for the first (Fig. 7b and Fig. 7c) and second (Fig. 7d and Fig. 7e) experiments.

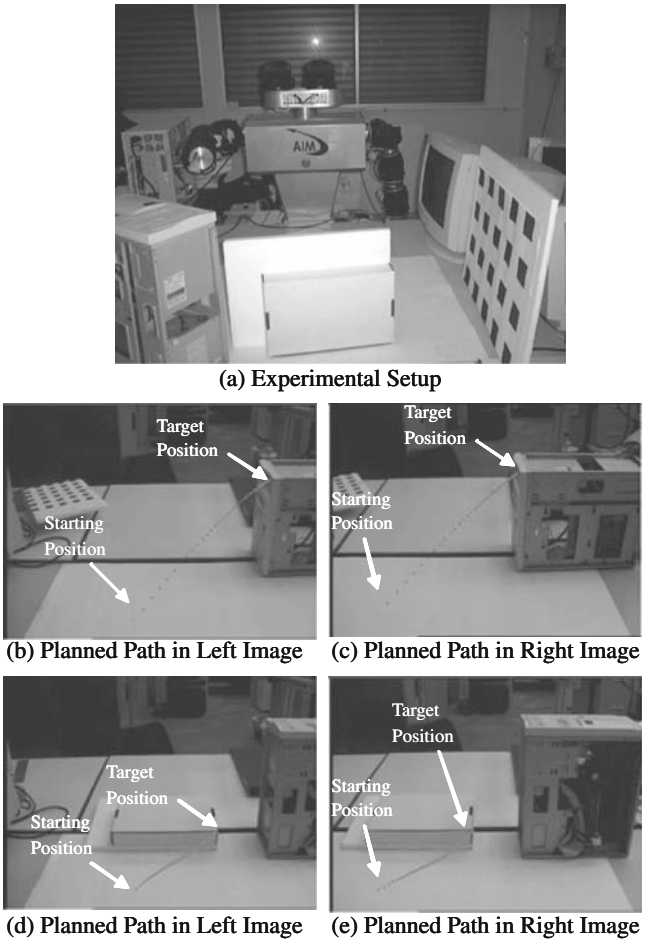


Fig. 7. Real examples of image-guided motion planning for hand-eye coordination

5 Reconstructive Vision

Reconstructive vision was popularized by David Marr [17] in the early 1980s, and aims at inferring 3-D geometry of a scene or object from (stereo) images. It is believed that reconstructive vision is a necessary step before visual guidance in robotics, and object recognition in machine perception. Although visual learning is indispensable, which will be discussed in the section on Model-Based Vision, the crucial issue here is the 3-D reconstruction of a visual object using binocular, or multiple-view, vision. [19]

Refer to Fig. 8. There are two strategies to tackle the problem of reconstructing the 3-D geometry of a visual object, namely: (a) forward reconstruction [18], and (b) inverse reconstruction [16, 20], which are outlined below.

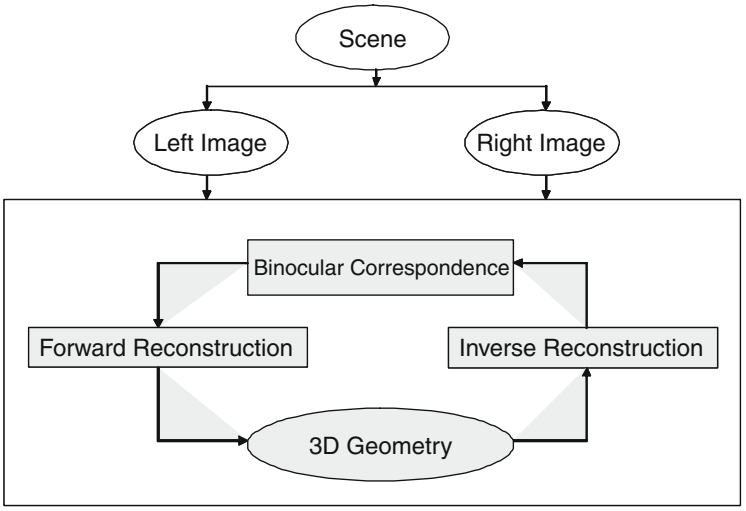


Fig. 8. A generic framework of reconstructive vision. The forward reconstruction starts with feature correspondence in image space, followed by geometry estimation in a 3-D scene. In contrast, the inverse reconstruction starts with the prediction of geometry in a discrete 3-D scene, following by verification of feature correspondence in image space

5.1 Forward Reconstruction

The theoretical foundation of forward reconstruction is the (13) and (14), in which the coordinates $({}^{w}X,\ {}^{w}Y,\ {}^{w}Z)$ are unknown, while the image coordinates, $({}^{l}u,\ {}^{l}v)$ and $({}^{r}u,\ {}^{r}v)$, and the calibration matrices, H_1 and H_2, are known in advance.

Mathematically, it appears straightforward to apply (13) and (14) to derive a unique solution for the unknown $({}^{w}X,\ {}^{w}Y,\ {}^{w}Z)$. However, this solution implies three necessary conditions to be met beforehand. They are:

- Camera Calibration: It means that the matrices H_1 and H_2 are available in advance or could be estimated online [22].
- Feature Extraction: In general, there are two types of image features: (a) discontinuity-related features (e.g. edges), [25] and (b) continuity-related features (e.g. uniformly colored, or textured, regions) [27].
- Binocular Correspondence: It refers to the issue of finding the corresponding feature in one image, given a similar feature in the other image of the binocular vision [28].

The first two conditions do not pose much technical challenge in robot vision. However, the third condition still remains an open issue. The current progress of research into practical solutions to binocular correspondence can be summarized as follows:

Approaches Using Epipolar Line

It is well-known that when the configuration of the two cameras in binocular vision remains unchanged, the coordinates of an image feature in the left camera are related to the coordinates of the corresponding image feature in the right camera by a so-called *fundamental matrix* F, in the following way:

$$\begin{pmatrix} {}^{l}u & {}^{l}v & 1 \end{pmatrix} \bullet F \bullet \begin{pmatrix} {}^{r}u \\ {}^{r}v \\ 1 \end{pmatrix} = 0\,. \tag{28}$$

It means that if the coordinates $({}^{l}u,\ {}^{l}v)$ are chosen in the left image-plane, its corresponding coordinates must lie on the so-called *epipolar line*, which is analytically described by (28). Here, we have only a necessary condition for the binocular correspondence of image features. The search for a correct match must subsequently invoke the so-called *similarity test* on appearance surrounding image features [28]. So far, there is no widely-accepted solution for similarity test. This is because there is no theoretical justification behind similarity test, if the baseline between the two cameras in binocular vision is reasonable large. As a result, we claim that there is no generally-practical solution for binocular correspondence using epipolar line.

Approaches Using Differential Epipolar Line

It is interesting to know that (28) could be differentiated, with respect to time, in order to yield the following constraint:

$$\begin{pmatrix} {}^{l}\dot{u} & {}^{l}\dot{v} & 0 \end{pmatrix} \bullet F \bullet \begin{pmatrix} {}^{r}u \\ {}^{r}v \\ 1 \end{pmatrix} + \begin{pmatrix} {}^{l}u & {}^{l}v & 1 \end{pmatrix} \bullet F \bullet \begin{pmatrix} {}^{r}\dot{u} \\ {}^{r}\dot{v} \\ 0 \end{pmatrix} = 0\,. \tag{29}$$

If we know the location and displacement for each image-feature in the case of dynamic binocular vision (or static binocular vision with a pair of composite cameras [16]), (29) is a sufficient condition for selecting the correct match of binocular correspondence.

Approaches Using Multiple Models

The spirit of using multiple models in handling binocular correspondence implicitly exists in all approaches using epipolar line or differential epipolar line. This implies that the explicit exploitation of multiple models may be a viable solution to tackle the difficult problem of binocular correspondence. [29]

Thus, how to use hierarchical models, such as appearance model, feature model and structure model, to solve the generic matching problem in reconstructive vision may be a promising topic for future investigation in robot vision.

5.2 Inverse Reconstruction

Inverse reconstruction consists of two steps: (a) to predict all permissible values of 3-D coordinates, and (b) to validate those coordinates whose images in the binocular, or multiple-view, vision are photometrically consistent.

The well-known approach of inverse reconstruction is the so-called *voxel coloring* [20]. Voxel coloring divides a 3-D space into a matrix of regular voxels, each of which is to be assigned one of these colors: on-surface, below-surface, and above-surface. The coloring of voxels invokes a process of testing the photometric consistency among the images.

Instead of digitizing all the three coordinates X, Y and Z into discrete values, a better approach is to selectively digitize one coordinate [22]. This is because, given image coordinates (u, v), the corresponding 3-D coordinates X, Y and Z are not independent. Interestingly enough, this approach has a sound mathematical principle, as outlined below:

Without loss of generality, we choose to digitize the Z coordinate corresponding to image coordinates $({}^{l}u,\ {}^{l}v)$ in the left camera. If we choose the expected accuracy of a Z coordinate to be within $\pm\triangle {}^{w}Z$, then we have:

$$ {}^{w}Z_k = {}^{w}Z_{min} + k \bullet \triangle\ {}^{w}Z \tag{30}$$

where k indicates the k-th discrete value of Z, and ${}^{w}Z_{min}$ the minimum value of Z.

Now, given an image feature at coordinates $({}^{l}u,\ {}^{l}v)$, we can predict a set of "n" possible 3-D coordinates:

$$\{({}^{w}X_k,\ {}^{w}Y_k,\ {}^{w}Z_k), \quad k = 0, 1, 2, \ldots, n\}$$

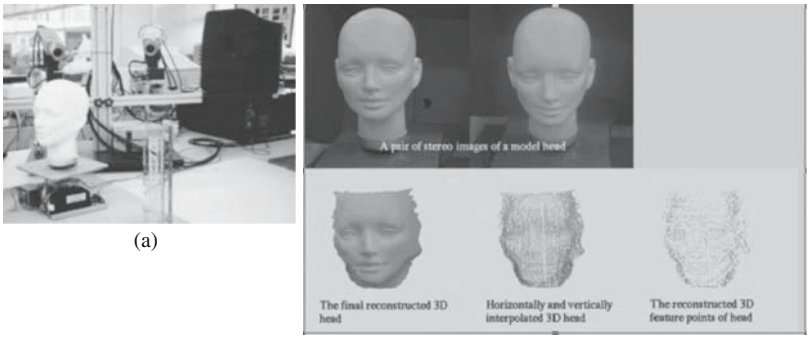
where ${}^{w}Z_k$ is first computed from (30), and $({}^{w}X_k,\ {}^{w}Y_k)$ are then computed from (13).

Knowing $\{({}^{w}X_k,\ {}^{w}Y_k,\ {}^{w}Z_k),\quad k = 0, 1, 2, \ldots, n\}$, all the possible locations

$$\{({}^{r}u_k,\ {}^{r}v_k),\quad k = 0, 1, 2, \ldots, n\}$$

of binocular correspondence in the right camera can be easily computed from (14). Among these locations, we select the index k, which optimizes the photometric consistency at coordinates $({}^{l}u,\ {}^{l}v)$ (in the left image) and coordinates $({}^{r}u_k,\ {}^{r}v_k)$ (in the right image).

In fact, index k not only indicates the correct match of binocular correspondence, but also the 3-D coordinates $({}^{w}X_k,\ {}^{w}Y_k,\ {}^{w}Z_k)$ in a direct way. Figure 9 shows one example of interesting results obtained by the proposed method.



(a)

(b)

Fig. 9. An example of experimental results by using the proposed approach of inverse reconstruction. The binocular vision consists of two cameras as shown in (**a**). For a pair of stereo images, we select edges as image features. The 3-D coordinates corresponding to edges are computed first. The final result is obtained by a series of successive interpolations of 3-D coordinates in both horizontal (i.e. along X axis) and vertical (i.e. along Y axis) directions, respectively

5.3 Future Challenge

Until today, it is still a challenge to infer the geometry of a 3-D object or scene with binocular vision in a general setting. The main obstacle to reconstructive vision is the problem of binocular correspondence.

In fact, there are two difficulties behind the problem of binocular correspondence. The first one is occlusion, which causes some features visible in one camera but not in the other camera. The second difficulty is photometric discrepancy, which means the variation in appearance as long as the baseline between two cameras are not equal to zero.

We know that human vision does not suffer the problem of binocular correspondence. In addition, each eye of human vision could perform well, independently. This observation suggests that a binocular vision should be treated as a combination of two cooperative monocular visions, each of which should be knowledge-, or model-based, as shown in Fig. 10. Interestingly enough, this hypothesis reveals the importance of model-based vision, and even knowledge-based vision (i.e. cognitive vision).

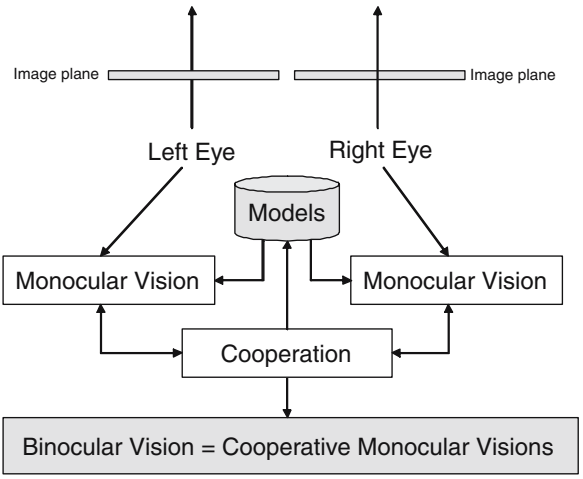


Fig. 10. A cooperative framework for binocular vision, in which the two monocular visions should be able to perform both independently and cooperatively

6 Model-based Vision

In instrumental, behavior-based, and reconstructive visions, only geometrical signals (i.e. pixels, or a combination of pixels) are considered. However, an image also contains other types of visual signal such as chromatic signals (i.e. colors) and textural signals (i.e. textures). Most importantly, the ultimate purpose of vision is not to simply manipulate geometrical signals, but to derive meanings (i.e. properties and constraints) from images, as outlined in Fig. 11.

Therefore, how to transform visual signals into knowledge via features is the biggest challenge in robot vision. It is worth noting that most of efforts in computer, or robot, vision has been devoted to feature extraction, perceptual grouping, and object recognition. Like it or not, all proposed methods are implicitly, or explicitly, model-based. This indicates that the aspect of model-based vision is extremely important in robot vision. And, interestingly enough, under the aspect of model-based vision, three important issues have been investigated, namely:

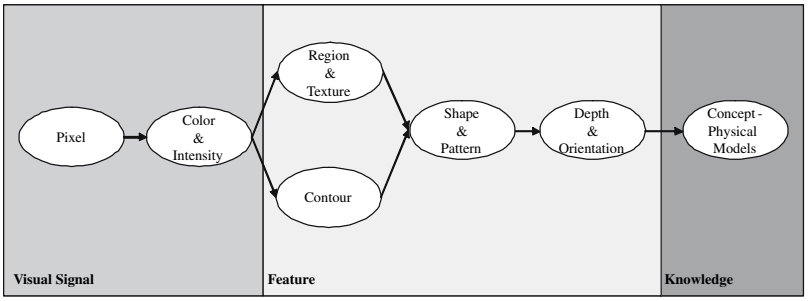


Fig. 11. The flowchart from visual signal to knowledge in robot vision. An image may contain three types of signal: geometrical, chromatic, and textural signals. These signals are transformed into features, which are finally grouped into meaningful entities to be recognized by innate mental processes

- Appearance, or Pattern, Recognition
- Object Recognition
- Image Interpretation

6.1 Appearance, or Pattern, Recognition

There is a very large community of researchers working on the interesting problem of detecting visual features, and recognizing visual shapes or forms which are commonly called as *pattern* [41].

Under the topic of pattern recognition, such as the recognition of printed or handwritten characters, the general framework consists of: (a) pattern detection, (b) pattern description, (c) pattern training, and (d) pattern classification. In this framework, models (i.e reference patterns) are represented in feature spaces. The process of learning is invoked in order to build the database of models (or the representation of models in feature space). It is clear that the general framework of pattern recognition relies on the so-called *2D-to-2D matching paradigm*. Recently, the hot topic on face recognition also falls into this paradigm.

Under the topic of feature extraction, Hough Transform is a typical example of model-based approach to detect image features (e.g. lines). Another typical example of model-based approaches in feature extraction is color image segmentation based on the use of neural network. As for the detection of discontinuity in images, most of the edge detectors are implicitly model-based because they always assume a certain type of edge model, before undertaking a process of optimization in order to enhance the response of signals at the locations of edge.

Here, we would like to define the problem of model-based detection of image features as the problem of *appearance recognition*, because it is beneficial to explicitly think that image features could be represented and recognized by effectively following a top-down approach (i.e. model-based approach).

Future Challenge

Although the success in pattern recognition is remarkable, the progress in appearance recognition (e.g. image segmentation) is too slow, and has already seriously undermine the standing of computer, or robot, vision among the other scientific disciplines (e.g. computer graphics). A promising direction toward making significant progress is to adopt the philosophical thinking of developmental principle [54, 55].

Human vision is essentially developmental. There are certainly innate mental processes underlying the developmental aspect of human vision. In our opinion, it is unlikely to discover what these innate mental processes will be, by simply studying human eyes and vision [5]. For example, the discovery of aerodynamics was not due to the study of birds, but to the repeated attempts of making flying machines. Similarly, we must make attempt to build robot vision with developmental characteristics. Hopefully, this endeavor of building developmental vision may lead to the thorough answer to the question of how human vision works. At this moment, we can raise two fundamental questions, as follows:

- What is the innate principle for a robot vision to autonomously learn, and recognize colors?
- What is the innate principle for a robot vision to autonomously learn, and recognize shapes?

6.2 Object Recognition

Object recognition has been considered to be the next step after the success of instrumental, or reconstructive, vision. And, it is believed that the most effective way to detect, categorize and identify objects is to compare 3-D description derived from range data with the possible 3-D descriptions pre-stored in a database (i.e. models). This motivation has resulted in the so-called *3D-to-3D matching paradigm* for object recognition.

Model-based object recognition was pioneered by Nevatia & Binford (1977) [31], followed by Oshima & Shirai (1983) [35], Fisher (1983) [36], Horaud & Bolles (1985) [38], Grimson & Lozano-perez (1984) [37], and Faugeras & Hebert (1986) [39]. As we mentioned above, the objective is to compare 3-D description of an observed object in range data with the possible 3-D descriptions pre-stored in a database in order to achieve the results of categorization, identification and positioning in a 3-D space (i.e. pose and scale). Although different researchers emphasize different aspects of geometric modelling, all 3D-to-3D matching approaches follow the similar framework as shown in Fig. 12.

Future Challenge

Under the paradigm of 3D-to-3D matching, researchers have extensively investigated the use of all possible geometric primitives or models. Despite the

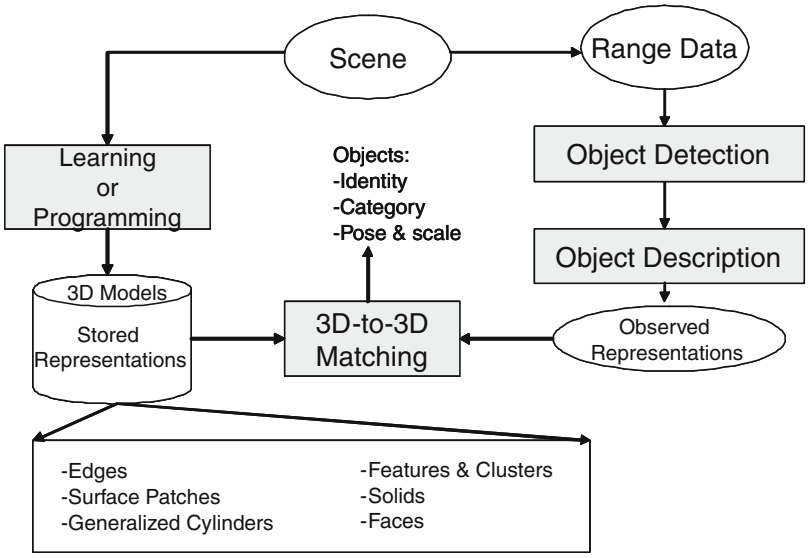


Fig. 12. The general 3D-to-3D matching paradigm for model-based object recognition. The pre-stored 3-D representations could be learnt or programmed. After object detection, 3-D description is derived from range data. Subsequently, the observed representation is matched against the pre-stored representation in order to achieve the results in terms of identity, category, pose and scale of an observed object

continuous effort devoted to object recognition, the success is very limited. Some reasons behind the limited success in object recognition are as follows:

(1) 3-D reconstruction and object segmentation (i.e. detection) is still problematic in real situation.
(2) Object modelling only considers geometric aspect of visual objects. There is no consideration of meanings.
(3) The negligence of "object semantics" results in the difficulty of coping with dynamic scene.

These issues still pose challenges to model-based object recognition.

6.3 Image Interpretation

It is a commonly accepted fact that human vision is able to interpret images, or photos, in the absence of 3-D range data. Therefore, it may be possible to compare 2-D description derived from images, or photos, with the possible 3-D description which pre-stored in a database (i.e. models). This speculation has motivated researchers to look into the problem of image or photo interpretation.

For example, inspired by the extraordinary ability of human vision in interpreting images or photos, Lowe (1980) [33] and Brooks (1981) [34] have

pioneered the works on model-based image interpretation. The objective is to compare 2-D description of an observed object in image space with the possible 3-D descriptions which is pre-stored in a database in order to achieve the results of categorization, identification and positioning in a 3-D space (i.e. pose and scale). The general framework underlying model-based image interpretation follows the so-called *2D-to-3D matching paradigm*, as shown in Fig. 13.

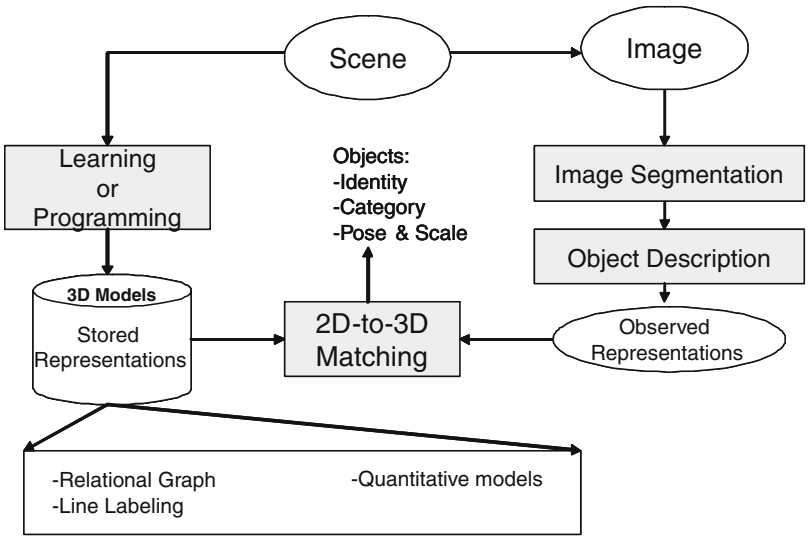


Fig. 13. The general 2D-to-3D paradigm for model-based image interpretation. Similarly, the pre-stored 3-D representations could be learnt or programmed. First of all, an image is segmented into a set of possible objects in image space. After object description in image space, the observed 2-D representation is matched against the pre-stored 3-D representation in order to achieve the results in terms of identity, category, pose and scale of an observed object

Future Challenge

Model-based image interpretation is not actively pursued because of the following major difficulties:

(1) Image segmentation is still problematic in real situation.
(2) It does not make too much sense to match 2-D representations against 3-D representations, especially when quantitative aspects of geometric models must be considered.
(3) This is no convincing solution to infer the pose or scale of a selected model in a 3-D scene.

Because of the difficulties faced by model-based vision for object recognition or image interpretation, there is no commonly accepted methodology so far. And, there is no method which could achieve acceptable results under any general context. Then, should we abandon the pursuit of model-based vision?

We know that human vision is robust in the presence of occlusion, because each of our eyes could perform independently. This observation indicates that human vision is cooperative as outlined in Fig. 10. In addition, human vision is powerful enough to effortlessly interpret images, and recognize familiar objects. This means that human vision is definitely model-based, and even knowledge-based. Thus, it is worth investigating new engineering solutions underlying model-based vision, which could enable future robots to meaningfully interpret images, and faithfully recognize objects.

Here, we advocate that model-based vision should take images as input, and produce 3-D representations of objects or scene in a 3-D space as output. In addition, we advocate the scheme of 2D-to-2D matching in order to make model-based image interpretation tractable. In other words, we believe that object-to-model matching should be done in image space. This is because object-to-model matching in 3-D space is not realistic, and the matching between 2-D objects and 3-D models does not make too much sense. In particular, we would like to suggest a new framework which could enable a model-based vision to both recognize objects and interpret images.

As shown in Fig. 14, the proposed new framework consists of six modules, namely: (a) model selection, (b) model sizing, (c) model placing, (d) model-image synthesis, (e) 2D-to-2D matching, and (f) visual learning.

Model Selection

Given an image as input, the first difficult task is to detect the presence of a possible object. So far, this is still an open issue. However, it is relatively easy to extract image features, and predict the groupings of image features into visual entities for recognition.

Hence, given a visual entity detected in an image as input, the purpose of model selection is to list out all the possible 3-D models, which could be matched against the visual entity in an image. Clearly, model selection should be feature-based. In the worst scenario, one can select all the 3-D models in the database as the candidates to match against a visual entity seen in an image.

Model Sizing

Once a 3-D model is chosen to match against a visual entity in an image, the next task is to synthesize a model-image at an appropriate view, because the matching has to be between a model-image and an object-image.

Before a model-image could be synthesized, we must know: (a) how to appropriately size a selected 3-D model, and (b) how to appropriately position

and orientate a selected 3-D model in a 3-D space. The former is the problem of model sizing. And, the latter is the problem of model placing.

In general, a visual object may vary in size. Given a particular visual object, it is not realistic to pre-store all the representations corresponding to all the possible sizes. Hence, model sizing must be done according to the object-image which is given as input.

We know that human vision is qualitative. In other words, we are able to meaningfully interpret images or photos, but without the detail about the exact geometrical dimension of a scene. This implies that human vision solves the problem of model sizing by simply fixing a model's size to a value of its own convenience.

However, as robot vision can be quantitative, it is possible to find an engineering solution to exactly fix the size of a selected 3-D model. One hypothesis is as follows:

- Step 1: Invoke reconstructive vision to compute view-independent characteristics of perceived image-features (e.g. the lengths of line segments).
- Step 2: Match the topology of perceived image-features against the topology of a selected model's features. This matching is size-independent.
- Step 3: Size the selected 3-D model according to the match result.

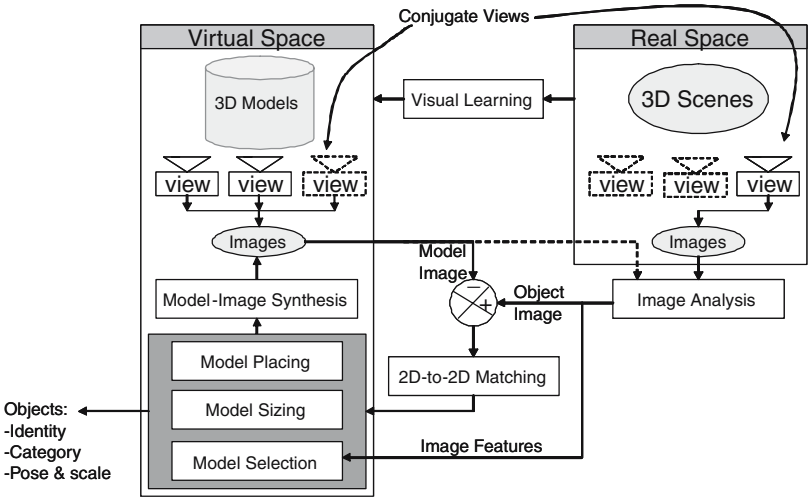


Fig. 14. A new framework of model-based vision, which consists of four important modules: (**a**) feature-guided selection of models, (**b**) model sizing for determining the scale, (**c**) model placing which determines the pose of a selected 3-D model in its 3-D space, and (**d**) the matching between a 2-D object-image and a 2-D model-image. Most importantly, these four modules are to operate in a closed-loop in order to achieve the robustness. And, the 3-D models should be learnt, instead of being pre-programmed by humans

Model Placing

As we mentioned above, a 3-D model must be properly sized and placed in its own 3-D space. Therefore, the purpose of model placing is to appropriately position and orientate a selected 3-D model in its own 3-D space, which is also called the *virtual space* as shown in Fig. 14.

Again, in general, a visual object may appear in different orientations and positions. Although it is possible to pre-store the reference views of a visual object's model as what has been proposed in the literature [49], it may not be accurate enough to generate an arbitrary view from a combination of reference views. Hence, it is necessary to determine the correct position and orientation of a selected 3-D model, before undertaking the synthesis of a model-image.

Mathematically, model placing with respect to a fixed (synthetic) view is equivalent to the problem of placing a (synthetic) view with respect to a fixed 3-D model. The advantage of considering the problem of placing a view with respect to a fixed 3-D model is that there is a readily available solution to it. For example, the three-view approach states that an unknown view (e.g. the conjugate view in the virtual space in Fig. 14) could be uniquely determined, if the motion transformation (i.e. rotation and translation) between another pair of views is known. [16] In practice, we can consider that the input image or photo is taken at the real view in the real world. Then, we can define a pair of synthetic views in the virtual world, as shown in Fig. 14. By applying the three-view approach, we can determine the conjugate view in the virtual space, which exactly corresponds to the real view if the real and virtual spaces have the same common contents.

The three-view approach represents a sound engineering solution to the problem of model placing, which is a decisive step toward transforming the ill-posed problem of 2D-to-3D matching into the tractable problem of 2D-to-2D matching. More sophisticated solutions are expected to emerge in future.

Model-image Synthesis

Given a selected 3-D model and a properly-placed synthetic view, the problem of model-image synthesis is to generate photo-realistic images. In this regard, we can leverage on the advance in computer graphics to come out suitable solutions. Nevertheless, there is still a challenge with regard to the modelling of objects, if we have to consider their complex aspects of geometry, appearance, physics, and even semantics. And, this leads to the important issue of visual learning.

2D-to-2D Matching

The purpose of 2D-to-2D matching aims at recognizing an object-image as the instance of an model-image. Here, recognition implies two prioritized objectives, namely: (a) object categorization, and (b) object identification. And, we advocate that 2D-to-2D matching should be hierarchical, and that the

geometrical aspects of models should include the representations of structure, feature and appearance. In general, we suggest the following procedure

- Step 1: Structure Matching
- Step 2: Feature Matching

for object categorization, and the following procedure

- Step 1: Feature Matching
- Step 2: Appearance Matching

for object identification.

Consequently, a complete procedure for 2D-to-2D matching in model-based vision will include

- Step 1: Structure Matching
- Step 2: Feature Matching
- Step 3: Appearance Matching

Visual Learning

Model-based vision is driven by models in order to achieve object recognition, and even image (or photo) interpretation (NOTE: features can be treated as degenerated objects). Thus, models play a crucial role in model-based vision. Then, the question will be: how to acquire models? In other words, should model acquisition be done by pre-programming, or by visual learning?

In our opinion, it is necessary to acquire object-models through visual learning. In engineering term, learning is a process consisting of modelling, optimization and representation. Learning can be either supervised or unsupervised. Learning could also be reinforced by an evaluation of the learning outcome.

In fact, the first concern in visual learning is how to model visual objects [32, 33]. So far, attempts have been made to model visual objects at these three hierarchical levels:

- Appearance Model:
  It refers to the representation of the image(s) of a visual object. A typical example is the sample image of a human face. An appearance model can be as simple as one, or a set, of sample images (i.e. templates) [44, 49]. But, it can also be very complex for textured images [27].
- Feature Model:
  It refers to the representation of a visual object (or its images) in terms of geometrical primitives (e.g. lines, curves, surface patches) [48]. The boundary representation (BR) method in computer graphics is a typical example of feature model [51].
- Structure Model:
  It refers to the representation of components and their combination in a visual object [2]. The constructive solid geometry (CSG) is a typical

example of structure model [51]. For complex shapes, one can use the so-called *generalized cones* [50].

In the literature, the existing strategy of exploiting object-models is narrowly focused on one or two types of models [44]. However, in order to cope with a vast variety of visual objects in the real world, we argue that a model-based vision system must adopt the hierarchical representations of visual objects at the above-mentioned three levels.

Then, the second concern in visual learning is how a model-based vision system acquires its database of object-models. In principle, there are at least two possible ways of acquiring object-models. They are:

- Pre-programming by Humans:
  It refers to the intervention of human beings in order to pre-program object-models into a database. This is a passive mode of learning (i.e. spoon-feeding). A pre-programmed object-model is the replicate of what the human vision perceives or has perceived. Since the viewing-parameters (e.g. intrinsic and extrinsic parameters of an eye) of a human's visual perception system is not the same as the ones of a robot's visual perception system, the problem of consistency of viewing-parameters arises. We believe that this is one of the main obstacles undermining the success of model-based vision.
- Autonomous Learning:
  It refers to a learning process, in which a robot's vision system autonomously models, optimizes and represents visual objects. In this way, a robot will have the capability of autonomously enriching its internal representation of the physical world. This spirit is in line with the framework of developmental robotics [54, 55]. Most importantly, if the object-models are autonomously learnt by a robot itself, the problem of consistency of viewing-parameters voids.

Autonomous mental development by robots is a promising direction of research. So far, there is no definite answer yet to the question of how to autonomously learn visual objects and their behaviors in a visual scene.

7 Cognitive Vision

A complex image is worth more than thousand words. Indeed, vision has a strong connection to cognition. This reality is largely due to the fact that vision enables the effective acquisition of knowledge about a scene through its images or videos. Hence, the cognitive aspect of robot vision should not be considered to be less important than behavior-based vision. On the contrary, the development of engineering principles and solutions underlying cognitive aspect of robot vision will still remain the biggest challenge for years.

7.1 A Meaning-Centric Framework

Cognitive vision aims to gain knowledge about a visual scene through its images or videos. Hence, the output from a cognitive vision system should be the meaningful description of a visual scene in terms of its properties, constraints and evolutions in space and time. In order to achieve this objective, we propose a meaning-centric framework, as shown in Fig. 15, in which there are many challenges waiting ahead.

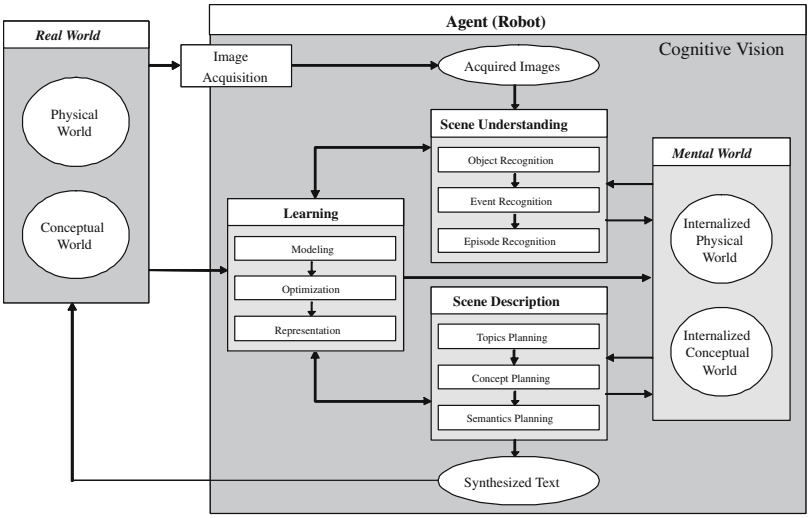


Fig. 15. A meaning-centric framework for cognitive vision. Learning plays a crucial role of internalizing not only the physical world but also the conceptual world. Scene understanding is model-driven, which consists of object recognition, event recognition and episode recognition. The ultimate output from cognitive vision is the description of a visual scene in the form of a natural language. Hence, scene description is a special module in cognitive vision

7.2 Future Challenges

The three functional modules in the proposed meaning-centric framework are, as follows:

- Learning:
  In general term, learning consists of modelling, optimization and representation. As the goal of cognitive vision is to gain the meanings of a visual scene through its images or videos, learning in cognitive vision plays a dual role. On one hand, it is necessary to learn the meanings of the physical world in order to understand a visual scene as a result of model-based

vision. On the other hand, it is necessary to learn the meanings of the conceptual world in order to meaningfully describe a visual scene. Hence, the learning in a cognitive vision system must model both the physical and conceptual worlds.

- Scene Understanding:
  The meanings of a visual scene consists of its properties, constraints and evolutions in space and time. Therefore, scene understanding must first tackle the problem of object recognition. When more than one visual object are present and interact in a visual scene, it is necessary to identify the layout (i.e. spatial configuration) of, and interactions among, these visual objects. This task is called the *event recognition*. As the spatial configurations and interactions may change and evolve in time, this gives rise to the need of undertaking the so-called *episode recognition*.
- Scene Description:
  In the conceptual world, meanings are categorized into topics and concepts. Hence, the meanings acquired from a visual scene must first be placed into the right context of topics (e.g. is it a foot-ball match or a manufacturing process in a production line of cars?). This task is called the *topics planning*. Under the right context of topics, then, the perceived meanings can be appropriately grouped into concepts. We call this task the *concept planning*. Finally, with the help of constraint networks in both physical and conceptual worlds[a], perceived concepts can be appropriately translated into the correct words and sentences. This last task is called the *semantic planning*.

As cognitive vision is a relatively new discipline in engineering, established theories or methodologies underlying learning, scene understanding and scene description are yet expected to surface out in future.

8 Conclusions

In this paper, we have presented a novel holistic view about robot vision. In particular, we have revealed the multidisciplinary nature of robot vision in terms of instrumental vision, behavior-based vision, reconstructive vision, model-based vision, and cognitive vision. As robot vision has not yet reached a mature stage, we do encourage more efforts to be devoted to this promising field.

Our first intention here is to help young researchers gain confidence in advancing robot vision research toward right directions. We hope that this paper will be perceived as having objectively highlighted the most, but not all, important achievements in robot vision research, and some biggest challenges for future investigations, especially issues concerning cognitive vision, which

[a]This is part of our ongoing research, which necessitates a full paper to clearly explain them

is indispensable for a physical agent, such as a humanoid robot, to gain visual awareness, to perform learned behaviors, and even to manifest emotions in response to certain types of visual awareness.

Our second intention here is to provide the necessary evidence to support the claim that a robot without vision can hardly become an intelligent and autonomous creature. We hope that this paper will be perceived as having clearly stated what robot vision is all about, and why robot vision is powerful enough to guide the behaviors of a physical agent such as a humanoid robot, and also to acquire the knowledge (or meanings) about a visual scene by a physical agent itself.

We acknowledge that this paper has said little about the great achievements in image processing, feature extraction and camera calibration. The reason behind this negligence is because the techniques in these areas play an important, but supporting, role to the success of robot vision.

References

1. Grimson, W. E. L.: Aspects of a Compuational Theory of Human Stereo Vision. DARPA Image Understanding Workshop (1980) 128–149.
2. Biederman, G.: Recognition by Components: A Theory of Human Image Understanding. Psychological Review **94** (1987) 115–147.
3. Hummel, J. E., Biederman, I.: Dynamic Binding in a Neural Network for Shape Recognition. Psychological Review **99** (1992) 480–517.
4. Crick, F. and Koch, C.: The Astonishing Hypothesis. Simon & Schuster (1994).
5. Kandel, E. R., Schwartz, J. H., and Jessel, T. M.: Essentials of Neural Science and Behavior. McGRAW-Hill (1995).
6. Zacks, J. M., Mires, J., Tverski, B. and Hazeltine, E.: Mental Spatial Transformation of Objects and Perspective. Spatial Cognition and Computation **2** (2000) 315–332.
7. Edelman, S.: Constraining the Neural Representation of the Visual World. Trends in Cognitive Sciences **6**(3) (2002) 125–131.
8. Shiffrin, R.: Modeling Memory and Perception. Cognitive Science **27** (2004) 341–378.
9. Jain, R., Kasturi, R. and Schunck, B. G.: Machine Vision. McGRAW-Hill (1995).
10. Shirai, Y. and Inoue, H.: Guiding a Robot by Visual Feedback in Assembly Tasks. Pattern Recognition **5** (1973) 99–108.
11. Sanderson, A. C., Weiss, L. E. and Neuman, C. P.: Dynamic Sensor-based Control of Robots with Visual Feedback. IEEE Transaction on Robotics and Automation **3** (1987) 404–417.
12. Espiau, B., Chaumette and Rives, P.: A New Approach to Visual Servoing in Robotics. IEEE Transaction on Robotics and Automation **8** (1992) 313–326.
13. Holinghurst, N. and Cipolla, R.: Uncalibrated Stereo Hand-Eye Coordination. Fourth British Conference on Machine Vision (1993) 783–790.
14. Hosoda, H. and Asada, M.: Versatile Visual Servoing without Knowledge of True Jacobian. IEEE International Conference on Robots and Systems (1994) 186-193.

15. Nasisi, O. and Carelli, R.: Adaptive Servo Visual Robot Control. Robotics and Autonomous Systems **43** (2003) 51–78.
16. Xie, M.: Fundamentals of Robotics: Linking Perception to Action. World Sceintific (2003).
17. Marr, D.: Vision. Freeman (1982).
18. Ohta, Y. and Kanade, T.: Stereo by Intra- and Inter-baseline Search using Dynamic Programming. IEEE Trans. on PAMI **7** (1985) 139–154.
19. Faugeras, O.: Three-Dimensional Computer Vision. The MIT Press (1993).
20. Seitz, S. and Dyer, C.: Photorealistic Scene Reconstruction by Voxel Coloring. IEEE International Conference on Computer Vision and Pattern Recognition (1997) 1067–1073.
21. Hartley, R. I. and Zisseman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (1998).
22. Zhang, Y. and Xie, M.: New Principle for Passive 3D Scanner. Third World Automation Congress (1998) 1–6.
23. Ziegler, R., Matusik, W., Pfsiter, H., McMillan, L.: 3D Reconstruction Using Labeled Image Regions. Eurographics Symposium on Geometry Processing (2003) 1–12.
24. Zhang, Z. Y.: Determining the Epipolar Geometry and Its Uncertainty: A Review. International Journal of Computer Vision **27** (1998) 161–195.
25. Canny, J. F.: A Computational Approach to Edge Detection. IEEE Trans. on PAMI **8** (1986) 769–798.
26. Kadir, T. Brady, M.: Scale, Saliency and Image Description. International Journal of Computer Vision **45** (2001) 83–105.
27. Guo, C. E., Zhu, S. C., Wu, Y. N.: Modeling Visual Patterns by Integrating Descriptive and Generative Methods. International Journal of Computer Vision **53** (2003) 5–29.
28. Lucas, B. and Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. International Joint Conference on Artificial Intelligence (1981) 674–679.
29. Xie, M.: A Cooperative Strategy for the Matching of Multiple-level Edge Primitives. Image and Vision Computing **13** (1995) 89–99.
30. Hager, G. D., Belhumeur, P. N.: Efficient Region Tracking with Parametric Models of Geometry and Illumination. IEEE Trans. on PAMI **20** (1998) 1025–1039.
31. Nevatia, Y. and Binford, T. O.: Description and Recognition of Curved Objects. Artificial Intelligence **8** (1977) 77–98.
32. Kanade, T.: Model Representation and Control Structures in Image Understanding. IJCAI-5 (1977) 1074–1082.
33. Lowe, D.: Solving for the Parameters of Object Models from Image Descriptions. DARPA Image Understanding Workshop (1980) 121–127.
34. Brooks, R. A.: Model-based Computer Vision. UMI Research Press (1981).
35. Oshima, M. and Shirai, Y.: Object Recognition Using 3-D Information. IEEE transaction on Pattern Analysis and Machine Intelligence **5** (1983) 353–361.
36. Fisher, R. B.: Using Surfaces and Object Models to Recognise Partially Obscured Objects. International Joint Conference on Artificial Intelligence (1983) 989–995.
37. Grimson, W. E. L. and Lozano-Perez, T.: Model-Based Recognition and Localisation From Sparse Range or Tactile Data. International Journal of Robotics Research **3** (1984) 3–35.

38. Horaud, P. and Bolles, R. C.: 3DPO's Strategy for Matching Three-dimensional Objects in Range Data. International Conference on Robotics (1984) 78–85.
39. Faugeras, O. and Hebert, M.: The Representation, Recognition and Locating of 3-D Objects. International Journal of Robotics Research **5** (1986) 27–52.
40. Marshall, A. D. and Martin, R. R.: Computer Vision, Models and Inspection. World Scientific (1992).
41. Chen, C. H., Pau, L. F. and Wang, P. S. P. (Editors): Handbook of Pattern Recognition and Computer Vision. World Scientific (1993).
42. Rohr, K.: Towards Model-based Recognition of Human Movements in Image Sequences. CVGIP **59** (1994) 95–115.
43. Zisserman, A., Forsyth, D. Mundy, J., Rothwell, C., Liu, J. and Pillow N.: 3D Object Recognition Using Invariance. Artificial Intelligence **78** (1995) 239–288.
44. Ullman, S.: High-level Vision. The MIT Press (1996).
45. Grimson, W. E. L.: Introduction: Object Recognition at MIT. International Journal of Computer Vision **21** (1997) 5–8.
46. Tarr, M. J. and Bulthoff, H. H.: Image-based Object Recognition in Man, Monkey and Machine. Cognition **67** (1998) 1–20.
47. Tan, T. N., Sullivan, G. D. and Baker, K. D.: Model-based Localisation and Recognition of Road Vehicles. International Journal of Computer Vision **27** (1998) 5–25.
48. Belongie, S., Malik, J. and Puzicha, J.: Shape Matching and Object Recognition Using Shape Context. IEEE Trans. on PAMI **24** (2002) 509–522.
49. Cyr, C. M. and Kimia, B. B.: A Similarity-based Aspect-Graph Approach to 3D Object Recognition. International Journal of Computer Vision. **57** (2004) 5–22.
50. Binford, T. O.: Visual Perception by Computer. IEEE Systems Science and Cybernetics Conference (1971).
51. Requicha, A. A. G.: Representations for Rigid Solids: Theory, Methods and Alternative Approaches. ACM Computing Surveys **12** (1980) 437–464.
52. Cootes, T. F., Edwards, G. J., Taylor, C. J.: Active Appearance Models. European Conference on Computer Vision (1998) 484–498.
53. Denzler, J.: Knowledge Based Image and Speech Analysis for Service Robots. Workshop on Integration of Speech and Image Understanding, International Conference on Computer Vision (1999).
54. Lungarella, M., Metta, G., Pfeifer, R. and Sandini, G.: Developmental Robotics: A Survey. Connection Science **0** (2004) 1–40.
55. Weng, J.: Developmental Robotics: A Theory and Experiments. International Journal of Humanoid Robotics **2** (2004) 199–236.

What Bipedal Human Locomotion Can Teach Us About Motor Control Synergies for Safe Robotic Locomotion

D.A. Winter

Dept. of Kinesiology, University of Waterloo, Waterloo, Ont. N2L2P6, Canada

1 Introduction

The human body in its daily locomotor tasks faces most of the challenges that a man-made robot encounters in moving over a known or unknown terrain. Over the past 35 years I have analyzed the kinematics and kinetics of human locomotion and have identified several major motor synergies that enable safe and efficient human gait and it is quite possible that similar synergies will assist in the control of robotic walking.

Major Sub-tasks in Human Locomotion Safe and efficient walking requires that all the following subtasks must be accomplished:

1. Generation of mechanical energy to maintain the present forward velocity or to increase the forward velocity of the body [1].
2. Absorption of mechanical energy for shock absorption and stability and to decrease the forward velocity of the body [1].
3. Support of the total body against a collapse against gravitational forces [2].
4. Maintenance of upright posture of the upper body and head [3].
5. Control of foot trajectory to achieve forward progression: safe ground or obstacle clearance and a "soft" foot landing [4].

All these tasks must be accomplished by the torque motors at the joints and the challenge is that any given torque motor may be involved with two or more of those tasks simultaneously and that each torque motor must collaborate with its neighboring motors. The torque motors at the three joints of the lower limb are responsible for about 95% of those subtasks while the muscles crossing the vertebrae are responsible for stabilizing the trunk and head against inertial and gravitational forces acting at the vertebral joints. This paper will confine its focus on the synergies in the motor patterns of the ankle, knee and hip joints.

2 Subtasks 1 and 2 – Generation and Absorption of Mechanical Energy

The standard technique for analyzing the energetics of human locomotion is through a mechanical power analysis at each joint [1]; the joint mechanical power, P_j, in watts is:

$$P_j = M_j T_j W \tag{1}$$

where: M_j is the joint moment (N.m) and T_j is the joint angular velocity (r/s). P_j can positive or negative; if M_j and T_j have the same polarity then P_j is positive (generation); if they have opposite polarities then P_j is negative (absorption). In the human system absorption of energy dissipates as heat; presumably in robotic systems this absorbed energy could be largely recovered by recharging the batteries. Over one gait stride, if the subject is walking at a constant velocity the energy absorbed ≈ energy generated; if they are increasing velocity the energy generated > energy absorbed, if they are decreasing velocity the energy generated < energy absorbed. Figure 1 gives the average power profiles (W/kg body mass) of 10 healthy adult subjects walking their natural cadence (≈ 110 steps/min). These powers are in the sagittal plane and are responsible for the locomotion of the subject in the plane of progression. A short description of the major phases of the power bursts now follows. At the ankle there are two power phases. First there is a low level –ve power, A1, as the leg rotates forward over the foot and the plantarflexors lengthen and

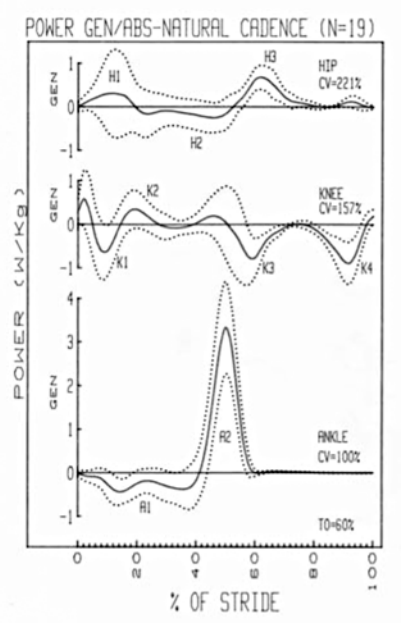


Fig. 1.

absorb energy. Then the largest power phase, A2, called push-off, indicates a large rapid generation of power by these same plantarflexors to accelerate the lower limb upwards and forwards. At 60% stride toe-off (TO) occurs and swing phase begins. At the knee the first power phase is just after weight acceptance, K1, as the quadriceps absorb energy as the knee flexes and is followed by a small generation by these same quadriceps, K2, as they cause the knee to extend adding some potential energy to the body.

Towards the end of stance the large K3 burst indicates a major energy absorption by the quadriceps to control the amount of knee flexion during the powerful push-off phase from the ankle. This represents a significant loss of the push-off energy as plantarflexor energy is dissipated in the quadriceps. Then during the latter half of swing the knee flexors (hamstrings) turn on and absorb energy (K4 burst) to decelerate the rapidly swinging leg and foot for a safe landing at heel contact (HC). At the hip during the first half of stance there is a generation of energy, H1, as the large hip extensor muscles shorten to cause a "push–from–behind".

Then during the latter half of stance the hip flexors lengthen absorbing energy, H2, as the hip continues to extend. Then just before TO these same hip flexors reverse the hip angular velocity causing it to flex and generate energy, H3, to accelerate the lower limb forward into swing. Step length is largely decided by the magnitude of the two generation bursts, A2 and H3, and the two absorption bursts, K3 and K4. Some comment is needed regarding the magnitude of the peak powers involved. A2 is 3.5 W/kg; for an 80 kg adult this is 280 W. In fast walking this power phase approaches 500 W and in running it reaches 1400 W, almost 2 HP!

3 Subtask 3 – Support of the Body Against a Vertical Collapse

Each support phase is 60% of the stride period, with a two 10% double support periods. During these double support periods one limb unloads and the opposite limb takes on weight bearing responsibilities. For efficient movement of the upper body and to maintain minimum changes in its vertical displacement the support knee flexes about 20°. Control of this knee flexion is not only the responsibility of the knee extensors (quadriceps) but also the ankle and hip extensors. Figure 2 demonstrates this three joint task and the resultant indeterminacy of the total motor control. For a given knee angle history there are an infinite number of possible combinations of M_a, M_k and M_h. Thus in repeat walking trials of the same subject when the variability in the knee angle is very small ($\approx \pm 1°$) the variability of the joint moments as is demonstrated in Fig. 3 is quite high [2]. Here repeat trials over 9 days are plotted as an ensemble average. Over the stance period the coefficient of variation (CV) of M_h is 68% and M_k is 60%. However, when we add all three joint moments (with extensor as +ve) to get the support moment the CV drops

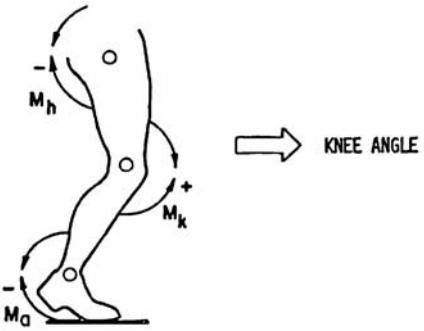


Fig. 2.

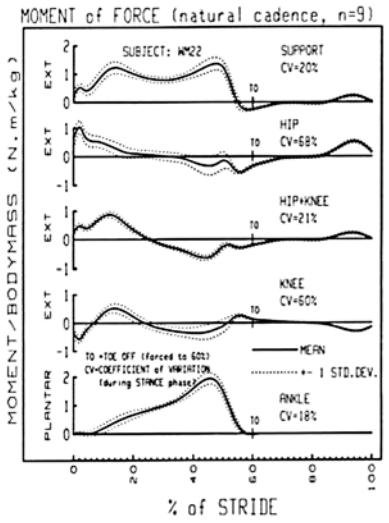


Fig. 3.

to 21%. Subsequent analysis of the hip and knee moment profiles showed an almost one-for-one stride-to-stride trade-off during stance. For example, on two strides the knee would be 5 N.m more extensor while the hip was 5 N.m less extensor. For this subject a covariance analysis quantifies the exact extent of the trade-offs between the adjacent joints; see Fig. 4 for the results for this day-to-day subject along with 10 repeat trials done on the same subject five minutes apart. The variance of M_h profiles was $\Phi_h^2 = 15.9$ (N.m)^2, for the knee $\Phi_k^2 = 12.6$ (N.m)^2 and for the ankle $\Phi_a^2 = 10.5$ (N.m)^2. For the $M_k + M_\mathrm{h}$ profile $\Phi_{h+k}^2 = 6.9$ (N.m)^2 and for $M_a + M_k$ profile $\Phi_{a+k}^2 = 8.1$ (N.m)^2. The covariance between the hip and knee, Φ_{hk}^2, is:

$$\Phi_{hk}^2 = \Phi_h^2 + \Phi_k^2 - \Phi_{h+k}^2 \tag{2}$$

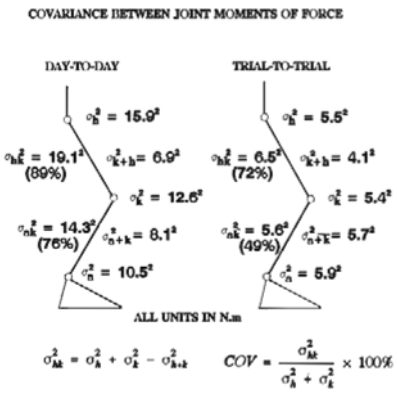


Fig. 4.

The term Φ^2_{hk} can be expressed as a percentage of the maximum possible covariance, which would be 100% if $\Phi^2_{h+k} = 0$, and that would mean the variability in the knee was exactly equal and opposite that at the hip. The minimum Φ^2_{hk} could be would be 0 if $\Phi^2_{h+k} = \Phi^2_h + \Phi^2_k$ and that would indicate that the knee and hip were varying totally independent of each other. The percent covariance, COV, is:

$$COV = \Phi^2_{hk}/(\Phi^2_h + \Phi^2_k) \times 100\% \tag{3}$$

Figure 4 reports these COVs, for the day-to day trials the hip/knee COV = 89% and the knee/ankle COV = 76%. For the repeat trials minutes apart the COVs drop to 72% and 49% respectively. Thus when we sum $M_s = M_a + M_k + M_h$ as shown in Fig. 3 the cancellations between the hip and knee and between the knee and ankle result in the considerably reduced variability in the support moment, M_s. As indicated in Fig. 2 the predicted variability in the individual joint moments is possible as long as the sum of the three remains reasonably constant and this can be worked to advantage to satisfy other subtasks, mainly that of upper body balance.

4 Subtask 4 – Maintenance of Upright Posture of the Upper Body

The upper body can be considered an inverted pendulum pivoting about the hip joint in both the sagittal and frontal planes, see Fig. 5. The hip moment is the prime motor to balance the upper body against the gravitational and inertial forces acting on it. The dynamic equilibrium equating the hip moment, M_j, as derived in Fig. 5 with these forces is:

$$M_j = mg(x_0 - x_j) + m\,\ddot{y}_j(x_0 - x_j) - m\,\ddot{x}_j(y_0 - y_j) + I_0\alpha \tag{4}$$

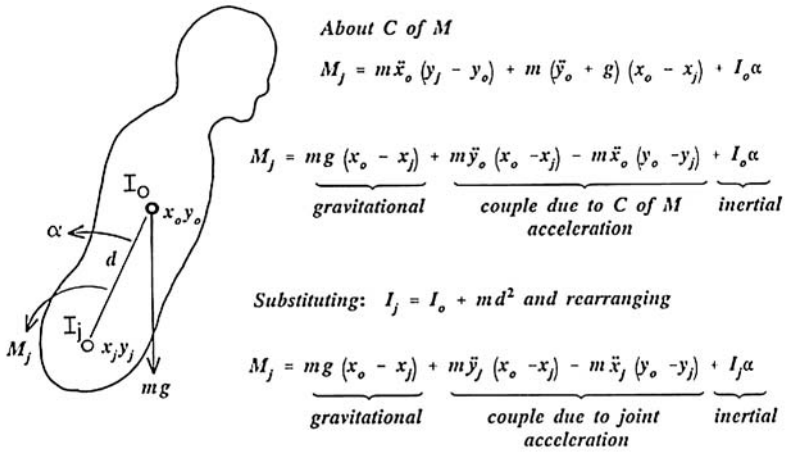


Fig. 5.

The first term on the right is the gravitational load, the second and third terms are the couples due to the hip joint acceleration and the last the inertial load of the inverted pendulum. In normal human walking with the upper body erect in the sagittal plane the gravitational load ≈ 0 and we also wish to keep α as low as possible (as the human motor system does reasonably well in this plane). Thus it was found during single support [3] that (4) simplified to:

$$M_j \approx m\ddot{y}_j (x_0 - x_j) - m\ddot{x}_j (y_0 - y_j) \tag{5}$$

The sum of the terms on the right was referred to as the unbalancing moment due to perturbing hip accelerations while M_j was referred to as the balancing moment. In this plane the hip extensors during the first half of stance and the hip flexors during the latter half of stance accomplish this task as a top priority. Thus α is kept as low as possible as the hip motors in combination with the spinal muscles (not discussed here) reduce the head acceleration to less than $0.5\,\mathrm{m/s^2}$. Thus the platform for the visual system is well stabilized in spite of the large linear accelerations of the hip joints themselves.

In the frontal plane this same dynamic equilibrium (4) applies but the gravitational term is quite significant because the centre of mass of the upper body lies medial of the hip joint [3]. Again during single support there is a moderate medial acceleration of the hip joint that creates a couple opposite to the gravitational moment. The hip abductors are now active to make up the difference and keep the frontal plane α to a low level. The net result is an almost erect upper body with a stable head platform.

5 Subtask 5 – Safe Trajectory of the Foot during Swing

The control of the foot during swing is a precise and multi-factorial task because the foot is the last segment of a multi-segment chain. The chain starts with the stance foot up to the pelvis, across the pelvis and down the swing limb; this consists of seven segments and twelve major joint angular degrees of freedom as depicted in Fig. 6. The length of this chain is about 2.3 m in an adult while toe clearance averages 1.3 ± 0.6 cm during mid swing when the horizontal velocity of the toe is 4.5 m/s. Six of the joint angles in the link chain can control this toe clearance, the most sensitive is stance hip abduction which controls the pelvic frontal plane angle and therefore the height of the swing hip above the ground [4].

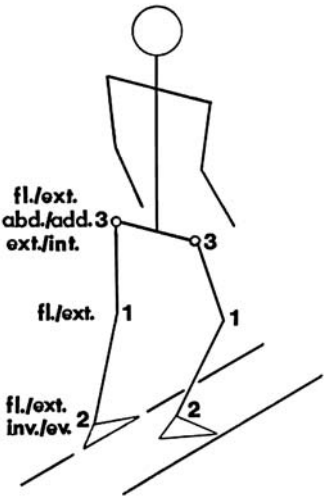


Fig. 6.

With the pelvis held almost horizontal the stance and swing muscles are now free to carry out subtasks 1, 2, and 3. The heel trajectory at heel contact has a vertical velocity ≈ 0 but with horizontal velocity, averaging 0.87 m/s (often described as a skid velocity). In a robot this skid velocity is probably excessive; the final velocity of the foot can be reduced by increasing the absorption K4 burst (see Fig. 1) by the knee flexors, but this will also reduce the step length. Figure 7 is a "birds-eye" view of the trajectory of the foot placement positions showing the normal placement positions along with two undesired positions, "tandem" and "wide". The desired trajectory of the foot has the body's C of G passing forward just inside border of the stance foot; in this position the C of P is just to the right or left of the C of G and causes the body (which acts as an "inverted pendulum") to accelerate medially towards the future position of the next foot. If a wide stance were achieved the medial

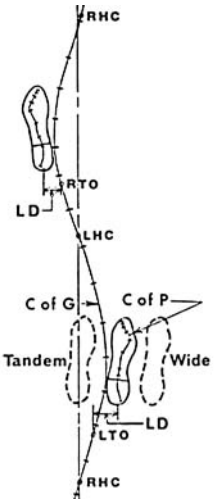


Fig. 7.

acceleration could be too high so as to set up an unstable sequence, and if a tandem position were achieved the acceleration could be the wrong polarity and accelerate the body sideways away from the line of progression and start a lateral fall.

References

1. Winter DA (1983) Biomechanical motor patterns in normal walking. J. Motor Behav. 15: 302–330
2. Winter DA (1984) Kinematic and kinetic patterns in human gait: Variability and compensating effects. Hum. Movement Sci. 3: 51–76
3. Winter DA, Ruder GK, MacKinnon CD (1990) Control of Upper Body Balance during Walking. In: Multiple Muscle Systems: Biomechanics and Movement Organization. J. Winters & S.Woo (eds.), pp. 680–693, Springer-Verlag, N.Y.
4. Winter DA (1992) Foot trajectory in human gait – a precise and multi-factorial control task. Physiotherapy, 72: 45–56

Problems of Scale for Walking and Climbing Animals

R. McNeill Alexander

School of Biology, University of Leeds, Leeds LS2 9JT, UK.
r.m.alexander@leeds.ac.uk

1 Introduction

This paper is about the design of walking, running and climbing animals of different sizes. It shows how body proportions and patterns of movement tend to differ, between similar animals of different sizes. It is largely about mammals ranging in mass from about 3g (shrews) to 3 tonnes (elephants), but includes some discussion of birds, lizards and frogs. Information about scale effects for other animal groups and other modes of locomotion can be found in [3].

2 Geometric Similarity

The simplest imaginable scaling rule for animals of different sizes would be geometric similarity. This would imply that the structure of an animal would become identical to that of a similar but larger animal, if all its linear dimensions could be multiplied by the same factor λ. Areas on the larger animal would then be λ^2 times the corresponding areas on the small one, and volumes would be λ^3 times the corresponding volumes. Because animals generally have more-or-less equal mass densities, similar animals of different sizes would have body masses proportional to (linear dimensions)^3, and linear dimensions proportional to (body mass)$^{0.33}$.

Some animals, in some respects, scale close to the predictions of geometric similarity. For example, whales of masses 30kg to 100 tonnes have lengths proportional to (body mass)$^{0.34}$ [15]. Alexander et al. [5] found that terrestrial mammals ranging from shrews to elephants had leg bone lengths about proportional to (body mass)$^{0.35}$. This rule worked well for insectivores, rodents, carnivores and primates, but surprisingly not for bovids (antelopes etc), which had leg bone lengths about proportional to (body mass)$^{0.25}$, as McMahon had previously shown [21]. Consequently a gazelle is not simply a scale model of a buffalo, but has quite different proportions. Most of the largest animals in

our sample were bovids, and later research has shown that we were wrong to regard the bovids as scaling differently from other large mammals. Garcia and da Silva [16] derived separate allometric equations for mammals of less than, and more than, 50kg. Bone lengths were about proportional to (body mass)$^{0.37}$ for mammals of less than 50kg, and (body mass)$^{0.26}$ for larger mammals. We will return to this observation, but first we will look at the scaling of patterns of movement.

3 Dynamic Similarity

Just as geometric similarity would be the simplest scaling rule for body structure, dynamic similarity would be the simplest for patterns of movement. These patterns are characterised by linear dimensions (for example stride length, leg length), times (for example, the stride period) and forces (for example, the peak force on a foot). One pattern of motion is dynamically similar to another if they could be made identical by multiplying all linear dimensions by a single factor λ, all times by a single factor τ, and all forces by a single factor ϕ.

In this paper we are concerned with walking, running and climbing, motions in which gravity is important. Whatever the size of the animal, it will fall, if unsupported, with acceleration g. Accelerations are proportional to λ/τ^2, or to $(\lambda/\tau)^2/\lambda$. Hence, in dynamically similar gaits, ν^2/h must be proportional to g, where ν is a velocity characteristic of the motion (for example, running speed) and h a linear dimension characteristic of the motion (for example, leg length). In other words, animals of different sizes cannot move in dynamically similar fashion, except when the ratio of their speeds is such as to give them equal Froude numbers ν^2/gh.

Alexander [2] and Alexander & Jayes [4] hypothesised that similar animals of different sizes would tend to run in dynamically similar fashion, when travelling with equal Froude numbers. This seemed likely, because if an animal adjusts its gait to minimise the work required for running, a geometrically similar animal running with the same Froude number must run in dynamically similar fashion, to minimise its work requirement. We defined our Froude number in terms of the running speed ν and the height h of the hip joint from the ground. Some other authors have used lower leg length for h, an equally valid choice.

Perfect dynamic similarity is possible, only if the animals are geometrically similar. Even closely related animals of different sizes fail to satisfy this condition, as we noted in our comparison of a gazelle with a buffalo. The deviation from geometric similarity is less marked in some other comparisons, for example between a lion and a domestic cat, but the similarity is never perfect. Consequently, though animals of different sizes may approach dynamic similarity in their gaits, they cannot achieve perfect similarity.

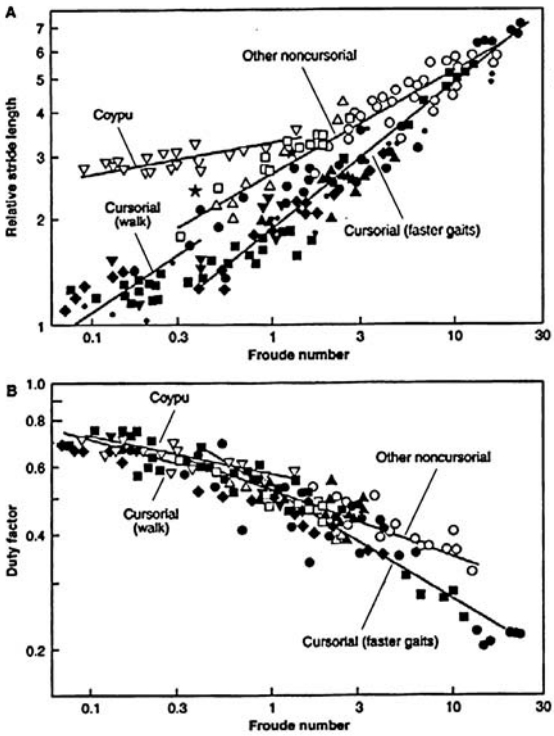


Fig. 1. Stride lengths and duty factors of mammals, related to speed. In (**a**) relative stride length and in (**b**) duty factor is plotted against Froude number. Different symbols are used for different species. Hollow symbols refer to non-cursorial species (rodents and ferret) and filled symbols to cursorial species (dog, cat, sheep, camel, horse and two species of rhinoceros). From [4]

Our dynamic similarity hypothesis implied that similar animals of different sizes should change gaits at equal Froude numbers. We found this to be approximately true. Mammals ranging in size from rats to rhinoceros walked at Froude numbers below 0.5, trotted at intermediate Froude numbers, and galloped at Froude numbers above 2.5. The hypothesis also implied that similar animals, travelling with equal Froude numbers, should have equal duty factors (the fraction of the stride duration, for which each foot is on the ground). Figure 1B shows that this is roughly true. Further, at equal Froude numbers, relative stride length (stride length/leg length) should be the same. Figure 1A shows that this prediction is not so successful.

There is a marked difference of posture between the groups of mammals called "cursorial" and "non-cursorial" in Fig. 1. Cursorial mammals (for example, dogs and horses) run on relatively straight legs. Non-cursorial mammals (for example, mice and ferrets) run in a crouched posture, with their legs

strongly bent. Domestic cats (about 2.5 kg) are among the smallest cursorial mammals, and coypu (about 7 kg) are among the largest non-cursorial ones. Figure 1A shows that at equal Froude numbers, non-cursorial mammals use longer relative stride lengths than cursorial ones. Alexander and Maloiy [6] found that primates take still longer strides. Within each of these three groups of mammals, however, the dynamic similarity hypothesis is fairly successful. It also succeeds quite well for running birds. At the same Froude number, however, ostriches and other ratites use shorter relative stride lengths than smaller birds, a striking parallel with the difference between cursorial and non-cursorial mammals [17].

4 Stress Similarity

The mean force exerted by the feet on the ground, averaged over a stride, must equal body weight. The faster an animal runs, the lower the duty factor (Fig. 1B), and the larger the force each foot must exert while on the ground, so fast running requires strong legs. For geometrically similar animals, weight is proportional to the cube of linear dimensions, but strengths of bones and muscles are proportional only to the square of linear dimensions. Thus larger animals cannot be expected exert as large forces, relative to body weight, as small ones, and are expected to have lower maximum Froude numbers. In an extreme case, an excessively large animal would be unable to support its own weight even when standing still.

Biewener [10] showed that this effect is largely countered by systematic size-related changes of posture. Larger mammals stand and run with their legs straighter than smaller mammals. This reduces the forces required of muscles and the bending moments acting on bones, when a given force acts on the foot. In Fig. 2, the ratio (foot force/muscle force) is r/R. This is the effective mechanical advantage of the muscle.

$$F_{\mathrm{m}} = (R/r)F_{\mathrm{g}} \tag{1}$$

where F_{m} is the force exerted by the muscle and F_{g} is the force on the ground. The stress in the muscle is $(R/r)F_{\mathrm{g}}/A_{\mathrm{m}}$, where A_{m} is the muscle's physiological cross sectional area. If F_{g} is proportional to body mass and A_{m} to (body mass)$^{0.67}$ (as required for geometric similarity), muscle stress can be equalised for animals of different sizes by making r/R proportional to (body mass)$^{0.33}$. In the previous section I distinguished between large, straight-legged cursorial mammals (with higher values of r/R), and smaller non-cursorial mammals with bent-legged postures (lower r/R). Biewener [10] showed that there is actually a gradation of posture running right through the mammalian size range. He found that for the principal leg joints of mammals ranging from mice to horses, r/R was approximately proportional to (body mass)$^{0.25}$.

Peak stresses in leg bones during running and jumping have been calculated for mammals ranging in size from mice to elephants. In some cases the

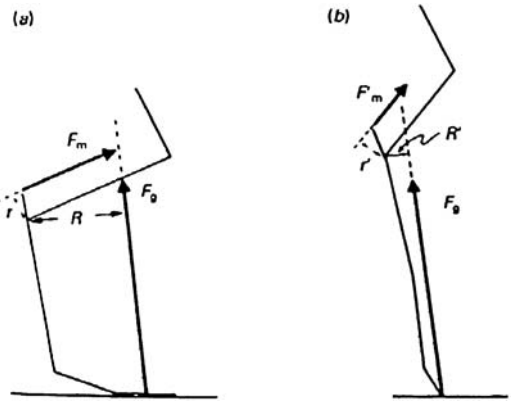


Fig. 2. Diagrams of the skeleton of the fore legs of (**a**) a 0.15 kg ground squirrel and (**b**) a 270 kg horse, at corresponding stages of their strides. The *thick arrows* represent the force F_g exerted by the ground on the foot, and the force F_m exerted by the triceps muscle (the extensor muscle of the elbow). R and r are the moment arms referred to in the text. From A. A. Biewener (1983) *Journal of Experimental Biology* **105**, 147–171

stresses have been calculated from records of ground forces, or from kinematic analysis of film. In others they have been calculated from the output of strain gauges surgically implanted on the bones. Biewener [11] showed that, largely due to the differences in effective mechanical advantage, similar stresses act in the leg bones of mammals of all sizes. The strengths of muscles and bones seem to increase a little faster, with increasing body size, than if geometric similarity prevailed, but these changes are much less important than the straightening of the legs and the resulting increase of effective mechanical advantage.

The scaling of r/R, and other deviations from geometric similarity, are not quite sufficient to enable large mammals to achieve the same maximum values of (foot force/body weight) as small ones. Elephants cannot leap like gazelles.

5 Elastic Similarity

If animals ran on frictionless wheels, no work would be needed for running at constant speed over level ground. By far the major part of the energy cost of running is due to legs doing work at one stage of a step (increasing the body's kinetic and gravitational potential energy), and acting like brakes (degrading mechanical energy to heat) at another. Some of the energy that would otherwise be needed is saved in walking by the principle of the pendulum, by swapping energy back and forth between the kinetic and potential forms. This principle is ineffective at Froude numbers greater than 1, because the

rate at which potential energy can be lost is limited by the acceleration of gravity [1]. At higher speeds mammals run, saving energy by a different principle: much of the kinetic and gravitational potential energy lost at one stage of the step is stored as elastic strain energy in tendons, and restored by their elastic recoil [7, 12, 14]. Running mammals bounce along like rubber balls.

Elastically similar structures [20] deform elastically in geometrically similar fashion, under their own weight. Animals of different sizes running in dynamically similar fashion exert forces proportional to their masses, and undergo vertical displacements proportional to the lengths of their legs. For tendon elasticity to contribute similarly to the energetics of running, the animals should be elastically similar. Bullimore and Burn [13] pointed out that elastic similarity is incompatible with strict dynamic similarity, because Young's modulus for tendon is about the same for mammals of all sizes. They discussed possible compromises.

Tendons are in series with muscles, so the force in a tendon is the same as the force in its muscle, $(R/r)F_g$. If the length of the tendon is l_t and its cross sectional area A_t, its elastic extension will be proportional to $(R/r)F_g l_t/A_t$. The corresponding vertical displacement of the foot will be (R/r) times this, $(R/r)^2 F_g l_t/A_t$. For elastic similarity, this vertical displacement must be proportional to leg length h when F_g is proportional to body mass. Let l_t and h be proportional to (body mass)$^{0.33}$, and A_t to (body mass)$^{0.67}$, as predicted for geometric similarity. Then elastic similarity requires r/R to be proportional to (body mass)$^{0.17}$. The observed exponent is about 0.25, as we have already seen, intermediate between the exponents predicted for elastic similarity and for stress similarity.

6 Climbing

Geckos, tree frogs, squirrels and some other vertebrate animals can climb vertical surfaces. Running requires a mean force equal to body weight, exerted at right angles to the long axis of the body. In contrast, vertical climbing requires a force equal to body weight, parallel to the long axis. In addition, the animal must exert forces perpendicular to the vertical surface to balance the moments on its body. If it is facing upwards, it must pull on the surface with its fore feet and push on it with its hind feet. Pulling is made possible either by mechanical anchors such as the claws of squirrels, or by adhesion as in geckos and tree frogs.

Claws are effective only on surfaces such as bark, which are soft enough for them to penetrate. The most obviously effective attachment devices for smooth, impenetrable surfaces are suckers that exploit atmospheric pressure. They have the merit of being able to supply tangential (frictional) as well as normal forces, if the pressure inside is low enough to press the sucker rim firmly against the surface. Octopus have suckers, but I do not know of any terrestrial animal that uses suckers for climbing.

The toe pads of tree frogs look like suckers, but have a different means of attachment. A very thin layer of liquid fills the space under the pad. Surface tension in the meniscus at the perimeter of the pad reduces the pressure under the pad. This is the principle of capillary adhesion. Hanna and Barnes [18] summarised the evidence that capillary adhesion, rather than Stefan adhesion due to the viscosity of the liquid, is the dominant provider of normal forces. Capillary adhesion cannot supply significant tangential forces, but viscosity may. Alternatively, as the liquid film under the pad may be as little as 50 nm thick [9], there may be friction between the relatively rough sole of the pad, and the surface being climbed.

Geckos, unlike tree frogs, have dry feet. A carpet-like pile of fine setae makes such close contact with the surfaces they climb, that van der Waals forces suffice to support the animals' weight [8]. Experiments on individual setae showed that a seta could adhere with a force of 0.2 mN. If every one of a 50 g gecko's 6.5 million setae exerted that force, the total would be 1300 N (133 kg wt)!

Both capillary adhesion by a tree frog's toe pads, and van der Waals adhesion by a gecko's setae, are expected to be proportional to foot area and so, for geometrically similar animals, to (body mass)$^{0.67}$. The forces required for climbing are proportional to body mass. This suggests that climbing should be more difficult for larger animals. It may possibly explain why no tree frogs are as large as bullfrogs, but cannot explain why geckos are limited to about 70 g. The huge forces estimated above are unattainable, but Irschick et al. [19] showed that even large geckos can run up walls carrying lead weights equal to their body weight.

References

1. Alexander RMcN (1976) Mechanics of bipedal locomotion. In P. S. Davies (edit.) Perspectives in Experimental Biology **1**, 493–504. Pergamon, Oxford
2. Alexander RMcN (1976) Estimates of speeds of dinosaurs. Nature **261**, 129–130
3. Alexander RMcN (2003) Principles of Animal Locomotion. Princeton: Princeton University Press
4. Alexander RMcN, Jayes AS (1983) A dynamic similarity hypothesis for the gaits of quadrupedal mammals. Journal of Zoology **201**, 135–152
5. Alexander RMcN, Jayes AS, Maloiy GMO, Wathuta EM (1979) Allometry of the limb bones of mammals from shrews (Sorex) to elephant (Loxodonta). Journal of Zoology **190**, 155–192
6. Alexander RMcN, Maloiy GMO (1984) Stride lengths and stride frequencies of primates. Journal of Zoology **202**, 577–582
7. Alexander RMcN, Vernon A (1975) Mechanics of hopping by kangaroos (Macropodidae). Journal of Zoology **177**, 265–303
8. Autumn K, Peattie AM (2002) Mechanisms of adhesion in geckos. Integrative and Comparative Biology **42**, 1081–1090
9. Barnes WJP, Riehle MO, Smith JM, Federle W (2004) Wet adhesion in tree frogs. Comparative Biochemistry and Physiology **137A**, S87 only

10. Biewener AA (1989) Scaling body support in mammals: limb posture and muscle mechanics. Science **245**, 45–48
11. Biewener AA (1990) Biomechanics of mammalian terrestrial locomotion. Science **250**, 1097–1103
12. Biewener AA, Konieczynski DD, Baudinette RV (1998) In vivo muscle force-length behavior during steady-speed hopping in tammar wallabies. Journal of Experimental Biology **201**, 1681–1694
13. Bullimore SR, Burn JF (2004) Distorting limb design for dynamically similar locomotion. Proceedings of the Royal Society B **271**, 285-289
14. Cavagna GA, Heglund NC, Taylor RC (1977) Mechanical work in terrestrial locomotion: two basic mechanisms for minimizing energy expenditure. American Journal of Physiology **233**, R243–R261
15. Economos AC (1983) Elastic and/or geometric similarity in mammalian design? Journal of Theoretical Biology **103**, 167–172
16. Garcia GJM, da Silva JKL (2004) On the scaling of mammalian long bones. Journal of Experimental Biology **207**, 1577–1584
17. Gatesy SM, Biewener AA (1991) Bipedal locomotion: effects of speed, size and limb posture in birds and humans. Journal of Zoology **224**, 127–147
18. Hanna G, Barnes WJP (1991) Adhesion and detachment of the toe pads of tree frogs. Journal of Experimental Biology **155**, 103–125
19. Irschick DJ, Vanhooydonck B, Herrel A, Andronescu A (2003) Effects of loading and size on maximum power output and gait characteristics in geckos. Journal of Experimental Biology **206**, 3923–3934
20. McMahon TA (1973) Size and shape in biology. Science **179**, 1201–1204
21. McMahon TA (1975) Allometry and biomechanics: limb bones in adult ungulates. American Naturalist **109**, 547–563

Biologically Motivated Control of Walking Machines

R. Dillmann, J. Albiez, B. Gaßmann, and T. Kerscher

Forschungszentrum Informatik, Haid-und-Neu-Str. 10–14, 76131 Karlsruhe, Germany
dillmann@ira.uka.de

1 Introduction

Walking robots represent a field of increasing research activities in the last years. Especially the ability of walking machines to adapt to unstructured terrain and the resulting requirements to the control architecture are emphasized by the researchers. These efforts can be separated into two different approaches, one being the classical engineering approach using and refining the classical methods of feed-back control structures and dynamic modelling to control the robot, e.g. [1] and [2]. The other way is to adopt as much from biological paragons for locomotion as possible regarding both mechanical design and control architecture, [3] and [4].

Biomechanical research of the last years has identified several key elements being used in nature for adapting locomotion. These range from the geometrical structure of legs [5] and dynamic properties of muscles ([6]) to neural networks used for walking by insects [7] and [8]. The results of this research imply a lot of benefits in case of a transfer of these principles to legged robots. Due to the high complexity of real walking machines and the impracticality of mimicking especially nature's actuators and sensors, up to now only some of the ideas have been transferred to the control architectures and the design of real robots.

In this paper we give an introduction to the results we gained in the are of biologically inspired walking machines over the last years.

2 Pneumatic Muscles as Acutators

Commonly walking and climbing robots are propelled using electrical motors combined with special gears and simple damper mechanisms. During locomotion of these machines, unforeseen ground or obstacle contact of a foot could lead to a strong disturbance of the movement or damage parts of the mechanics. Even if control concepts with active compliance are implemented, it is very

hard to cover these impact problems, especially in cases of fast movements. A second motivation for the use of other kind of actuators for walking machines is the possibility to redesign the locomotion apparatus of animals [5]. In both cases, soft actuators like artificial muscles seem to solve the actuator problem.

Pneumatic muscles as actuators offer several advantages due to their performance weight relation or to the inherent passive compliance of this type of actuators. Passive compliance is an important component for the control of locomotion in rough terrain to absorb the power stroke energy. Such passive compliance existing within biological muscles, which mammals use actively for locomotion [9]. With fluidic muscles, it is possible to copy this behaviour with a technical system because of the elasticity of the fluidic muscles similar to biological muscles. In addition they reach a relatively high velocity for the contraction and a high-energy efficiency [10]. Further, they possess a high power to weight relation and a long life span.

2.1 Fluidic Muscles as Actuators for Walking Machines

The fluidic muscles MAS-20 produced by the company FESTO are selected as actuators for the machines developed at the FZI. This muscles are based on the well known McKibben principle which is described in [9]. The static correlation between contractive force, contraction and pressure of the muscle is shown in Fig. 1. The following equation describing the static force of the muscle:

$$F_{stat.}(p,\kappa) = \mu \cdot (\pi r_0^2) \cdot p \cdot (a \cdot (1 - (a_\varepsilon \cdot e^{-p} - b_\varepsilon) \cdot \kappa)^2 - b) \tag{1}$$

with the contraction κ, the pressure p, the initial muscle radius r_0, the initial fibre angle α_0, the muscle specific parameters [10] a and b, the initial length of the muscle l_0, the scalar factor μ and the parameters a_ε and b_ε which has been found using the least squares method. Figure 1(a) presents a comparison between this model and the real muscle curve.

(a) (b)

Fig. 1. (**a**) Comparison between muscle model and the real muscle curve (**b**) The MAS-20 fluidic muscle, upper part fully contracted, lower part initial length

Due to the fact, that the fluidic muscles only produce tractive forces it is necessary to use at least two muscles for one joint. This to muscles work antagonistic. With the help of a sophisticated joint controller it is possible to control not only the joint position but also the stiffiness of the joint [9].

2.2 AirInsect

AirInsect is a new biologically inspired six-legged walking machine (see Fig. 2). As model from nature the stick insect, Carausius Morosus, was consulted. The stick insect has been chosen because it has a optimize geometry of the body and the legs for the locomotion with six legs in rough terrain. The different distance between the leg pairs and also the length of the leg segments are very important characteristic for this.

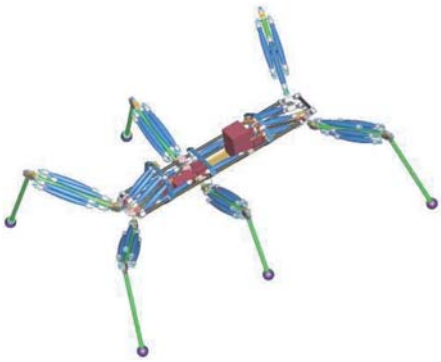


Fig. 2. Construction drawing of the whole machine airInsect

The new approach of AirInsect compared with other biologically motivated walking machines are developed based on the model of the stick insect like MAX, Tarry or LAURON is that the arrangement of the legs and the design of the joints were built as an exact imitation of what can be found in nature [11]. So the body geometry of the stick insect and the different leg distances and segment lengths of the legs are taken over for the built-up of airInsect.

However, the mobility and flexibility of the stick insect are still unequalled with technical machines up to now. The aim of airInsect is to examine the advantages of a more exact reproduction of the stick insect geometry for a walking machine.

Based on the experiences with the 2000/2001 through pneumatic muscles driven six-legged walking machine AirBug developed at the FZI AirInsect is developed as an improved robot. In addition with the construction of the legs very large attention is put on a protection against environmental influences for example splash-water, dust, etc.

The electronic components of AirInsect essentially consist of industrial computer and six microcontrollers. The machine is equipped with pressure, orientation, force and angle sensors, which supply data over the current machine condition. The joint control orients itself to the one from AirBug and PANTER [9]. For the upper control layers the control algorithms from the six-legged machine LAURON are used [12].

2.3 PANTER

In the last years many biologically inspired walking machines were developed. Examples are the mammal-like BISAM [13], the primate-like Salford gorilla [14] or the stick-insect like robot LAURON [12]. There are also a lot of other quadruped robots like TITAN VII/VIII [15], Patrush [16] and Geo II [17]. In common to all these robots is that they only operate at low speed. Faster moving legged robots like the Raibert-hopper [18], the Scout II [19] or Tekken [20] operate purely in the fast running mode, without the ability to accomplish controlled precise movements. In nature mammals demonstrate every day, that it is possible to run fast, jump and do precise motions. Therefore, it is obvious to have a closer look at the principles used in nature and to evaluate where and how they can be adapted to robotics.

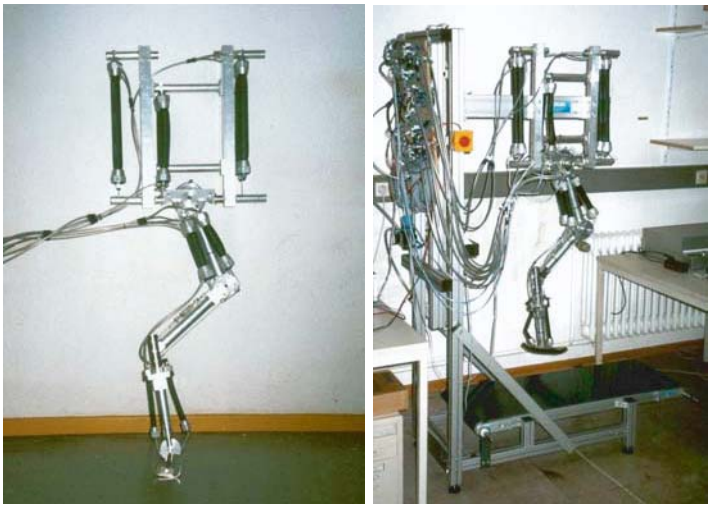
Concerning the behaviour of mammalian legs during running biomechanical research has shown that the morphology of the leg and the spring-damper characteristic of the muscles are key elements of quick locomotion. Evolution has optimized the walking apparatus of mammals to use self stabilization cycles while running and in case of disturbances adapting the spring-load of the leg to stay in these cycles [9].

The idea to adapt the above-described biological principles for a running robot was the starting point of our research. The construction of a prototypical leg and its closed loop control has been developed. This leg is part of the PANTER – Pneumatically Actuated dyNamically sTable quadrupEd Robot.

In Fig. 3(a) the leg for the PANTER-robot is shown. The leg has a weight of 4.6 kg. The shoulder fastener with the muscles for the α- and β-joints has a weight of 3.5 kg. The whole leg weights 8.1 kg. The muscles contribute about 40 percent of the whole weight.

3 Behaviour Networks for Adaptive Locomotion

There have been several approaches to use behaviour based control in mobile robots over the last few years. The main area of application have been wheel driven, kinematically extremely simple robots, with the main focus on the robot's task e.g. navigation and group behaviour (see [21, 22] and [23]). Therefore these architectures concentrate on the coordination of a set of high level behaviours operating mostly on the same semantic level and producing



(a) (b)

Fig. 3. PANTER leg: (**a**) leg with shoulder fastener and muscles (**b**) test-rig

abstract commands for the robot's hardware. Since the coordination of behaviours is crucial part in behaviour based robotics a lot of work has been done there. Examples include comfort zones [24] or case based reasoning [25]; a good overview on behaviour coordination can be found in [26].

There have only been a few attempts to use behaviour based architectures on the lower levels of the control architecture for kinematically more complex robots like walking machines. The best known and most successful is the subsumption architecture [27, 28] used on several hexapods. A more biological inspired approach for a lobster-like robot is proposed in [3]. But there are several drawbacks to these architectures, among them a general tendency towards scalability problems, weaknesses when adding new behaviours or trying reusing existing ones and in most cases a highly problem specific approach (see [29]).

3.1 Motivation

When considering the insights gained through PET and EEG scans and spinal cord activity plots of animals performing certain tasks ([30] and [6]), as well as the problems when dealing with real sensor information and highly complex robots, the following key aspects can be identified:

- A certain action of an animal always creates activity in the same area of the animal's brain or its spinal cord shows an example of the activity pattern in a human brain).
- Such an active area can result in the stimulation of further regions as well as inhibit activity in others.

- Even though the classical approach to robot control has difficulties in handling the complexity of the whole system, these established methods should be applied to solve simpler sub-problems.
- As hierarchical systems have been approved in robotics as well as in nature it is advisable to use some kind of levelled system with an increasing degree of abstraction regarding sensor data and motor signals.

Taking these observation into consideration, we designed a control architecture consisting of a hierarchical network of behaviours ([31] and [32]). Each behaviour or reflex[a] is developed using methods of classical control system design or artificial intelligence. Only the interaction of the behaviours and their placement in the network will result in the desired actions of the overall system.

3.2 Behaviours

A behaviour or reflex $\mathcal{B}$ in the sense of this architecture is a functional unit which generates an output vector $\mathbf{u}$ using an input vector $\mathbf{e}$ and an activation ι according to a transfer function $F(\mathbf{e})$. Additionally a target rating criterion $r(\mathbf{e})$ and a behaviour activity a is calculated. Mathematically this can be combined to the 6-tuple

$$\mathcal{B} = (\mathbf{e}, \mathbf{u}, \iota, F, r, a)$$

The transfer function F is defined as in (2) for an input dimension of n and an output dimension of m. It implements the fundamental action being performed by the behaviour. The output vector $\mathbf{u}$ is generated in two steps. First an unmodified output with respect to the input vector $\mathbf{e}$ is produced. In the second step this output is modified according to the activation ι. Generally speaking a ι of 0 means that the behaviour will generate no output at all, one of 1 will lead to the full output. For simple behaviours, for example posture control reflexes implemented by a classical controller, the output is scaled between 0 and 1. In more deliberative behaviours, like the swing or a gait behaviour, ι could express an increasing tendency to perform an action.

$$F : \Re^n \times [0; 1] \rightarrow \Re^m; \quad F(\mathbf{e}, \iota) = \mathbf{u} \tag{2}$$

This activation mechanism allows to ensure the robot's safety to a certain degree by activating only a defined set of behaviours and enables the usage of the behaviour as an abstract actor by other higher level behaviours.

While the formal definition above is an abstract view of the behaviour, the actual implementation method can vary from simple feed-forward controllers up to more complicated systems like finite state automata or algorithms used in artificial intelligence.

[a] A reflex refers to a simple behaviour close to the hardware thus being more reactive than deliberative

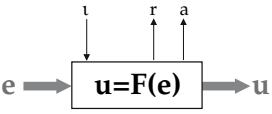


Fig. 4. Behaviour design

s mentioned above each behaviour generates a target rating r (3) evaluating how far the actual state of the robot matches the aspired goal of the behaviour. For this estimation the input vector $\mathbf{e}$ consisting of sensor information from the robot and outputs of other behaviours is used. A value of 0 indicates that the robot's state matches the behaviour's goal, a value of 1 that it does not. It is constantly updated even if the behaviour is deactivated and generates no output.

$$r : \Re^n \rightarrow [0;1]; r(\mathbf{e}) = r \tag{3}$$

For monitoring reasons it is desirable to have some visualisation of the behaviour's activity. But even more importantly, the activity is used as feedback information for other behaviours and for behaviour coordination. The activity a is defined as in (4). Colloquial a can be interpreted as indication on the behaviour's effort in transferring the robot to a state achieving its goal.

$$a : \Re^m \rightarrow [0;1] : a(\mathbf{u}) \sim \|\mathbf{u}\| \tag{4}$$

3.3 Behaviour Coordination

Following the approach of Brook's subsumption architecture, the behaviours are placed on layers depending on how much they actually work on the robot's hardware. The inputs and outputs of each behaviour are connected with each other to form a network structure (see Fig. 5 for an example). These connections transport control and sensor information as well as loop-back information like r, ι and a of different behaviours. In the case that several outputs of different behaviours are connected to the same input of another behaviour, an behaviour activity based fusion scheme is used (Fig. 6). This scheme favours the output of behaviours having an unmet target (i.e. a high r) and a high activity a implying a ι greater than 0. The actual fusion of the outputs is done with the help of a fusion knot. Basically a fusion knot is no more than a fusion function f ((5)) combining the outputs $\mathbf{u}$, the activities a and the target ratings r of all behaviours involved to generate one output vector $\mathbf{u}'$.

$$\begin{aligned} &f : \Re^n \times \ldots \times \Re^n \times \Re \times \ldots \Re \rightarrow \Re^n : \\ &f(\mathbf{u}_0, \ldots, \mathbf{u}_n, a_0, \ldots, a_n, r_0, \ldots, r_n) = \mathbf{u}' \end{aligned} \tag{5}$$

In most cases either only the output of the behaviour with the highest activity (winner takes it all) is used or the average of all outputs weighted by the

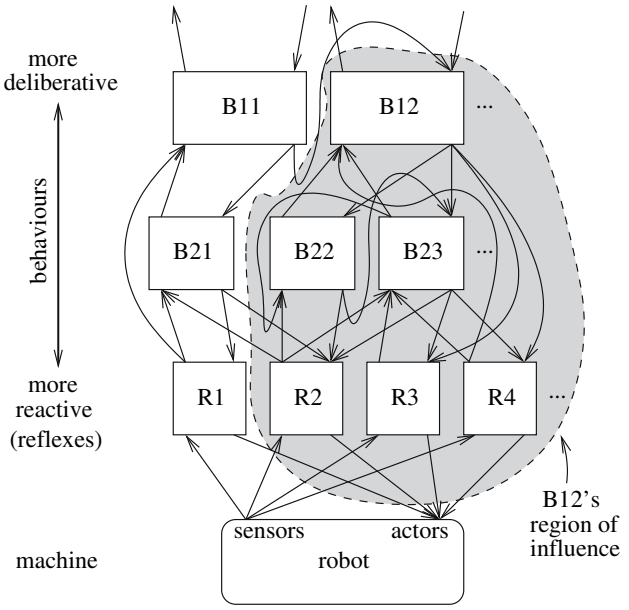


Fig. 5. Behaviour coordination network

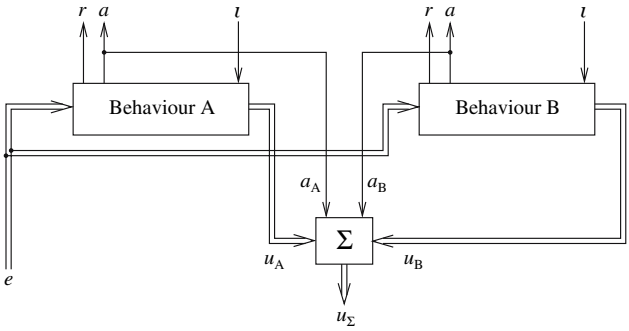


Fig. 6. Fusion of different behaviours outputs using their activation as weighting or selection criterion

activities is calculated by *f*. However it is possible to implement even more complex fusion schemes if it is necessary. This allows the usage of the behaviours activity a and target rating r as a mean for behaviour coordination without directly using the values of the actual control data generated by the behaviour. This way the output fusion is independent from the scaling of the control data and it is ensured that two behaviours competing over the influence on another behaviour won't annihilate each others output. This fusion scheme can be positioned between fuzzy and multiple objective behaviour coordination [26].

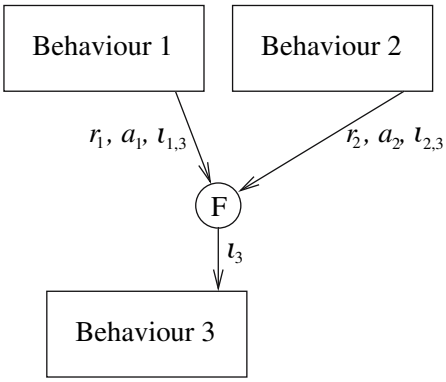


Fig. 7. The general fusion scheme

The activity in such a network will concentrate inside the influence area or region $\mathcal{R}$ of a higher level behaviour. $\mathcal{R}$ is recursively defined as in (6), where $act(B)$ is the set of behaviours being influenced by $\mathcal{B}$ via ι.

$$\begin{aligned} \mathcal{R}(B) &= \bigcup_{B_i \in act(B)} \{B_i \cup \mathcal{R}(B_i)\}, \\ \mathcal{R}(B) &= \emptyset, \quad \text{if} \quad act(B) = \emptyset \, . \end{aligned} \tag{6}$$

A region is an organisational unit of cooperating behaviours. The affiliation of a behaviour to a region is not exclusive. For example a leg's swing behaviour is located in the region of a stride and a trot behaviour. To coordinate the influence of several highlevel behaviours on a lower one, a fusion knot is inserted before its ι input. An example for the coordination of the influence of two behaviours on a third is shown in Fig. 7.

To allow the transition of the activity between different regions, r and a are loop-backed between behaviours on the same level as well as on others. This way it is possible that at a point where one behaviour cannot handle a situation anymore even when it is fully in charge, another behaviour which can solve the situation is activated while the other behaviour is deactivated. If, for example, the desired speed of a walking machine cannot be achieved a gait change is initiated.

4 Navigation in Unstructured Terrain

Navigation in unstructured terrain is a difficult challenge. Compared to structured environments, the geometry of unstructured areas cannot be described in detail, because there is no complete a-priori knowledge available on the surface and the ground conditions on which the walking machine has to operate. Legged machines are very suitable for navigating in this sort of terrain due

to the great flexibility in their motion and their ability to cope with very uneven environment. Proper navigation of walking machines, however, demands fairly complex control mechanisms, and suitable walking behaviour can only be achieved by considering knowledge of the surrounding environment. Therefore, building a world model and analysing the gathered information is a necessary assumption to modify the robot's behaviour in a sensible way. For the build up of a geometrical and/or topological model of the environment the knowledge of the spatial position and orientation of the robot is mandatory.

Following this the navigation task for a walking machine operating in unstructured environment demands an adequate solution of the sub-tasks *localisation, map building and planning and action selection*.

4.1 Lauron III

For the analysis of navigation aspects a robust and fully developed test platform becomes necessary. LAURON III is a stick insect like walking machine built as a universal platform for locomotion in rough terrain (see Fig. 8 (right)). It consists of a main body, a head and six equal legs, each providing three degrees of freedom ($\alpha \pm 80, \beta \pm 100, \gamma \pm 140$). With all components mounted on, the weight of LAURON III is about 18 kg. It has a length and width between the footsteps of about 70 cm and a maximal payload of 10 kg. The maximum distance between body and ground is 36 cm, the maximum velocity is given by $0.3\,\mathrm{ms}^{-1}$. LAURON III matches the requirements for autonomy, whereas the accumulators last for about 45 min (average power consumption 80 Watt). Walking in rough terrain requires information about the local environment and the inner state of the walking machine. Its joint angles are measured by optical encoders, the load of the motors could be determined by the measured currents. LAURON III is equipped with foot force sensors mounted on the lower leg, allowing the measurement of all three axial force components.

Fig. 8. (*Right*) Six legged test platform LAURON III walking in an outdoor environment. (*Left*) Visualisation of the 3D environement grid. The two linear traces have been acquired by the foot sensors only, since the distance sensor only scans in front of the robot. Darker cells (as seen especially on the trace of the robot's feet) indicate higher credibility

This construction enables the determination of the contact forces including direction and quantity. The main body carries an inertial sensor box delivering information about the angular velocity and the axial acceleration, each in three dimensions. Beyond that information about the magnetical field is available. Furthermore, a laser distance sensors (range: 20–500 cm, resolution: 0.5–3 cm) is fixed in the middle of the head for scanning the near environment. The sensor is fully movable around two axes, allowing the laser to be pointed at every point in the proximity of the machine. A GPS receiver mounted on LAURON III provides global position information in outdoor environment.

4.2 Localisation of Walking Machines

Small walking robots are not able to carry big and heavy sensor equipment, so navigation must be possible on the basis of sparse sensory information. Therefore the robot localisation task must be performed with high accuracy to guarantee a consistent and thus a useable model of the environment. Because a single localisation method is not expected to provide the required precision in the rough outdoor environment, the combination of different localisation systems has to be investigated.

The first relative measuring system to be examined is the calculation of the robot odometry on base of the joint encoders of the legs [33]. This system provides the general trend of the robot movement. The use of an inertial navigation system including magnetic field sensors provides further information about the orientation of the robot. In addition it enables the detection of disturbances like slipping of the whole robot which could not be detected by the odometry system. To compensate the drift of the relative measuring systems a DGPS system is introduced. This enables the robot to obtain global position information if the mandatory satellites are available. In case the DGPS system is unsuitable the detection and tracking of natural landmarks with the help of a stereo camera system should be introduced.

4.3 Environment Modelling

Because navigation is supposed to be done in unstructured, mostly unknown terrain, the representation of the environment is subject to some restrictions. It must be possible to start with an empty map because no a-priori knowledge can be assumed. Also our method must be completely independent of edge detection, as no edges can be expected in a natural environment. In order to place the feet at proper positions the resolution of the maps, especially in the direct surroundings of the machine, must be adequately high and as exact as possible. Taking these points into account leads us to the necessity of using a geometric approach for representing the environment. To handle spots of insufficient environment information, the model must provide the possibility of invoking additional measurements.

Taking all these points into account the use of some kind of *occupancy grid* [34] is very suitable. Our approach, the *advanced inference grid* [27], is a variant of the *vector field histogram* [36]. The fusion of the sparse sensor data of the walking robot is realised in a three dimensional grid by using fuzzy reasoning (see Fig. 8 (left)). Out of each measurement (which can be achieved by the distance laser sensor and the pressure sensors in the robot's feet) the global coordinates of the hit position and the corresponding grid cell are calculated, and its content is modified. In our approach the content of each cell does not consist only of an occupancy value, a *credibility* value is also stored. Both values range from 0 to 1. The occupancy value reveals the occupancy state of the area represented by the grid cell – insofar as it is known. The credibility value states how much we are able to trust this observation. This division is done by the following purpose: we want to distinguish whether a stored occupancy of i.e. 0.5 indicates a fairly correct observation of an inhomogeneous environment or just measurements of improper quality, a drawback that, if detected, can easily be repaired by invoking additional measurements or, alternatively, a more cautious walking behaviour of the robot.

One of the main differences from the vector field histogram is the use of additional meta information in the fuzzy reasoning based sensor fusion procedure. Meta information is e.g. the quality of the sensor delivering the measurement, the motion status of the robot the quantity of prior measurements or the neighbourhood homogeneity of the determined cell. So the influence of each measurement is dependent on the quality of both prior and actual data. Overaged information will be easily erased out of the model, new and good observations will have greater influence than redundant weak-quality information.

The use of fuzzy reasoning for cell updating has many advantages: non-linear correlations are easily couched in terms, external model knowledge can be written down in a straightforward manner, the rules are easily extensible by just adding new ones, and they are simple to understand and adapt. All of this offers excellent portability to different robot architectures, which was one of the main goals of that work. Finally, the method proved to be very efficient under real time conditions, offering better results with fewer measurements compared to the vector field histogram approach.

4.4 Planning and Action Selection

To enable the robot to select its future actions is the main goal of the localisation and map building efforts. The current work on the robot Lauron III is focussing that issue. In this connexion there could be different intentions be distinguished. The first is the determination of the very next foothold positions of the robot. In respect to walking robots this decision is critical because it directly influences the stability and therefore the success of the walking movement especially on irregular terrain. As second goal the overall path planning of the walking robot has to be taken in account. In contrary

to wheeled robots walking machines are able to overcome obstacles. So the simple classification in occupied and not occupied regions for path planning is not sufficient any more. To make use of the great flexibility of walking robots the standard path planning algorithms have to be extended. Finally it is appreciated that the robot adapts its walking behaviour to the environmental conditions. In complex terrain the robot should walk in a quite safe manner and react on disturbances carefully whereas in easy terrain the robot could increase its speed.

Because planning in a three dimensional environment model is computational very expensive and small autonomous walking robots don't provide the latest computer technology the 3D model is only present for the near environment. The global environment model is realised as two dimensional geometrical height map and is generated out of the 3D model by projection. In the 2D model the meta information for the sensor data fusion is obsolete. On the other hand the 2D model is augmented by reusable information of the planning algorithms.

References

1. Gienger, M., Löffler, K., and Pfeiffer, F. (2001). In *Proc. of the IEEE International Conference on Robotics and automation (ICRA)*.
2. Löffler, K., Gienger, M., and Pfeiffer, F. (2001). Simulation and control of a biped jogging robot. In *Proceedings of the 4th International Conference on Climbing and Walking Robots (CLAWAR)*.
3. J. Ayers, J. Witting, C. Wilbur, P. Zavracky, N. McGruer, and D. Massa. Biomimetic robots for shallow water mine countermeasures. In *Proc. of the Autonomous Vehicles in Mine Countermeasures Symposium*, 2000.
4. Kimura, H., Fukuoka, Y., Hada, Y., and Takase, K. (2001). Three-dimensional adpative dynamic walking of a quadruped robot by using neural system model. In *Proc. of the 4th International Conference on Climbing and Walking Robots (CLAWAR)*, Karlsruhe. FZI.
5. H. Witte, R. Hackert, K. Lilje, N. Schilling, D. Voges, G. Klauer, W. Ilg, J. Albiez, A. Seyfarth, D. Germann, M. Hiller, R. Dillmann, M. Fischer: Transfer of biological principles into the construction of quadruped walking machines, 2nd International Workshop On Robot Motion and Control (ROMOCO), Poland, 2001
6. Pearson, K. (1995). Proprioceptive regulation of locomotion. *Current Opinions in Neurobiology*, 5(6):768–791.
7. Cruse, H. and Bartling, C. (1995). Movement of joint angles in the legs of a walking insect, carausius morosus. *J. Insect Physiol.*, (41):761–771.
8. Cruse, H., Dürr, V., and Schmitz, J. (2001). Control of a hexapod walking – a decentralized solution based on biological data. In *Proc. of the 4th International Conference on Climbing and Walking Robots (CLAWAR)*, Karlsruhe, Germany.
9. J. Albiez, T. Kerscher, F. Grimminger, U. Hochholdinger, R. Dillmann, K. Berns PANTER – Prototype for a fast running Quadruped Robot with pneumatic Muscles CLAWAR 2003, 6th International Conference on Climbing and Walking

Robots and the Support Technologies for Mobile Machines, pp. 617–624, 17–19 September 2003, Catania, Italy
10. B. Tondu, P. Lopez: Modeling and Control of McKibben Artificial Muscle Robotti Actuators, IEEE Control Systems Magazine, April 2000, Vol. 20, 15–38
11. T. Kerscher, J. Albiez, J. M. Zoellner and R. Dillmann AirInsect A New Innovative Biological Inspired Six-Legged Walking Machine Driven by Fluidic Muscles Proceedings of IAS 8, The 8th Conference on Intelligent Autonomous Systems, 10–13 March 2004, Amsterdam, The Netherlands
12. B. Gassmann, K.-U. Scholl, K. Berns: *Locomotion of LAURON III in rough terrain*. Proceedings of Advanced Intelligent Mechatronics, 2001, 959–965
13. J. Albiez, T. Luksch, K. Berns, R. Dillmann: An Activation Based Behaviour Control Architecture for Walking Machines, In Proccedings of the 7th International Conference on Simulation of Adaptive Behaviour, Edingburgh, 2002
14. S. Davis, D. Caldwell: The bio-mimetic design of a robot primate using pneumatic muscle actuators, In Proc. of the 4th International Conference on Climbing and Walking Robots (CLAWAR), 2001
15. S. Hirose, K. Yoneda, H. Tsukagoshi; TITAN VII: Quadruped Walking and Manipulating Robot on a Steep Slope, Proc. Int. Conf. on Robotics and Automation, Albuquerque, New Mexico, 1997, 494–500
16. Y. Fukuoka, H. Nakamura and H. Kimura: Biologically-Inspired Adaptive Dynamic Walking of the Quadruped on Irregular Terrain, Proc. of IEEE&RSJ IROS99, Kyongju, 1999, 1657–1662
17. M. A. Lewis: Autonomous Robots, In Press, 2002
18. M. Raibert: Legged Robots that Balance, MIT Press, Cambridge, MA, 1986
19. S. Talebi, I. Poulakakis, E. Papadopoulos and M. Buehler: Qaudruped Robot Running with a bounding gait, Int. Symp. Experimental Robotics, Honolulu, 2000
20. H. Kimura, Y. Fukuoka, Y. Hada and K. Takase: Three-dimensional adaptive dynamic walking of a quadruped robot by using neural system model, In Proc. of the 4th International Conference on Climbing and Walking Robots (CLAWAR), 2001
21. Y. Endo and R. Arkin. Implementing tolman's schematic sowbug: Behaviour-based robotics in the 1930's. In *Proceedings of the 2001 IEEE International Conference on Robotics and Autonomous Systems*, 2001.
22. R. Arkin, A. Kahled, A. Weitzenfeld, and F. Cervantes-Prez. Behavioral models of the praying mantis as a basis for robotic behavior. *Journal of Autonomous Systems*, 2000.
23. M. J. Mataric. Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, Special issue on Software Architectures for Physical Agents, 9(2–3):323–336, 1997.
24. M. Likhachev and R. Arkin. Robotic comfort zones. In *Proceedings of the SPIE: Sensor Fusion and Decentralized Control in Robotic Systems*, volume 4196, pp. 27–41, 2000.
25. M. Likhachev and R. Arkin. Spatio-temporal case-based reasoning for behavioral selection. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1627–1634, 2001.
26. P. Pirajanian. Behaviour coordination mechanisms – state-of-the-art. Technical Report IRIS-99-375, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, 1999.

27. R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, April 1986.
28. C. Ferrell. Global behavior via cooperative local control. volume 2, pp. 105–125, 1995.
29. R. Arkin. *Behavior-Based Robotics*. MIT Press, 2000.
30. E.R. Kandel, J.H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. McGraw-Hill, 4th ed. edition, 2000.
31. J. Albiez and T. Luksch and K. Berns and R. Dillmann An Activation-Based Behavior Control Architecture for Walking Machines *The International Journal of Robotics Research*, 22(3–4), 2003, pp. 203–211.
32. J. Albiez and T. Luksch and K. Berns and R. Dillmann Reactive reflex-based control for a four-legged walking machine newblock *Robotics and Autonomous Systems*, 44, 2003, pp. 181–189
33. G. P. Roston and E. P. Krotkov. Dead reckoning navigation for walking robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 607–612, Raleigh, NC, USA, July 1992. IEEE/RSJ.
34. A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.
35. B. Gaßmann and K. Berns. Local navigation of LAURON III for walking in rough terrain. In *Proceedings of the 5th International Conference on Climbing and Walking Robots*, pp. 509–514, Paris, France, September 2002.
36. J. Borenstein and Y. Koren. The vector field histogram – fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, June 1991.

Part II

Control

Integer vs. Fractional Order Control of a Hexapod Robot

Manuel F. Silva[1], J. A. Tenreiro Machado[1], and António M. Lopes[2]

[1] Department of Electrical Engineering, Institute of Engineering of Porto, Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal
{mfsilva,jtm}@dee.isep.ipp.pt
[2] Department of Mechanical Engineering, Faculty of Engineering of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
aml@fe.up.pt

Summary. This paper studies the performance of integer and fractional order controllers in a hexapod robot with joints at the legs having viscous friction and flexibility. For that objective the robot prescribed motion is characterized in terms of several locomotion variables. The controller performance is analised through the Nyquist stability criterion. A set of model-based experiments reveals the influence of the different controller implementations upon the proposed metrics.

1 Introduction

Walking machines allow locomotion in terrain inaccessible to other type of vehicles, since they do not need a continuous support surface. On the other hand, the requirements for leg coordination and control impose difficulties beyond those encountered in wheeled robots. Previous studies focused mainly in the control at the leg level and leg coordination using neural networks, fuzzy logic, central pattern generators, subsumption architecture and virtual model control. There is also a growing interest in using insect locomotion schemes to control walking robots. In spite of the diversity of approaches, the control at the joint level is usually implemented through a simple *PID* like scheme. Other approaches include sliding mode control [1], computed torque control [2] and hybrid force/position control [3].

The application of the theory of fractional calculus in robotics is still in a research stage, but the recent progress in this area reveals promising aspects for future developments [4]. With these facts in mind, a simulation model of multi-leg locomotion systems was developed. Based on this tool, the present study compares different robot controllers, namely integer and fractional order proportional and derivative algorithms, in the presence of motors with saturation and joints with viscous friction and flexibility. Simulations reveal

the superior performance of the fractional order controller, in terms of Phase Margin, both for the case of gain variation, due to actuator saturation, and for the case of non-ideal joint transmissions, due to viscous friction and flexibility.

Bearing these ideas in mind, the paper is organized as follows. Section two introduces the hexapod model and the motion planning scheme. Section three presents the robot dynamics and control architecture. Section four establishes the optimizing metrics and section five develops a set of experiments that compare the performance of the different controllers. Finally, section six outlines the main conclusions and directions towards future developments.

2 Robot Kinematics and Trajectory Planning

We consider a walking system (Fig. 1) with $n = 6$ legs, equally distributed along both sides of the robot body, having each one $m = 2$ rotational joints.

The kinematic model comprises: the cycle time T, the duty factor β, the transference time $t_T = (1-\beta)T$, the support time $t_S = \beta T$, the step length L_S, the stroke pitch S_P, the body height H_B, the maximum foot clearance F_C, the ith leg lengths L_{i1} and L_{i2} and the foot trajectory offset O_i $(i = 1, \ldots, n)$.

The robot body, and by consequence the legs hips, is assumed to have a horizontal movement with a constant forward speed $V_F = L_S/T$. Concerning the robot feet, we consider a periodic trajectory for each foot, being the

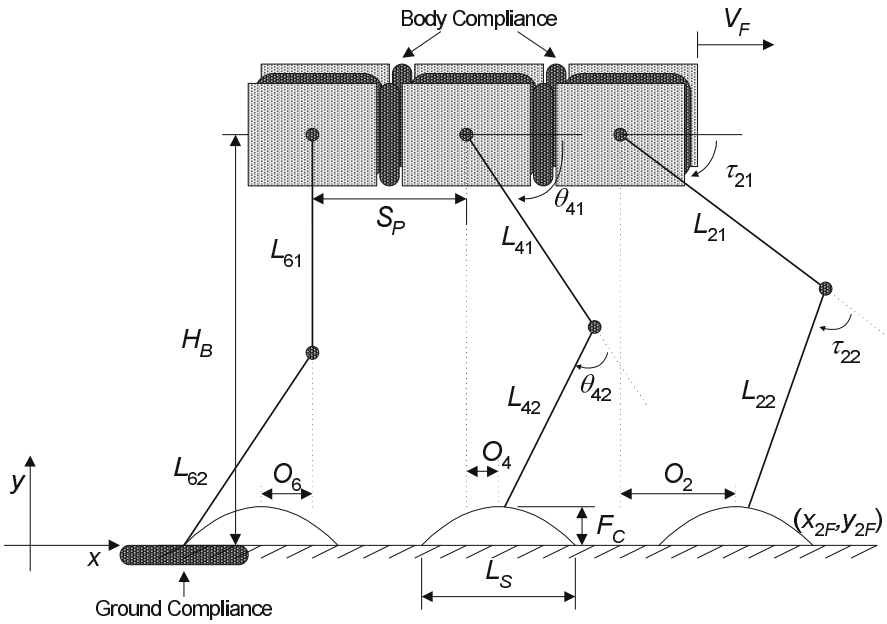


Fig. 1. Kinematic and dynamic multi-legged robot model

trajectory of the foot of the swing leg, for each cycle, computed through a cycloid function [5]. Based on this data, the trajectory generator is responsible for producing a motion that synchronizes and coordinates the legs.

The algorithm for the forward motion planning accepts the desired cartesian trajectories of the legs hips $\mathbf{p_{Hd}}(t) = [x_{iHd}(t),\ y_{iHd}(t)]^{\mathrm{T}}$ and feet $\mathbf{p_{Fd}}(t) = [x_{iFd}(t),\ y_{iFd}(t)]^{\mathrm{T}}$ as inputs and, by means of an inverse kinematics algorithm, generates the related joint trajectories $\mathbf{q_d}(t)$=$[\theta_{i1d}(t),\ \theta_{i2d}(t)]^{\mathrm{T}}$, selecting the solution corresponding to a forward knee [5].

3 Robot Dynamics and Control Architecture

3.1 System Dynamic Model

The model for the robot inverse dynamics is formulated as:

$$\mathbf{\Gamma} = \mathbf{H}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) - \mathbf{F_{RH}} - \mathbf{J_F^T}(\mathbf{q})\mathbf{F_{RF}} \tag{1}$$

where $\mathbf{\Gamma} = [f_{ixH},\ f_{iyH},\ \tau_{i1},\ \tau_{i2}]^{\mathrm{T}}$ $(i = 1, \ldots, n)$ is the vector of forces/ torques, $\mathbf{q} = [x_{iH},\ y_{iH},\ \theta_{i1},\ \theta_{i2}]^{\mathrm{T}}$ is the vector of position coordinates, $\mathbf{H}(\mathbf{q})$ is the inertia matrix and $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{g}(\mathbf{q})$ are the vectors of centrifugal/Coriolis and gravitational forces/torques, respectively. The matrix $\mathbf{J_F^T}(\mathbf{q})$ is the transpose of the robot Jacobian matrix, $\mathbf{F_{RH}}$ is the vector of the body inter-segment forces and $\mathbf{F_{RF}}$ is the vector of the reaction forces that the ground exerts on the robot feet. Moreover, it is considered that the forces $\mathbf{F_{RF}}$ are null during the foot transfer phase.

In a first phase, we consider that the joint actuators and transmissions are ideal, as presented in Fig. 2a. Afterwards, in a second phase, we consider that the joint transmissions are non ideal, exhibiting a compliant behaviour (Fig. 2b) that can be described by:

$$\mathbf{\Gamma_m} = \mathbf{J_m}\ddot{\mathbf{q}}_\mathbf{m} + \mathbf{B_m}\dot{\mathbf{q}}_\mathbf{m} + \mathbf{K_T}(\mathbf{q_m} - \mathbf{q}) \tag{2}$$

$$\mathbf{K_T}(\mathbf{q_m} - \mathbf{q}) = \mathbf{H}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) - \mathbf{F_{RH}} - \mathbf{J_F^T}(\mathbf{q})\mathbf{F_{RF}} \tag{3}$$

where $\mathbf{\Gamma_m} = [\tau_{i1m},\ \tau_{i2m}]^{\mathrm{T}}$ $(i = 1, \ldots, n)$ is the vector of motor torques, $\mathbf{q_m} = [\theta_{i1m},\ \theta_{i2m}]^{\mathrm{T}}$ is the vector of motor position coordinates and $\mathbf{J_m}$, $\mathbf{B_m}$ and $\mathbf{K_T}$ are the matrices of the motor and transmissions inertia, viscous friction coefficients and stiffness, respectively.

Furthermore, we also consider that the joint actuators are not ideal, exhibiting a saturation given by:

$$\tau_{ijm} = \begin{cases} \tau_{ijC} & ,\ |\tau_{ijm}| \leq \tau_{ijMax} \\ sgn(\tau_{ijC})\,\tau_{ijMax} & ,\ |\tau_{ijm}| > \tau_{ijMax} \end{cases} \tag{4}$$

where, for leg i and joint j, τ_{ijC} is the controller demanded torque, τ_{ijMax} is the maximum torque that the actuator can supply and τ_{ijm} is the motor effective torque.

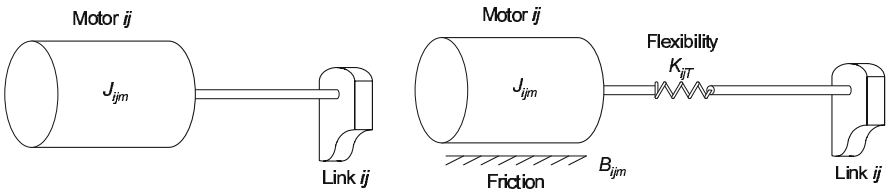


Fig. 2. Model of the leg joint: ideal actuator and transmission (*left*) and actuator with friction and transmission flexibility (*right*)

Table 1. System parameters

Robot model parameters			
S_P	1 m	J_{i1m}	0.00375 kgm^2
L_{ij}	0.5 m	J_{i2m}	0.000625 kgm^2
O_i	0 m	B_{ijm}	10 Nm rad^{-1}s
M_{ij}	1 kg	K_{ijT}	100000 Nm rad^{-1}
M_b	88.0 kg	J_{i1T}	0.001875 kgm^2
M_{if}	0.1 kg	J_{i2T}	0.0003125 kgm^2
K_{xH}	10^5 Nm^{-1}	B_{ijT}	10 Nm rad^{-1}s
K_{yH}	10^4 Nm^{-1}		
B_{xH}	10^3 Nsm^{-1}		
B_{yH}	10^2 Nsm^{-1}		
Locomotion parameters		**Ground parameters**	
β	50%	K_{xF}	1302152.0 Nm^{-1}
L_S	1 m	K_{yF}	1705199.0 Nm^{-1}
H_B	0.9 m	B_{xF}	2364932.0 Nsm^{-1}
F_C	0.1 m	B_{yF}	2706233.0 Nsm^{-1}
V_F	1 ms^{-1}	v	0.9

Figure 1 presents the dynamic model for the hexapod body and foot-ground interaction [5].

The robot body is divided in n identical segments (each with mass $M_b n^{-1}$) and a linear spring-damper system is adopted to implement the intra-body compliance, being its parameters ($B_{\eta H}$ and $K_{\eta H}$ ($\eta = \{x, y\}$) in the {horizontal, vertical} directions, respectively) defined so that the body behaviour is similar to the one expected to occur on a living animal (Table 1).

The contact of the ith robot feet with the ground is modelled through a non-linear system with a linear stiffness $K_{\eta F}$ and a non-linear damping $B_{\eta F}$ ($\eta = \{x, y\}$) in the {horizontal, vertical} directions, respectively. The values of the parameters $K_{\eta F}$ and $B_{\eta F}$ (Table 1) are based on the studies of soil mechanics.

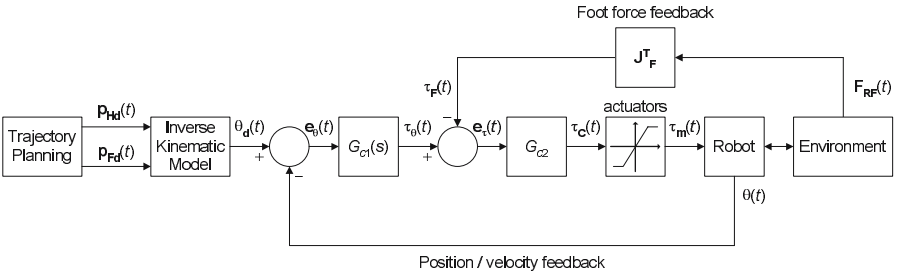


Fig. 3. Hexapod robot control architecture

3.2 Control Architecture

The general control architecture of the hexapod robot is presented in Fig. 3. The superior performance of this controller structure, with PD position control and an inner loop involving foot force feedback, was highlighted for the case of having non-ideal actuators with saturation or variable ground characteristics [5]. Based on these results, in this study we evaluate the performance of integer and fractional order proportional and derivative algorithms for $G_{c1}(s)$, while for G_{c2} it is considered a simple P controller.

The proportional and derivative algorithm (PD^α) is defined as:

$$G_{C1j}(s) = K_{pj} + K_{\alpha j} s^{\alpha_j}, \alpha_j \in \Re,\ j = 1, 2 \tag{5}$$

where K_{pj} and $K_{\alpha j}$ are the proportional and derivative gains, respectively, and α_j is the fractional order. Therefore, the classical PD^1 algorithm in (5) occurs when the fractional order $\alpha_j = 1.0$.

In this paper, for implementing the fractional PD^α algorithm (5) it is adopted a discrete-time 4^{th}-order Padé approximation (a_{ij}, $b_{ij} \in \Re$, $j = 1,2$) [6] yielding an equation in the z-domain of the type:

$$G_{C1j}(z) \approx K_j \sum_{i=0}^{i=4} a_{ij}\, z^{-i} \Big/ \sum_{i=0}^{i=4} b_{ij}\, z^{-i} \tag{6}$$

where K_j is the controller gain and a_{ij} and b_{ij} are the coefficients of the Padé approximation.

4 Metrics for Performance Evaluation

As a measure of the controller performance, we consider the Phase Margin (PM) based on the Nyquist plot. Since this MIMO system presents coupling between the joint variables of the legs, we adopt the block diagram of Fig. 4 for its approximate representation.

The corresponding characteristic equation ($Q(j\omega)$) yields:

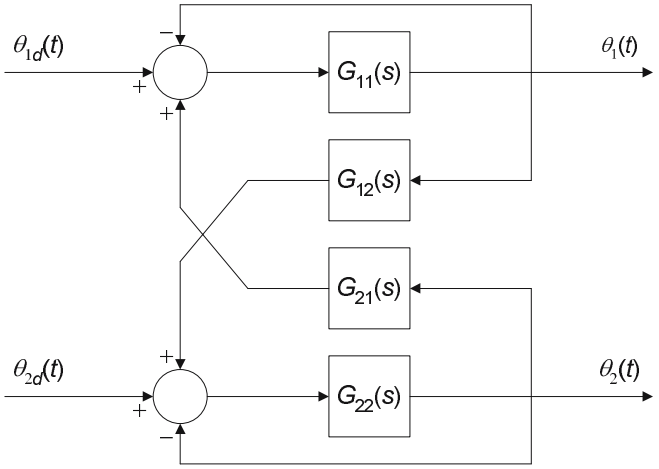


Fig. 4. MIMO block diagram of the walking robot at the leg joints level

$$Q(j\omega) = 1 + G_{11} + G_{22} + G_{11}G_{22} - G_{11}G_{12}G_{21}G_{22} \tag{7}$$

where G_{12} and G_{21} represent the transfer functions corresponding to the feedback coupling between the joint leg positions.

5 Simulation Results

In this section we develop a set of simulations to compare the controller performances during a periodic wave gait with different dynamic phenomena at the leg joints. For simulation purposes we consider the robot body parameters, the locomotion parameters and the ground parameters presented in Table 1.

To tune the controller we follow a systematic method, testing and evaluating a grid of several possible combinations of controller parameters, for the PD^1 controller architecture, while establishing a compromise in what concerns the minimization of the hip trajectory following errors (ε_{xyH}) and the mean absolute density of energy per travelled distance (E_{av}) [6]. Moreover, it is assumed high performance joint actuators with a maximum actuator torque in (4) of $\tau_{ijMax} = 400\,\text{Nm}$. Therefore, we adopt the $G_{c1}(s)$ parameters presented in Table 2 and a proportional controller G_{c2} with gain $K_{pj} = 0.9$ $(j = 1, 2)$.

To easy the comparison, in the case of the fractional PD^α controller we restrict to gain values identical to those established for the integer-order PD^1 algorithm. Moreover, we consider PD^α controllers corresponding to distinct values of the fractional order α_j. We start by considering the interval $-2.0 \leq \alpha_j \leq 1.0$ for the fractional order, but we verify that for $-2.0 \leq \alpha_j \leq -0.1$ the PD^α controller is unstable. Moreover, for $0.1 \leq \alpha_j \leq 0.3$ the controller performance is weak, with the robot presenting large trajectory tracking errors

Table 2. Controller parameters for $G_{c1}(s)$ when establishing a compromise between the minimization of ε_{xyH} and E_{av} for $G_{c2} = 0.9$

PD^1	Joint $j = 1$	K_{p1}	1500	$\varepsilon_{xyH} = 0.73$
		$K_{\alpha 1}$	300	
	Joint $j = 2$	K_{p2}	4000	$E_{av} = 373.7$
		$K_{\alpha 2}$	10	

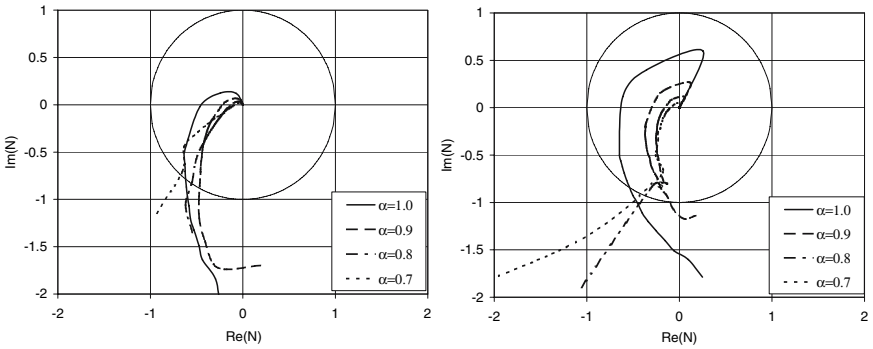


Fig. 5. Nyquist plots for the PD^{α} ($\alpha = \{0.7,\ 0.8,\ 0.9,\ 1.0\}$) controllers considering ideal joints (*left*) and joints with viscous friction and flexibility (*right*) and $\tau_{ijMax} =$ 400 Nm

while walking. Therefore, in the sequel we limit to values of the fractional order in the interval $0.4 \leq \alpha_j \leq 1.0$.

With these controller parameters, we analyze the system performance for the different controller implementations, while having dynamical effects on the leg joints. In a first phase, we start by studying the case of ideal joint transmission and, afterwards, we augment the model by including flexibility, in order to observe its influence upon the performance metrics. In a second phase, we repeat the experiments for lower values of τ_{ijMax} to study the effect of increasing levels of actuator saturation.

Figure 5 depicts the Nyquist plots for the different order of the PD^{α} controller ($\alpha = \{0.7,\ 0.8,\ 0.9,\ 1.0\}$) with ideal joints (left) and joints with viscous friction and flexibility (right), when considering $\tau_{ijMax} = 400$ Nm. For the case of ideal joint transmission on the robot legs, it is possible to observe that the PM is maximum for the $PD^{0.9}$ controller implementation. Comparing with the classical PD^1 controller the PM is also higher for the $PD^{0.8}$ algorithm.

The Nyquist plots corresponding to increasing levels of saturation are presented in Figs. 6 and 7, namely for $\tau_{ijMax} = 200$ Nm and $\tau_{ijMax} = 100$ Nm. From the analysis of these plots it is possible to conclude that in the case of ideal joint transmissions, from the point of view of the PM, the relative performance of the different controller implementations remains unchanged.

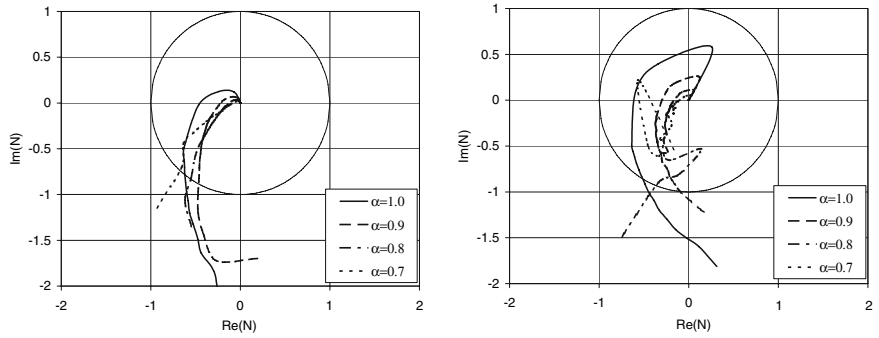


Fig. 6. Nyquist plots for the PD^{α} ($\alpha = \{0.7,\ 0.8,\ 0.9,\ 1.0\}$) controllers considering ideal joints (*left*) and joints with viscous friction and flexibility (*right*) and $\tau_{ijMax} = 200\,\text{Nm}$

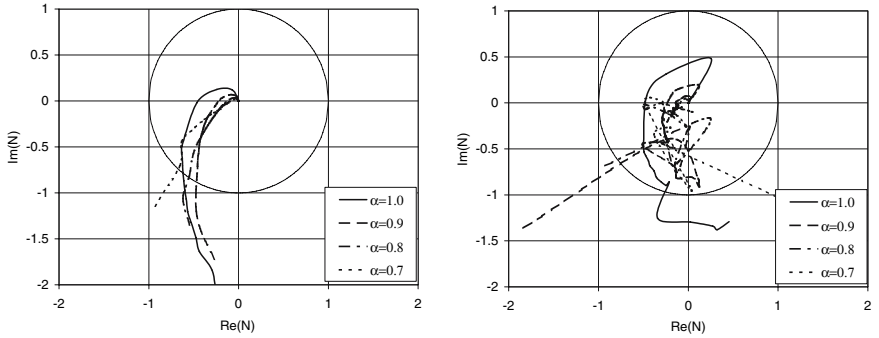


Fig. 7. Nyquist plots for the PD^{α} ($\alpha = \{0.7,\ 0.8,\ 0.9,\ 1.0\}$) controllers considering ideal joints (*left*) and joints with viscous friction and flexibility (*right*) and $\tau_{ijMax} = 100\,\text{Nm}$

For joints with viscous friction and flexibility the situation is relatively different. When considering almost ideal actuators (i.e., $\tau_{ijMax} = 400$ Nm) the fractional PD^{α} ($0.7 \leq \alpha \leq 0.9$) controller implementation presents a better PM than the integer PD^{1} algorithm (see Fig. 5 (right)). For moderate saturation levels (e.g., $\tau_{ijMax} \simeq 200$ Nm) we observe that, the relative PM of the different controller implementations presents no significant variation (see Fig. 6 (right)). From Fig. 7 (right) we can conclude that, in case of strong actuator saturation (e.g., $\tau_{ijMax} \simeq 100\,\text{Nm}$), the Nyquist plot reveals a large degradation, indicating stability difficulties for all tested control algorithms.

In order to verify the above presented results we study the time response. Therefore, we apply a pulse of amplitude $\delta y_{1F} = 0.01$ m, with duration $\delta t = 0.1$ s, at the robot feet reference trajectory. The resulting feet trajectory error is then analyzed in order to determine the Percentual Overshoot (PO) of the response to the pulse input, for the different situations under study.

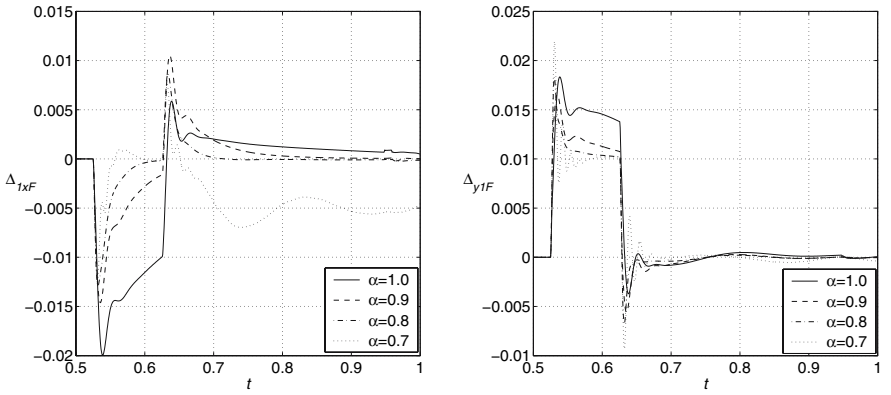


Fig. 8. Plots of the feet trajectory errors Δ_{1xF} and Δ_{1yF} *vs.* t, when the pulse perturbation ($\delta y_{1F} = 0.01$ m, $\delta t = 0.1$ s) is introduced in the beginning of the foot transfer phase, for the PD^{α} ($\alpha = \{0.7,\ 0.8,\ 0.9,\ 1.0\}$) controllers considering ideal joints and $\tau_{ijMax} = 400\,\mathrm{Nm}$

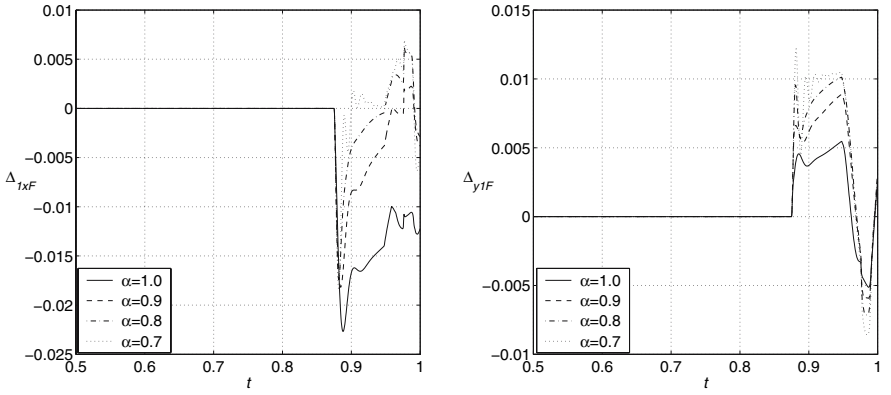


Fig. 9. Plots of the feet trajectory errors Δ_{1xF} and Δ_{1yF} *vs.* t, when the pulse perturbation ($\delta y_{1F} = 0.01$ m, $\delta t = 0.1$ s) is introduced near the end of the foot transfer phase, for the PD^{α} ($\alpha = \{0.7,\ 0.8,\ 0.9,\ 1.0\}$) controllers considering ideal joints and $\tau_{ijMax} = 400\,\mathrm{Nm}$

The PO of the resulting foot trajectories presents different values as a function of the time instant of the pulse introduction. For the case of ideal joints with $\tau_{ijMax} = 400$ Nm, we can observe in Figs. 8 and 9 the resulting x_{1F} and y_{1F} foot trajectories, for the different order of the controller algorithms, when the pulse is introduced at the beginning or at the end of the transfer phase, respectively.

For the case in which the pulse perturbation is introduced in the begining of the foot transfer phase (Fig. 8) it is possible to conclude that the $PD^{0.8}/PD^{0.9}$ controller algorithm presents the smaller PO in the x_{1F}/y_{1F} foot trajectory,

respectively. For this situation, the $PD^{0.8}$ controller algorithm presents the best trajectory following characteristics, for both trajectories.

When the pulse perturbation is introduced near the end of the foot transfer phase, the results are somehow different. In this case (Fig. 9) it is seen that the $PD^{0.8}/PD^{1}$ controller algorithm presents the smaller PO in the x_{1F}/y_{1F} foot trajectory, respectively. For this situation, the $PD^{0.8}$ controller algorithm reveals again the best trajectory following characteristics, for both trajectories.

For different values of the motor saturation torque level and joints with viscous frictions and flexibility the behavior is similar.

The different controller responses are due to the fact that we have a nonlinear and coupled system. Therefore, we conclude that the Nyquist Plots present an "average" indication of the controller performances along a full locomotion cycle, consisting on a support phase (in which the feet are on the ground supporting the robot weight) and a transfer phase (corresponding to the feet movement in the air to the new footstep position).

Analyzing the results of the previous simulations, we conclude that the fractional PD^{α} controller, for values of $0.8 \leq \alpha \leq 0.9$, presents superior results when we have actuator saturation and dynamical phenomena at the joints. This means that the fractional-order PD^{α} algorithm is more robust in a real operating condition than the classical integer-order PD^{1} algorithm that reveals robustness limitations.

Finally, another characteristic that makes the fractional PD^{α} control algorithm advantageous when compared with the classical PD^{1} scheme results from the fact that the fractional-order (α) represents an extra degree-of-freedom when tuning the controller.

6 Conclusions

In this paper we have compared the performance of integer and fractional-order controllers for joint leg control of hexapod robots.

By implementing joint leg actuator and transmission models that incorporate dynamical characteristics, such as a non-ideal saturation, viscous friction and flexibility, we estimated how the robot controllers respond to non-ideal joint actuators and transmissions.

The experiments reveal that the fractional-order PD^{α} controller implementation is superior to the integer-order PD^{1} algorithm, from the point of view of robustness, that is to say, when we have models that incorporate real operating dynamical phenomena.

While our focus has been on a dynamic analysis in periodic gaits, considering actuators with saturation and transmissions with viscous friction and flexibility, many aspects of locomotion are not necessarily captured by this study. Consequently, future work in this area will address the study of the

performance of the controllers when the hexapod is faced with variable ground conditions and obstacles in its path and different locomotion parameters.

References

1. Martins-Filho LS, Silvino JL, Resende P and Assunção TC (2003) Control of Robotic Leg Joints – Comparing PD and Sliding Mode Approaches. In: Muscato G and Longo D. (eds) Proc. CLAWAR'2003 – 6th Int. Conf. on Climbing and Walking Robots, Catania, Italy, 147–153.
2. Lee K-P, Koo T-W and Yoon Y-S (1998) Real-Time Dynamic Simulation of Quadruped Using Modified Velocity Transformation. In: Proc. of the IEEE Int. Conf. on Rob. and Aut. 1701–1706.
3. Song J, Low KH and Guo W (1999) A Simplified Hybrid Force/Position Controller Method for the Walking Robots. Robotica 17:583–589.
4. Silva MF, Machado JAT and Lopes AM (2003) Comparison of fractional and integer order control of an hexapod robot. In: Proc. VIB 2003 – ASME Int. 19th Biennial Conf. on Mech. Vib. and Noise, ASME, Chicago, Illinois, USA.
5. Silva MF, Machado JAT and Lopes AM (2003) Position / Force Control of a Walking Robot. MIROC–Machine Intelligence and Robotic Control 5:33–44.
6. Silva MF, Machado JAT and Lopes AM (2004) Fractional Order Control of a Hexapod Robot. Accepted for publication at Nonlinear Dynamics.

Synchronous Landing Control of a Rotating 4-Legged Robot, PEOPLER, for Stable Direction Change

T. Okada, Y. Hirokawa, T. Sakai, and K. Shibuya

Department of Biocybernetics, Niigata University, Ikarashi 2-8050, Niigata-shi, JAPAN 950-2181

Synopsis. The walking robot, PEOPLER can change its direction by making right and left leg's locomotion speed different. However, robot body swings when leg motions are not synchronized in left and right. Therefore, we control the leg posture so that the swing disappears. The principal idea is to close one side leg a little inside and open other side leg a little outside at the same time under such condition that close and open amounts are equal. This is intended to keep all hip joints at same height on the ground. We show four functions to determine leg postures depending on arm's angular shift. Simulation of steering is performed to clarify the robot body's trajectory. Leg positions are traced in experiment and its results show that the leg posture control is useful in practical applications.

Key words: Leg Robot, Walking, Steering, Leg Posture, Friction Coefficient

1 Introduction

Various kinds of mechanisms for designing mobile, climbing, and walking robots on irregular terrain have been proposed so far. These robots are requested to cope with plenty of bumpy obstacles. Particularly, leg type mechanism can manage to drive over the obstacles easily as compared with wheel type mechanism. Also, the mechanism is useful in climbing up and down stairs. These robots are seen in literatures, e.g. (1)–(6). Some of them drive legs toward up and down. In a complicated system, upward and downward motions are not direct because legs are articulated with joints. Insect design (7) for improved robot mobility and simple design (8) for robustness against unexpected environment have been proposed. Six-legged robot can change locomotion direction easily without leg slide, because redundant legs are divided into two complementary 3-legged groups freely as required for landing (9).

This paper proposes fundamental control associated with arms and legs of the robot, PEOPLER (10). Since leg posture is quite important to determine the amount of strides, we investigate appropriate posture for changing

direction of locomotion without rolling. We build a robot model and simulate its walking affected by leg postures and robot size. In a design of the miniature PEOPLER Mk-2, eccentric pulley is utilized and the robot demonstrates changing its direction while it walks, however the steer angle is fixed in a specific value since leg posture is connected mechanically with the hip joint axis. The principal idea is to close two legs a little inside for short stride, or open them a little outside for long stride. These treatments are performed at each hip joint and amount of the stride is adjusted in left and right for steering in need. We show four leg posture patterns. Simulation of the steering is performed to estimate the robot body's trajectory. Leg positions are traced and the results make it clear that the synchronous leg landing control is good for changing direction of walking without generating swings like pitch, roll, and yaw.

2 Nomenclature

b_l : distance between front and rear hips
b_w : width between right and left hips
d_i : displacement of leg landing position
i : subscript identifying hip, arm, and leg
n: natural number
r: radius of arm
D: difference between right and left strides
E: function of energy consumption
L: length of leg
W_i: load of each leg to support robot weight
α: rotation angle of the robot body
α_r: robot walk direction from Y-axis
β_i: leg sliding angle from front
γ : leg posture angle (γ_{max} when $\theta_r = 0$)
θ_a: arm angle from robot front
θ_r: arm angle from horizontal
θ_s: inclination angle of road surface
μ_i friction coefficient at leg landing position
ϕ: robot's turn angle when arm rotates π rad.
Σ_o: world co-ordinate system

3 Method of Changing Leg Landing Posture

Leg motion of bending and extending at knee joint contributes to change the amount of stride. But no relation among leg motions generates such uncomfortable swings like pitch, roll, and yaw while robot walks. Also, a control algorithm of the multijointed system is not simple, in general. Therefore, we

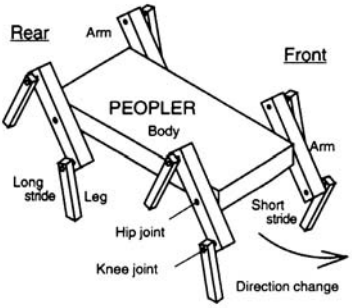


Fig. 1. Sketch of the PEOPLER in steering

synchronize the motions of both legs and arms to reduce these swings. Synchronized leg landing of the PEOPLER in walking is shown in Fig. 1.

3.1 Amount of a Stride

The PEOPLER has 4 hip joints and each joint is equipped with two arms connecting their ends with leg links individually. Joint of each connection behaves as a planetary knee joint (10). Two legs around the hip joint land alternately as the hip joint rotates. Generally, only four of eight leg links have contacts on the ground at an instantaneous landing. However in critical cases, more than 5 leg links have contacts on the ground. Actually, these cases are considered as leg landing combinations and we call the legs making these cases *virtual legs*. From simplified sketch in Fig. 2 we notice that each leg is free to take vertical or inclined postures as far as leg end can land while it walks. At

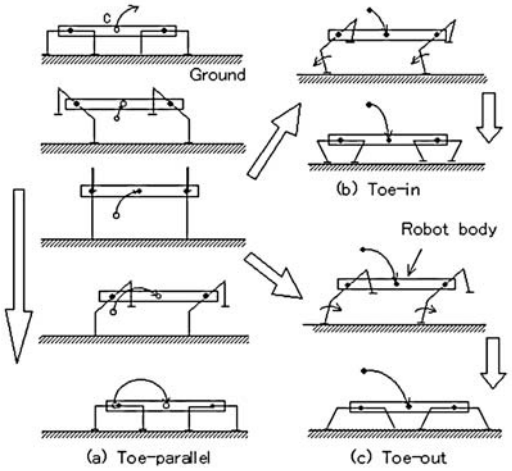


Fig. 2. Types of a stride constraints

this moment, we recognize that the stride means the distance between two leg ends located at a same hip joint. Principally, it is defined when there is eight virtual legs on a flat terrain. In fact, when strides of all legs at hip joints are same in their amounts, the robot can walk with a short stride, standard stride, or long stride toward forward or backward directions.

3.2 Steering by a Stride Control as a Function of Arm's Angular Shift

In the Fig. 2, legs in (a) are always constrained to be parallel together to walk forward or backward. But the legs in (b) and (c) are constrained to have designated postures. Therefore, two legs assembled in the same hip joint can swing toward opposite directions from gravity direction with equal angles. If these constraints are oppositely given to right and left hip joints, difference between the short and long strides makes the robot possible to change locomotion direction by making leg ends slide on the ground.

Fundamental leg postures are illustrated in Fig. 3. Amount of the stride is adjustable by controlling leg angle γ as a function of the arm's angular shift θ_r. Constraint in (a) is useful to keep legs always parallel for the toe-parallel in Fig. 2. On the other hand, the constraints in (b) and (c) are useful to generate the toe-in and toe-out, respectively. Visual sketch in Fig. 1 depicts the case when the short and long strides are produced at left and right hips, respectively to change walking direction toward left. In order to change leg postures while the arm rotates, it is recommended to switch these patterns when the arm becomes vertical since there is no interference among these constraints. Of course, leg ends are allowed to slide on the ground so that horizontal stumble disappears. Optional patterns for leg postures are considered in such condition that γ is uniquely determined by a periodic function of θ_r with a period 2π[rad]. These functions are found and written as follows.

$$\gamma = \gamma_{\max} \cos\{\theta_r\} \tag{1}$$

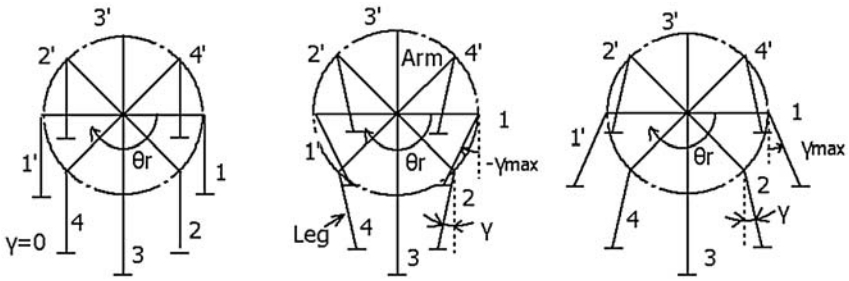


Fig. 3. Three fundamental leg posture patterns, that is, toe-parallel, toe-in, and toe-out in (**a**), (**b**), and (**c**), respectively

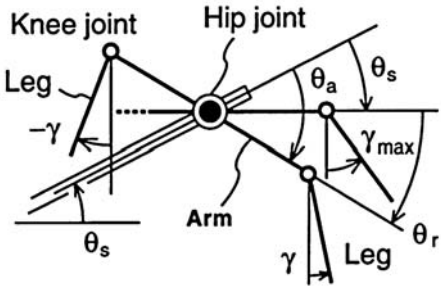


Fig. 4. Definition of angular parameters

$$\gamma = \gamma_{\max} \sin\{2\theta_r\} \tag{2}$$

$$\gamma = \gamma_{\max} \cos\{3\theta_r\} \tag{3}$$

$$\gamma = \tan^{-1}\{\tan\gamma_{\max} \cos\theta_r/(1 + \tan\gamma_{\max} \sin\theta_r)\} \tag{4}$$

where the parameters $\theta_a, \theta_r, \theta_s$, and γ are defined in Fig. 4. The parameters θ_r and γ stand for angles of the arm from horizontal and leg from vertical directions, respectively. Arm angle θ_r is equal to $\theta_a - \theta_s$. θ_s is the inclination angle of the ground. Amounts of θ_a and θ_r vary cyclically, but $|\theta_s|$ is less than $\pi/4$ [rad] on the road surface, in general. $\gamma_{\max}$ is a constant in $-\pi/4 < \gamma_{\max} <$ [rad]$\pi/4$ irrelevantly with θ_s, but flexible to accept a request for adjusting walking rhythm. In all functions, γ takes $\gamma_{\max}$ when θ_r is $0, \pi, 2\pi, \ldots (= n\pi)$ with some exception. Also, γ takes zero when θ_r is $n\pi \pm \pi/2$.

Above-mentioned leg postures are visible in Fig. 5. In the figure, four patterns from (a) to (d) are obtained from equations from (1) to (4), respectively. Legs are overwritten under such condition that the rotating center, i.e. hip joint remains in a same position to make the continuous swing clear. Also, left and right postures are a set of toe-in and toe-out walk. Angular increments are all same to $\pi/9$ [rad]. Particularly, the pattern in Fig. 5d has such features

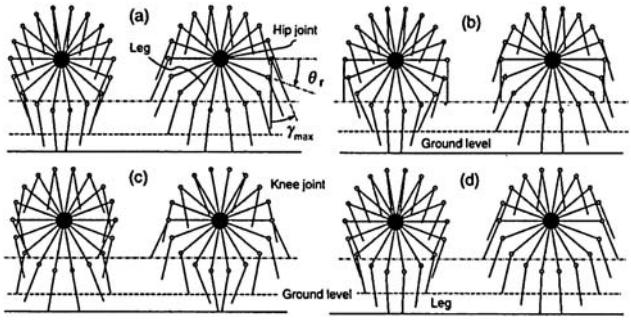


Fig. 5. Various patterns for changing leg posture

that all of the leg lines cross together on a single point at downside and upside for short and long strides, respectively.

4 Estimation of Leg Slide to Make Robot's Direction Change

To estimate the robot's walk or turn (spinning) by simulating leg slide with a numerical method, we use the hypothesis that the leg slides so that total sliding energy becomes minimal.

4.1 Minimum Energy Consumption to Find Robot's Trajectory

We treat the model shown in Fig. 6 for analyzing steering of the PEOPLER. Left part in the figure shows projection of the robot on a horizontal plane. Therefore, lengths of arm and leg look as if they shrink or extend while the arm rotates. Note that the leg slide is dependent on environmental conditions of the leg end on the ground. Then, we assume the following.

1. Weight operating on each leg is equal, say W_i.
2. Friction coefficient of each leg is same, say μ_i.
3. Robot center moves by satisfying that the sum of energy consumption, say E, becomes minimal while legs slide on the ground.
4. Leg posture pattern is simply given by (1) in the condition $L \leq r$.

Let suppose that the amount of leg slide displacement is expressed as d_i for the i-th leg, and leg landing point is expressed in the co-ordinates (x, y) with an angular parameter α elated to robot's gravity center G (normally equal to geometrical center). Then total energy consumption is written as $E = \Sigma(\mu_i W_i d_i)^2$ $(i = 1\text{–}4)$. Notice that the initial values of the robot are (x_0, y_0) and α_0, (see Fig. 6). Then the equations $@E/@x = @E/@y = @E/@\alpha = 0$ are valid to find the next leg position (x_0', y_0') and angular shift α. In fact,

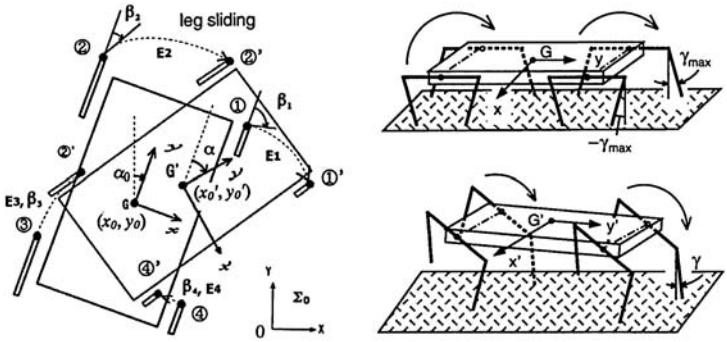


Fig. 6. Leg model for illustrating slide on the ground in different stride modes

$$\{x_0' = x_0 + rc\alpha_0 \pm rc\theta_r c\alpha \tag{5}$$

$$\{y_0' = y_0 + rs\alpha_0 \pm rc\theta_r s\alpha \tag{6}$$

$$\alpha = \tan^{-1}(K_1/K_2) \tag{7}$$

where

$$K_1 = (b_L^2 + b_w^2)s\alpha_0 + D\ b_w c\alpha_0 + 2L(D\ s\alpha_0 + b_w c\alpha_0)s\gamma \tag{8}$$

$$K_2 = (b_L^2 + b_w^2)c\alpha_0 + D\ b_w s\alpha_0 + 2\ L(D\ c\alpha_0 - b_w s\alpha_0)s\gamma \tag{9}$$

Symbols s and c are sine and cosine functions. D stands for the difference between right and left amount of strides. Notice that the arm rotates cyclically, then the gravity center position (x_n, y_n) expressed in place of (x_0, y_0), in general, is calculated and walk direction from Y-axis, say α_r is written as $n\phi$, where ϕ means one angular cycle of the robot turn. Step value 1 is added to or subtracted from the counter number n when the arm rotates π[rad] in CW or CCW, respectively. That is, n is written as $Int\ (\theta_r \pi)$.

4.2 Simulation Results of the Steering

Simulation results are depicted on the screen as projection of the robot on the Σ_0 plane (see Fig. 7). Regarding with four legs, both angular shift β_i and slide shift d_i are shown in Fig. 8. In this case, parameters are such that $b_L = 100, b_W = 50, L = r = 20, \gamma_{\max} = \pi/6$[rad] (this is transferred to left leg, but same amount with opposite sign to right leg). The circled number identifies locations of each leg (see Fig. 6). Figure 6 shows the characteristics of α_r

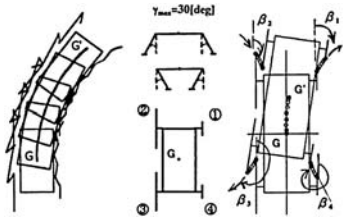


Fig. 7. Simulation of the steering

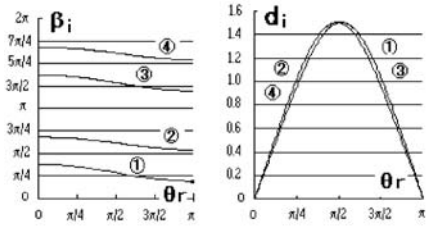


Fig. 8. Simulation results on each leg trajectories

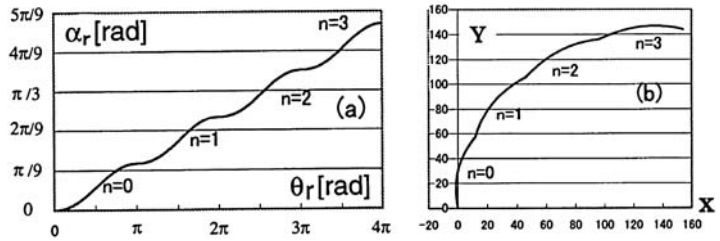


Fig. 9. Result of simulation ($b_L = 50, b_W = 50, L = 20, r = 30$). Characteristics of α_r versus θ_r is in (**a**). Locus of the gravity centre on (x, y) plane is in (**b**)

versus θ_r in the left and trajectories of the robot center in the right. From these figures, continuous trajectory is understandable and we can confirm that the PEOPLER turns its direction with the angle ϕ about 25[deg] in every arm rotation cycle, i.e. π[rad]. Also, it is observed that the trajectory of the gravity center is not a single curve since robot's backside displaces toward opposite direction in its initial stage.

4.3 Principal Factors Determining Robot Walk Direction

Obviously, the trajectories are different depending on the value of D which is determined by leg posture function with the values of L, r, and θ_r. In addition, such physical specifications of the robot like b_L and b_w have direct affect on the trajectories. It may be possible to steer by adjusting lengths of arm or leg. However, these adjustments are not practical since the PEOPLER causes a rolling to make walking unstable.

5 Experimental Set and its Ddemonstration for Steering

Since the PEOPLER Mk-1(10) is not powerful to make legs slide on the ground, we fabricated a lightweight miniature Mk-2 as an experimental set for demonstrating the steering.

5.1 Generation of Reciprocal Leg Motion

Design sketch of the PEOPLER Mk-2 is shown in Fig. 10. Specifications are such that $b_L = 23.0$[cm], $b_w = 21.3$[cm], $L = 4.7$[cm], $r = 4.0$[cm], and $\gamma_{\max} = 8$[deg]. The leg posture pattern is generated in mechanical synchronism. Actually, eccentric sheave driven by hip joint axis generates reciprocal motion for leg swing. Detailed mechanisms are shown in Fig. 11. The pattern is roughly similar to that given by (4) and it is inevitable to have small error between open and close angles. The Mk-2 is overviewed in Fig. 12. It is simpler than the Mk-1 in mechanism since its aim is to demonstrate steering in a

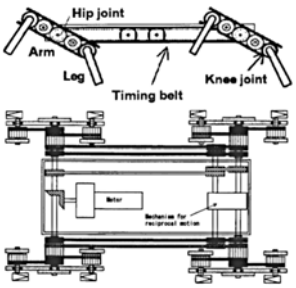


Fig. 10. Design sketch of the Mk-2

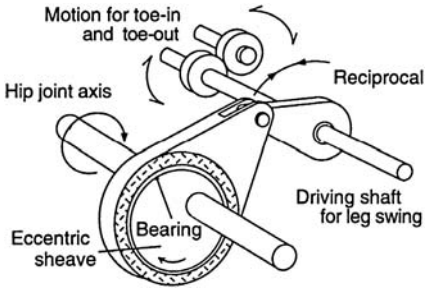


Fig. 11. Reciprocal motion for leg posture control

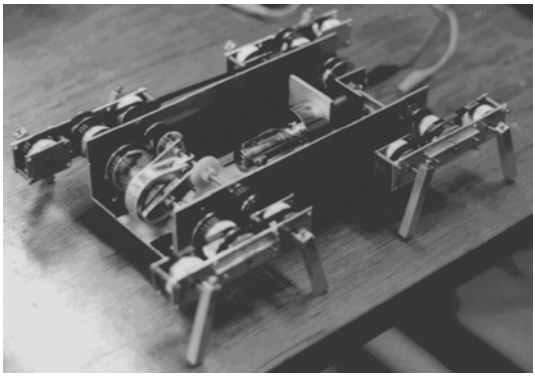


Fig. 12. Outlook of the Mk-2

fixed direction angle. Arms and legs are activated by one DC motor (Olympus CL3B, 15V, 1W) with reduction gear G-5 (1/625). Timing belts and gears are used. The robot weighs 1.6 [kg].

5.2 Demonstration of Walking and Steering

The Mk-2 walked on a flat place (see Fig. 12). And we could observe that there is no swings of pitch, yaw, and roll. Of course there are vertical shakes and irregular velocity synchronized with the hip joint rotation. However, the robot changed locomotion direction steadily toward back and forth while it walks. From the demonstration of walking toward right by making long stride to left and short stride to right, i.e. $0 < \gamma_{\max}$, we could collect positional and angular displacements in every arm rotation cycle. These are shown in Fig. 13. Black rectangle points initial leg in stance phase. And after sliding while the arm rotates π[rad], the leg terminates at the white rectangle. Then, the leg changes to swing phase. At the same time the other leg which were in swing phase takes its turn to have stance phase depicted by next black rectangle. This process is repetitively continued. Amounts of translation and rotation are 7.7[cm] and 8.5[deg](=ϕ), respectively.

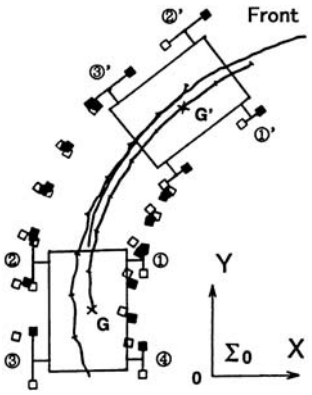


Fig. 13. Experimental results on the trajectories

Three markers are housed in cylindrical tubes so that they can move up and down freely and trace their positions at front, center and rear parts of the robot. Phase turns are discriminated by hand-written marks which are seen on the trajectories Angular displacement of legs β_i versus θ_r is calculated based on positions of black and white rectangles since the form of each leg's sliding became almost linear. In fact, such data are collected that $\beta_1 = 38.9, \beta_2 = 138.8, \beta_3 = 245.2, \beta_4 = 308.6$ in degree. This implies an important fact that all legs slide almost on the line passing the robot center G. Radius of the robot revolution is about 45[cm]. Analytical data are not overwritten on the sheet, however experimental results seem to be quite similar to those of simulation. Demonstration of spinning is also performed in other experiments, and it is confirmed that analytical and experimental trajectories are quite close. The PEOPLER fabricated for this demonstration drives left and right arms to

opposite directions in synchronism. Short and long strides made the spinning angle small and large, respectively.

6 Concluding Remarks

To make it possible for a legged robot to walk not only toward back and forth, but also toward different directions, we proposed the synchronized leg control for long stride and short stride. Such undesirable swings like pitch, yaw, and roll are completely disappeared by synchronizing control among arms and legs. Since leg slide is inevitable in the direction change of the robot, we considered trajectories of legs and robot's gravity center under reasonable physical assumptions. Then we succeeded in getting them by minimizing the function expressing total energy consumption. Trajectories of all legs are also clarified. These analytical results are proved to be similar to those demonstrated by the experimental set, PEOPLER Mk-2. For controlling leg posture, we proposed four periodic functions. Also, we showed that the direction of the robot walk is affected by robot's dimensional parameters.

The ability of changing the amount of stride is quite important to make the robot walk over obstacles without bumping. In addition, the ability of changing locomotion direction is important to detour obstacles. The robot Mk-2 showed actually that synchronized leg posture control is quite beneficial for mobility advancement. It has no flexibility and intelligence to make such a plan to determine appropriate amount of stride and direction angle in real environment. Therefore, our next subject is to design a full-scaled robot controlled by an onboard computer with battery for autonomy. Moreover, it became clear that leg's sliding direction is within narrow range except in spinning. This knowledge implies us to attach rollers at leg end so that the robot can turn easily to minimize power consumption. These studies are our future works.

References

1. Taguchi K, Hisakata H (1997) A Study of the Wheel-Feet Mechanism for Stair Climbing. J. of RSJ, Vol. 15, no. 1, pp. 118–123
2. Hirose S, Yoneda K, Tsukagoshi H (1997) TITAN VII: Quadruped Walking and Manipulating Robot on a Steep Slope. Proc. ICRA, pp. 494–500
3. Kaneko M, Abe M, Tanie K (1985) A Hexapad Walking Machine with Decoupled Freedom. IEEE J. of R&A, RA-1, 4, pp. 183–190
4. Krotkov E, Simmons R, Whittaker WRL (1993) Autonomous Walking Results with the Ambler Hexapod Planetary Rovers. Poc. Int. Conf. on Intelligent Autonomous Systems, pp. 46–53
5. Bourbakis NG (1998) Kydonas – An Autonomous Hybrid Robot: Waking and Climbing. IEEE R&A Magazine, June, pp. 52–59

6. Zhang L, Kitagawa A, et al (2001) Development of a Water Hydraulic Fire-Fighting Robot with Wavy Movement. Proc. INTERMAC Joint Technical Conf., Tokyo
7. Quinn RD, et al (2001) Insect Designs for Improved Robot Mobility. Proc. of 4th Int. Conf. on Climbing and Walking Robots (CLAWAR), Karlsruhe, Sep., pp. 69–76
8. Moore EZ, Buehler M, Stable Stair Climbing in a Simple Hexapad Robot, ibid., pp. 603–609
9. Schmucker U, Schneider A, Rusin V (2003) Interactive virtual simulator (IVS) of six-legged robot 'Katharina'. Proc. of 6th Int. Conf. on CLAWAR, Catania, Sep., pp. 327–332
10. Okada T, Hirokawa Y, Sakai T, Development of a rotating four-legged robot, PEPLER for walking on irregular terrain, ibid., pp. 593–600

Neuro-Controllers for Walking Machines – An Evolutionary Approach to Robust Behavior

Joern Fischer, Frank Pasemann, and Poramate Manoonpong

Fraunhofer Gesellschaft, Department of Autonomous Intelligent Systems (AIS.INDY), Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
joern.fischer@ais.fhg.de, frank.pasemann@ais.fhg.de, poramate.manoonpong@ais.fhg.de

On a first view complex machines with many degrees of freedom seem to need a complex control structure. Articles about the architecture of the controllers of walking machines, which probably belong to this category, often confirm this prejudice [3, 8]. Using evolutionary techniques to obtain small neural controllers seems much more promising [1, 2, 4, 5]. In the following article we present a method to evolve neural controllers keeping the complexity of the networks and the expense to evolve them small. We analyze the neural entities in order to understand them and to be able to prevent an unwanted behavior. As a last proof the resulting networks are tested on the real walking machine to find out the limitations of the physical simulation environment as well as of the neural controller. Our real platform is a six-legged walking machine called Morpheus. The task it performs is an obstacle avoidance behavior in an unknown environment.

1 Introduction

To evolve and to understand efficient neural controllers for physical walking machines is still a challenge in the area of robotics [5]. Many attempts to evolve such controllers end up in networks, which are too large to be understood in detail. Our aim is to evolve small, simple and robust networks, which can be coupled together in order to get a more complex behavior. This modular neurodynamics approach is used together with a physical simulation environment to evolve controllers for real world tasks. Here this procedure is demonstrated for a six-legged robot, called Morpheus. The following chapter describes the main technical specifications of the platform. The next one starts with the evolution of behavior in a physical simulation environment, which then leads to an implementation on the real platform. Finally, the results are discussed and an outlook is given.

2 The Research Platform: Morpheus

Morpheus is a six-legged robot, where each leg has two degrees of freedom. Each joint is controlled by a strong full metal servomotor. Like on real insects and scorpions the alignment of the feet is kept on an ellipsoid, which helps the robot, when it is turning around and enhances its stability. The levers are kept short and the body is narrow to ensure optimal force exploitation. The building blocks are kept modular to be able to change components very quickly and to enable the construction of different morphologies. Each of Morpheus joints has a potentiometer, which is used as a sensor for the actual leg position. In addition an infrared sensor is placed on a pivoting head on the front side of the robots body. All in all Morpheus has 13 active degrees of freedom and 13 sensors (the head motor does not return an angle value). The control of the robot is kept on a simple but powerful board, the "MBoard", which is able to control up to 32 motors, and which has 36 analog sensor inputs and a size of 130 mm $\times$ 42 mm.

3 Evolving Neurocontrollers for Walking

To use an evolutionary approach effectively it is necessary to have a physical simulation environment, which simulates the robot together with the environment as fast and as exact as possible. Our simulator is based on ODE (Open Dynamics Engine)[a] and enables an implementation, which is faster than real time and which is precise enough to mirrow corresponding behavior of a physical robot. This simulation environment is connected to our Integrated Structure Evolution Environment (ISEE). Structure evolution refers to the fact that the networks architecture is not fixed and that the number of neurons as well as of connections can be enlarged or reduced during the evolutionary process [7]. To preserve parts of a neural network which have already an appropriate performance, it is possible to fix these structures, and to continue the evolution by adding neurons or connections or removing parts of the network which are not fixed.

After having implemented Morpheus in the physical simulator one has to define a suitable fitness function, which in the simplest case may be given by the distance the robot walked during a given time interval. The desired neurocontroller for the pure walking task has no input neurons and 12 output neurons providing the motor signals. For simplicity connections between output neurons are suppressed in the first experiment. The simulation environment is a simple plane on which Morpheus should learn to walk along a straight line. The simplest solution for the hidden layer was found to be a quasi-periodic oscillator composed of two neurons [6]. By using symmetric output weights a typical tripod gate was obtained, which enabled an efficient forward movement.

[a] see also: http://opende.sourceforge.net/

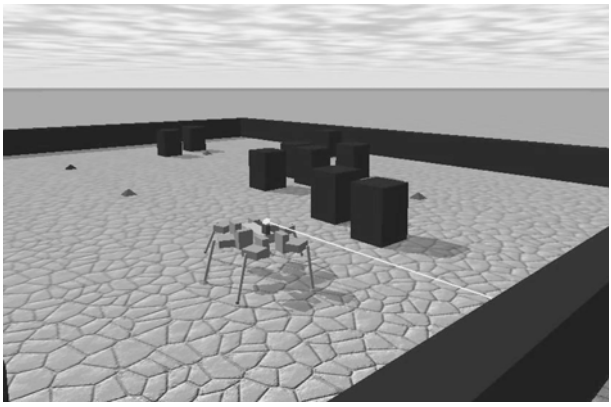


Fig. 1. The simulated robot surviving by performing an obstacle avoidance behavior

3.1 Evolving Neurocontrollers for Obstacle Avoidance

For the second task, obstacle avoidance, is endowed with an additional motor, turning an infrared sensor left and right in an oscillatory motion. Furthermore, one has to insert obstacles into the simulated environment. The difficulty with environments for obstacle avoidance is to define a suitable fitness function. The robot should move as far as it can, but to avoid obstacles may start to walk in circles. This should of course be prevented. The simplest way to define the fitness is to take the Euclidean distance from the start to the end point of the robots trajectory, and to let it run in an environment, called the "witches cauldron" (comp. Fig. 2).

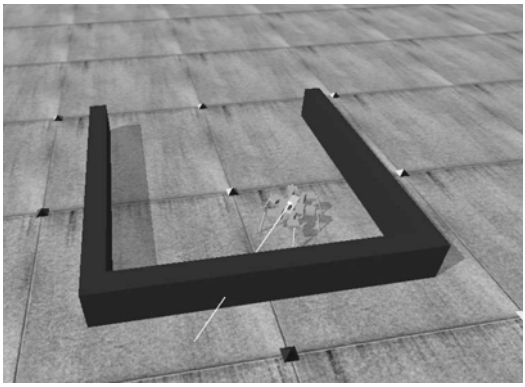


Fig. 2. For each episode only one of the 4 walls is left out at random. With this strategy only those individuals survive, which are able to avoid the walls. The fitness function is simplified to the Euclidean distance of the robots movement

This environment consists of walls around the starting point, with one of the sides open. This opening is chosen at random at the beginning of each episode. Then each individual has to pass through the opening in order to maximize its fitness value. To reuse the controller evolved first for the tripod gait, its structure is fixed and used for a starting population of obstacle avoidance controllers. Various results were obtained by the evolution process, performing more or less good, but to describe the mechanisms will go beyond the scope of this article. The simplest network with a reasonable performance is shown in Fig. 3. The infrared sensor directly inhibits the neurons, which are responsible for the movement of the upper leg motors of the right side. If an obstacle is detected, the amplitude of the movement of these motors is scaled down and the robot turns to the right side in order to avoid the obstacle.

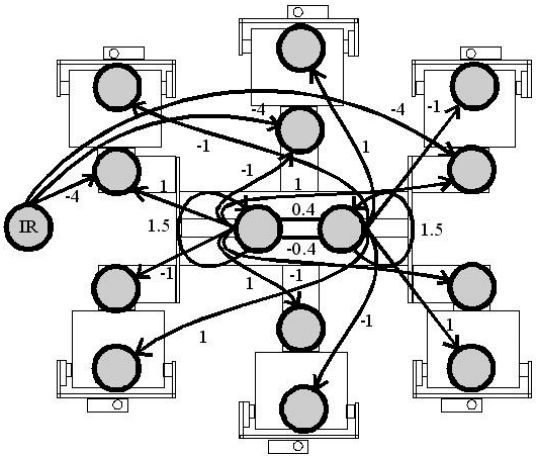


Fig. 3. This is the simplest neural network we evolved for an obstacle avoidance behavior. The input neuron and the 13 output neurons were predefined and should not have recurrent connections. With only two hidden neurons Morpheus walks forward. The infrared sensor is directly connected to the neurons for the upper motors on the right side of Morpheus. If an obstacle is detected these neurons are driven into saturation, the concerning motors move with lower amplitude and the robot turns to the left

4 Implementation on the Physical Machine

The network shown in Fig. 3 is first tested on Morpheus with a serial connection from a 1.8 Ghz personal computer and later it is implemented as an assembler program directly into the robots microcontroller. In both cases the network was updated with 50 Hz. This limit was given by the servo controller boards which work with a PWM signal of a period of 20 ms. The behavior was

the same in both cases, though the autonomous version calculated the output and the synaptic weights each in a range of only one byte. Both systems showed a good performance, even in more complex environments. Difficulties appeared only for legs of chairs or desks. Figure 4 shows a typical behavior of the six-legged robot avoiding an unknown obstacle.

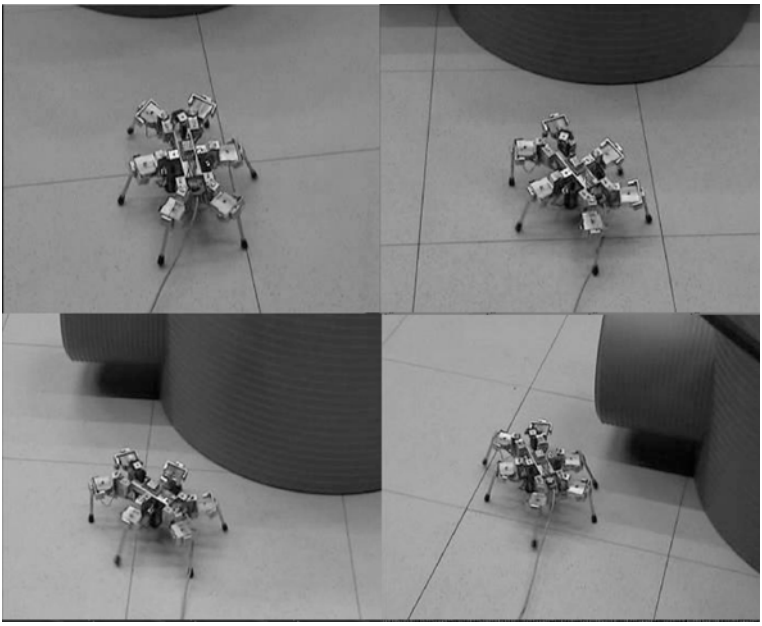


Fig. 4. The obstacle avoidance behavior of the six legged walking machine Morpheus

5 Conclusion and Outlook

With a physical simulation environment and our structure evolution algorithm we evolve networks for different tasks. We begin with a task of low complexity and analyze the resulting networks. We decide which weights should be fixed and after increasing the task difficulty we continue with the evolution by adding or removing neurons from the network and evaluating the new individuals. We obtain small and robust networks, which may be understood in their functionality as well as in their dynamical properties. This method is demonstrated on the six-legged robot Morpheus performing an obstacle avoidance task. In our future research we will work on the fusion of neural entities and of new methods to analyze time discrete neural networks.

References

1. Beer, R.D., Chiel, H.J. and Gallagher, J.C. (1999) Evolution and analysis of model CPGs for walking II. General principles and individual variability. J. Computational Neuroscience 7(2):119–147
2. Calvitti, A. and Beer, R.D. (2000) Analysis of a distributed model of Leg coordination. I. Individual coordination mechanisms. Biological Cybernetics 82:197–206
3. Cruse, H., Dean, J., Kindermann, T., Schmitz, J., Schumm, M. (1999) Walknet – a decentralized architecture for the control of walking behavior based on insect studies In: Hybrid Information Processing in Adaptive Autonomous Vehicles. (ed) G. Palm. Springer
4. Gruau, F. (1995) Automatic definition of modular neural networks, Ph.D. thesis, Ecole Normale Superieure de Lyon, Adaptive Behavior 3(2), 151–184.
5. Parker G.B., Li Z. (2003) Evolving Neural Networks for Hexapod Leg Controllers, Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(pp. 1376–1381).
6. Pasemann, F., Hild, M., Zahedi, K., SO(2)-Networks as Neural Oscillators, Lecture Notes In Computer Science, Biological and Artificial computation: Methodologies, Neural Modeling and Bioengineereng Applications, IWANN 7'th 2003
7. Pasemann, F., Steinmetz, U. Hülse, M. and Lara (2001) B. "Robot Control and the Evolution of Modular Neurodynamics", Theory in Biosciences, 120, pp. 311–326
8. Schmitz, J., Dean, J., Kindermann, Th., Schumm, M., Cruse, H. (2001) A biologically inspired controller for hexapod walking: simple solutions by exploiting physical properties. Biol. Bull. 200, 195–200 (invited paper)

Decentralized Dynamic Force Distribution for Multi-legged Locomotion

T. Odashima[1] and Z.W. Luo[1]

Bio-mimetic Control Research Center, RIKEN
odashima@bmc.riken.jp
luo@bmc.riken.jp

Multi-legged locomotion enables a mobile robot to behave flexibly in complex environment. In order to realize such flexible locomotion, however, high computational load will be imposed to the controller. In order to reduce the computational load, we propose a decentralized control approach. In this approach, the desired motion of the robot's body is converted to the local desired value of each leg. By using local force feedbacks within local controllers for each leg, it realizes the desired motion of the robot's body.

1 Introduction

Multi-legged locomotion makes it possible for a robot to move flexibly in complex environment. However, in order to realize the multi-legged locomotion, it is necessary to solve not only the locomotion pattern generation and planning problems but also the real time dynamic control problem. The difficulties come from the robot's redundant motion degrees of freedom as well as the uncertainties of the environmental geometry. Therefore, to control many activated degrees of freedom of a multi-legged robot results in a large computational load. One of the most feasible solutions to reduce the load is to build a decentralized control system. Inspired from the hierarchical nervous structure and CPG found in animals, in our previous work, we proposed our hierarchical decentralized locomotion generation approach and implemented it into a modular multi-legged robot. The effectiveness was shown in the reference [1], where the robot can walk along arbitral paths on a flat plane. However, since the leg motions were regulated by only position control, it was impossible to make sure the interaction between the robot's legs and the ground, as the result, the robot has poor adaptability to the disturbance from the roughness of the uneven terrain. In order to overcome this problem and to increase the robot's environmental adaptability, in this paper we propose a novel decentralized force control approach.

Force control is widely studied for a robot to interact with its environment physically [2, 3]. The same situation happened in the multi-legged locomotion. As seen from the body motion of the robot, dynamic control of a multi-legged locomotion can be treated as the coordinated control problem of a closed chain parallel mechanism from the legs at stance phase. In the problem the force distribution problem between each leg arises, which basically should be solved based on the total robot's global information [4, 5]. Therefore it has been considered that the decentralized control is not suitable for treating the force distribution problem. In this paper, as one solution of force distribution problem, we realize desired body motion via decentralized local force control of each leg in a bottom up manner.

2 Stracture and Modeling of the Robot

We developed a new module robot named "MoNOLeg" as shown in Fig. 1. The basic hierarchical control structure and modular construction of MoNOLeg is the same as the robot discussed in our previous paper [1]. As a lower layer of this robot, it is constructed from 6 homogeneous modules in which there are local controllers to perform the force control of each leg. In the upper layer, one controller gets the body states and plans body motion. Except in initialization, the upper controller always sends the same information to all lower layer controllers. It is expected that such a modular structure may produce some industrial advantages.

In the simulation shown in later section, we use the same parameters and functions with our real robot.

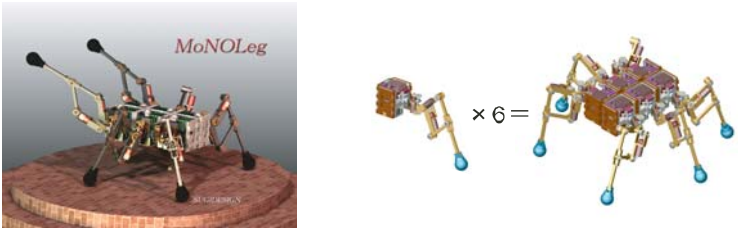


Fig. 1. Modular multi-legged robot "MoNOLeg" in six modules case

2.1 Mechanical Structure

As shown in Fig. 2, the mechanism of one leg has four activated joints. For simplicity, in this paper one of them is locked. Since the ball-shaped tip part of the leg contacts with the ground at one point, the constraint on the ground can be regarded as ball-socket joint during stance phase when there is no

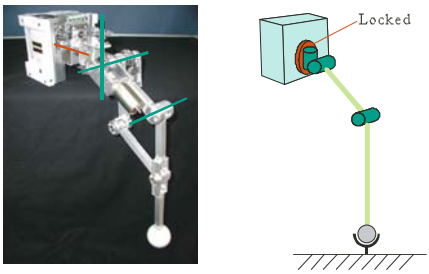


Fig. 2. Mechanical structure of a leg module

slip. Each leg, which has three active joints and three passive joints in stance phase, supports the body at the hip. If a mass of the leg is negligible or the angular velocities of joints are very low, no torque can be added at the hip joint in swing phase. The motion of the robot body is decided by a sum of the added force at hip from the legs in stance phase as shown in Fig. 3.

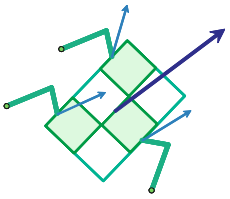


Fig. 3. Motion of a robot body via a sum of forces from three legs at stance phase

2.2 Nervous Oscillator

During walking, since the leg motion is a periodic movement, it is feasible to associate the leg state with the phase of a periodic oscillator. Figure 4 represents the relation between the leg state and the oscillator phase. In this figure, small circle in the left panel indicates the state variable of the oscillator and the value is θ. θ in the shaded part means that the leg state is in stance

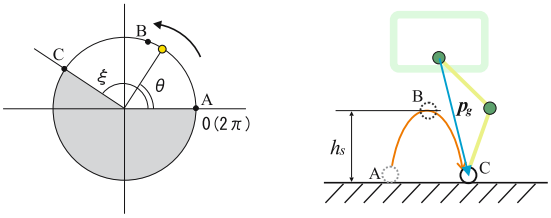


Fig. 4. Oscillator model

phase. ξ decides a duty factor β (the temporal ratio of stance phase to one locomotion period). The lower controllers interact locally with each other and adjust their own oscillator's phase according to phase differences between neighbors based on the following gradient dynamics.

$$\frac{d\theta_i}{dt} = \omega - \sum_p \frac{\partial W(\psi_{ip})}{\partial \psi_{ip}}\,, \tag{1}$$

where θ_i is the phase of the oscillator in ith module, ω is the constant angular velocity, $W(\psi_{ip})$ is the potential function defined as

$$W(\psi_{ip}) = -h \exp\left[-T\left\{1 - \cos(\psi_{ip} - \psi_{ipd})\right\}\right], \tag{2}$$

where T and h are constant values. ψ_{ip} in the (2) means the phase difference between thr ith module and its neighbor. Subscript p means a direction where neighbors exist, (forward, backward or side), ψ_{ipd} is the target value of ψ_{ip}. In the gradient system with potential given by (2), ψ_{ip} will converge to ψ_{ipd} [6]. Therefore, the phase of each oscillator and state of each leg is determined relatively against each other. Currently, ψ_{ipd}s for all i and all ξ_i are set to π, which guarantee a tripod gait because each leg is in anti-phase against its neighbors, and a duty factor becomes 0.5.

In the first half of swing phase, the tip of the leg is controlled to a constant height h_s, then the tip is controlled to a relative position $\mathbf{p}_g$, which is coordinated on the hip joint during remained swing phase. $\mathbf{p}_g$ is decided by each lower controller. In the stance phase, the leg is controlled by force control. The details of its reference signal are described in the next section.

3 Decentralized Force Distribution

The detailed decentralized force distribution is as follows. Initially, the upper controller informs all lower controllers their specific location and relative orientation between a coordinate system fixed on the robot body and on the each leg module as shown in Fig. 5(a). This notification is given only for one time, which does not increase the computational load for the upper controller during locomotion. After the lower controllers get the current linear and rotational velocity, $\mathbf{V}_B$ and $\mathbf{\Omega}_B$ of the robot body as in Fig. 5(b), they calculate their own linear velocity $\mathbf{V}_i$ at each hip as follows:

$$\mathbf{V}_i = \mathbf{V}_B + \mathbf{\Omega}_B \times \mathbf{r}_i \tag{3}$$

The upper controller also gives the desired values of body velocities $\mathbf{V}_{Bd}$ and $\mathbf{\Omega}_{Bd}$, height h_{Bd}, pitch θ_{pd} and roll θ_{rd} to the lower controllers so as the lower controllers calculate their respective desired linear velocity at each hip similarly as:

$$\mathbf{V}_{id} = \mathbf{V}_{Bd} + \mathbf{\Omega}_{Bd} \times \mathbf{r}_i \tag{4}$$

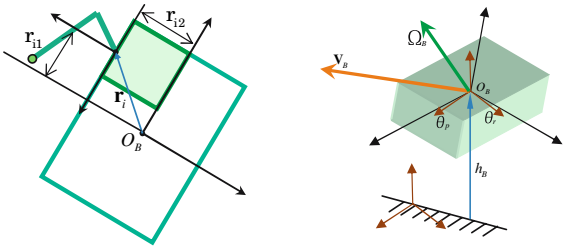


Fig. 5. Coordinates between the robot's body and its leg

The attribute of the robot body is basically controlled by the height of each hip joint. Each lower controller can calculate the current height of its hip h_i from the leg's joint angles as well as the stance conditions and controls it by proportional control. This impedance control plays a part of suspension of robot body and controls only the height direction. The reference input f_{iref} is,

$$f_{iref} = k_p(h_{id} - h_i)\,, \tag{5}$$

where h_{id} is the desired height for each module, which is calculated as follows from the desired body height and an inclination:

$$h_{id} = h_{Bd} + r_{i1}sin(\theta_{pd}) + r_{i2}sin(\theta_{rd})\,, \tag{6}$$

where r_{i*} is the distance between center of robot body and each hip and is an element of $\mathbf{r}_i$ (See Fig. 5(a)).

Lower controllers then generate the reference force signals for propelling the robot body by considering the gravity compensation term and the error between current and target velocity as follows:

$$\mathbf{F}_{Ri} = \mathbf{f}_{vi} + \mathbf{f}_{hi} + \mathbf{M}\mathbf{g}\,, \tag{7}$$

where $\mathbf{f}_{vi}$ and $\mathbf{f}_{hi}$ are the error compensation term for velocity and height, respectively. $\mathbf{M}$ is a inertia matrix of one module, and $\mathbf{g}$ is a gravitational vector. The $\mathbf{f}_{vi}$ and $\mathbf{f}_{hi}$ are expressed as follows, respectively

$$\mathbf{f}_{vi} = K_v(\mathbf{V}_{id} - \mathbf{T}_{Bd}^{-1}\mathbf{V}_i)\,, \tag{8}$$

$$\mathbf{f}_{hi} = [0, K_h(h_{id} - h_i), 0]^T, \tag{9}$$

where $\mathbf{T}_{Bd}$ is a translation matrix from the current orientation of robot body to target orientation decided by θ_{pd} and θ_{rd}. $\mathbf{F}_{Ri}$ is converted into joint torques by $-\mathbf{J}^T$. $\mathbf{J}$ is a jacobian, and the minus means that a reaction force from a ground is equal to $\mathbf{F}_{Ri}$.

Force control is specified to the leg so as to realize the desired velocity of (4). Therefore, unlike the position control approach, in which even if the leg joints reached the desired values, the legs may still failed in contact with the ground, force control increases the robot's adaptability to interact with uncertain rough ground. In Fig. 6, a block diagram of our method is shown.

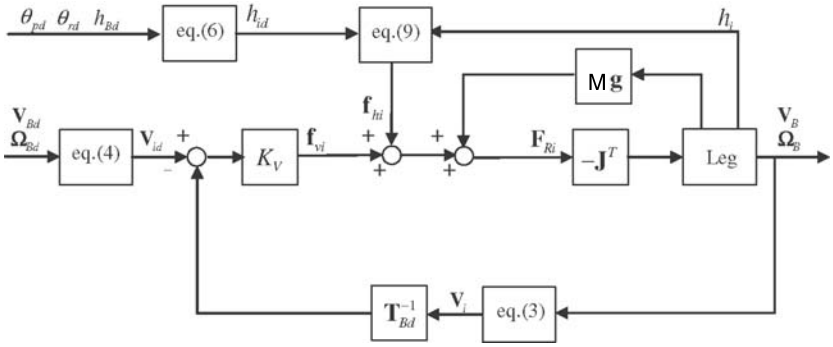


Fig. 6. Block diagram of the force control

4 Simulation Results

First of all, we confirmed the possibility of our method by dynamic simulation. The results are shown in Figs. 7 and 8 with command series given in Tables 1 and 2. These results are calculated under the condition that the force at each hip joint is added directly without leg. No forces are added in swing phase. If the desired forces include the element of gravitational direction, the element is made to be zero. We assume two optimization with these assumptions. First optimization is an ideal force control in lower controller level with mass less leg. Second is that no slippage happens at contact point. In these figures, bold line means the trajectory of the center of the robot body. The arrow shows the position and direction of a robot in every second. The unit of value on both axes is [m]. The coordinate system is fixed on the world. The simulation result

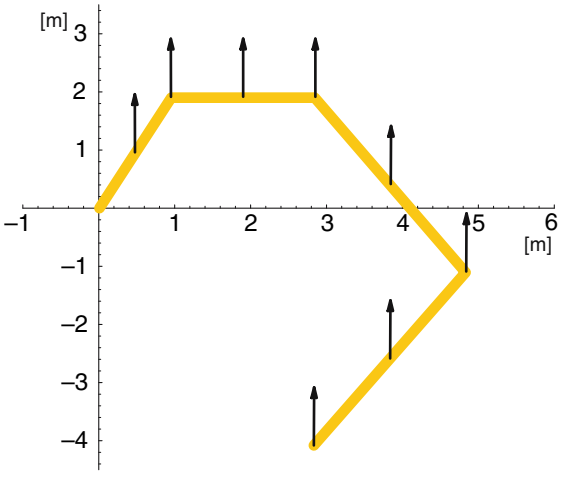


Fig. 7. Body trajectory in translational movement

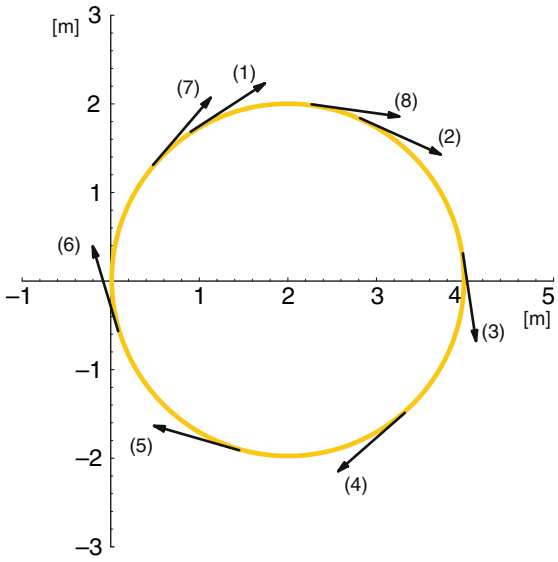


Fig. 8. Body trajectory in rotational movement

shows that the perturbation around yaw axis is suppressed well. Table 1 shows the time series of desired body velocity. During all movement, the desired height is constant, and desired pitch and roll angle are set as 0. In Fig. 8, the numbers indicated on the arrows show the time order of the robot's positions. The rotational angles along the circle made by the interval of the beginning points of two arrows correspond to the desired rotational velocity. The position of circle center is consistent with the point calculated from the command value [1].

Table 1. Commands for translational movements

	Go Forward	Step to Right	Rotate to Right
$t \leq 20$[s]	0.1	0.05	0.0
$20 < t \leq 40$	0.0	0.1	0.0
$40 < t \leq 60$	−0.15	0.1	0.0
$60 < t \leq 80$	−0.15	−0.1	0.0

In Fig. 9, the simulation result that the force is added at hip joint by leg is shown. These legs have a mass, and a slippage happens at the contact point. The bold line is the legged propulsion result generated by the command values shown in Table 1. In order to compare this result with an ideal case, the result shown in Fig. 7 is also drawn on same figure (fine line). Although the walking

Table 2. Commands for a rotational movement

	Go Forward	Step to Right	Rotate to Right
All time	0.2	0.0	0.1

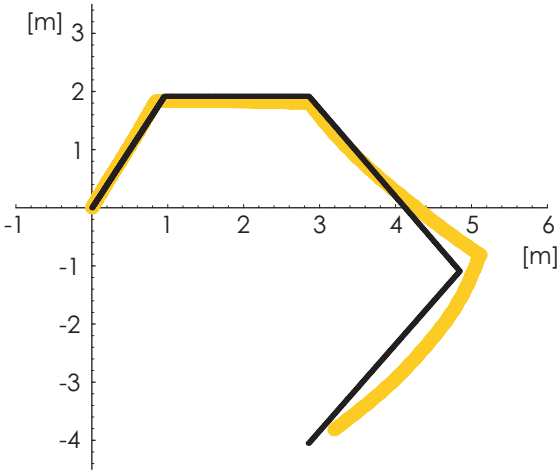


Fig. 9. Body trajectory in rotational movement

speed is not consistent with the commanded value because of slippage, it turns out that the walking direction is consistent with the commanded direction.

5 Conclusion

We proposed decentralized force distribution method and confirmed the performance of the method by dynamic simulator. Sometimes significant rotational error is resulted around yaw axis in simulation. The error comes from a power loss caused by slippage and/or less manipulability which depends on the posture of the stance legs. The more intelligent motion or planning in lower controllers can derive the precise motion. For example, to sense the slippage and control the normal reaction force from ground, or to decide the stance point ($\mathbf{p}_g$ see Fig. 4) based on the manipulability. These are very interesting themes, however, in our system structure, the upper controller can regulate the global position and direction by giving frequently the command instead of equipping lower controllers with high intelligence. Owing to hierarchical decentralized system, the additional computational cost for updating the commands is much smaller than the same cost in a centralized control system. To install this algorithm and check the performance in experiments using MoNOLeg is our next work.

References

1. T. Odashima, Z. Luo, Y. Kishi and S. Hosoe (2001) *Hierarchical control structure of a multi-legged robot for environmental adaptive locomotion*, Proc. CLAWAR 4th 105–112.
2. N. Hogan (1985) *Impedance Control: An Approach to Manipulation Part I, II, III*, ASME, J. Dynamic Systems, Measurement, and Control, 107-1:1–24.
3. M.H. Raibert and J.J. Craig (1981) *Hybrid Position/Force Control of Manipulators*, ASME, J. Dynamic Systems, Measurement, and Control, 103-2:126–133.
4. D. Zhou, K.H. Low and T. Zielinska (2000) *An efficient foot-force distribution algorithm for quadruped walking robots*, Robotica, 18:403–413.
5. C.A. Klein and T.S. Chung (1987) *Force Interaction and Allocation for the Legs of a Walking Vehicle*, J. Robotics and Automation, RA-3(6):546–555.
6. H. Yuasa and M. Ito (1990) *Coordination of Many Oscillators and Generation of Locomotory Patterns*, Biol. Cybern. 63:177–184.

An Outdoor Vehicle Control Method Based Body Configuration Information

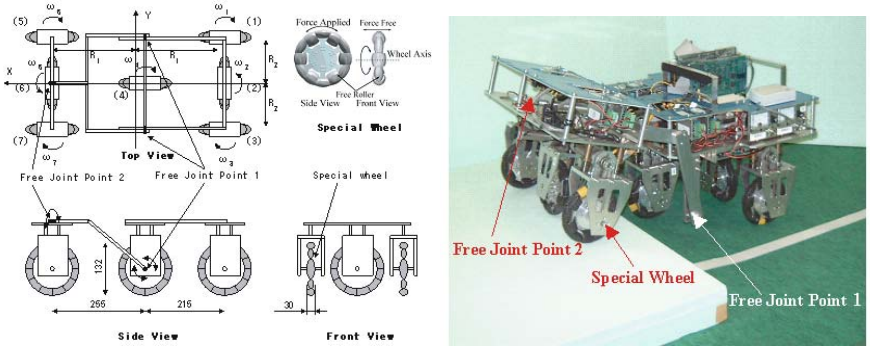
Daisuke Chugo[1], Kuniaki Kawabata[2], Hayato Kaetsu[3], Hajime Asama[4], and Taketoshi Mishima[5]

[1] Saitama University, 255, Shimo-Ookubo, Saimata-shi, Saitama, Japan
chugo@riken.jp
[2] RIKEN (The Institute of Physical and Chemical Research), 2-1, Hirosawa, Wako-shi, Saitama, Japan
kuniakik@riken.jp
[3] RIKEN (The Institute of Physical and Chemical Research), 2-1, Hirosawa, Wako-shi, Saitama, Japan
kaetsu@riken.jp
[4] The University of Tokyo, 4-6-1, Komaba, Meguro-ku, Tokyo, Japan
asama@race.u-tokyo.ac.jp
[5] Saitama University, 255, Shimo-Ookubo, Saimata-shi, Saitama, Japan
mishima@me.ics.saitama-u.ac.jp

In our current research, we are developing a holonomic mobile vehicle which is capable of running over the irregular terrain. Our developing vehicle realizes omni-directional motion on flat floor using special wheels and passes over non-flat ground using the passive suspension mechanism. This paper proposes a vehicle control method according to change of vehicle's body configuration for passing over the irregular terrain. The developed vehicle utilizes the passive suspension mechanism connected by two free joints that provide to change the body configuration on the terrain condition. Therefore, it is required to coordinate the suitable rotation velocity of each wheel according to the body configuration. In our previous work, the vehicle motion in sagital plane (2D) was discussed and the control reference calculation method based on the body configuration was proposed. However, there are various types of irregular terrain in the general environment and our method must be extended considering to pass over such environment. In this paper, we extended our method to adapt the 3D irregular terrain. The performance of our proposed method is verified by the experiments using our prototype vehicle.

1 Introduction

In recent years, mobile robot technologies are expected to perform various tasks in general environment such as nuclear power plants, large factories,



(a) Overview of the mechanism (b) Prototype Vehicle

Fig. 1. Our prototype mechanism

welfare care facilities and hospitals. However, there are narrow spaces with steps and slopes in such environments and the vehicle is required to have quick mobility for effective task execution. The omni-directional mobility is useful for moving in narrow spaces, because there is no holonomic constraint on its motion [1, 2]. Furthermore, the step-overcoming function is necessary when the vehicle runs in rough environment. Therefore, we are developing a holonomic omni-directional vehicle with step-climbing ability [5].

Our prototype mechanism consists of seven special wheels with free rollers (Fig. 1) and a passive suspension mechanism. The special wheel equips twelve cylindrical free rollers [3] and helps to generate the omni-directional motion with suitable wheel arrangement and wheel control. Furthermore, our mechanism utilizes new passive suspension system, which is more suitable for the step than general rocker-bogie suspensions [5, 7]. The free joint point 1 is in the same height as the axle and this system helps that the vehicle can pass over the step smoothly when the wheel contacts it. No sensors and no additional actuators are equipped to pass over the irregular terrain.

Our prototype has redundant actuations, therefore the vehicle controller calculates the control reference of each wheel based on the kinematic model and controls each actuator to take the synchronization among the wheels using PID-based control. When the vehicle overcomes the obstacles, the moving distance of each wheel is different because the body configuration and its kinematic model change. Therefore, it is required to modify the control reference referring to the change of its body configuration Unstable wheel control reference, which is derived without the consideration of body configuration, causes the wheel slippage or rotation error and these actions disturb the mobile performance of the vehicle.

In this paper, we propose a wheel control method, which is according to the body configuration. Our proposed method realizes that the vehicle passes over the irregular terrain using suitable control reference and improves its

performance. In our previous work, we discussed the overcoming motion of the vehicle in the advance direction and developed the adjusting method of wheel control reference referring to the modification of the body shape in sagital plane (2D) [7]. However, in general environment, there are many obstacles which have various forms and it is necessary to refer to the body configuration in all direction. Therefore, in this paper, we extend this method to three dimensions for increasing the mobile performance of the vehicle, especially when the vehicle overcomes the obstacle from slant direction.

2 Control System

2.1 Kinematics

Our vehicle has the passive suspension mechanism in its body and the body configuration changes according to the terrain condition when the vehicle passes over the irregular terrain. Therefore, it is required to modify the wheel control referring to its configuration.

We consider the relationship of rotation velocity vector of each wheel and the change of vehicle's body configuration. We assume that the vehicle has n passive linkages and all wheels have grounded and actuated. When the vehicle passes over the obstacle as shown in Fig. 2, the velocity vector of wheel $i+1$ is calculated by the velocity vector of wheel i and the rotation vector of wheel $i+1$ in (1).

$$\mathbf{v}_{i+1} = {}^{i}\mathbf{v}_i + {}^{i}\sigma_i \times {}^{i}\mathbf{P}^{i}_{i+1} \tag{1}$$

where i is the number of wheel ($i = 1 \cdots n$), ${}^{i}\mathbf{v}_i$ and ${}^{i}\sigma_i$ are the velocity vector and the rotation vector of wheel i on the coordinate system i, respectively. ${}^{i}\mathbf{P}^{i}_{i+1}$ is the position vector from the wheel i to wheel $i+1$.

The coordinate system of each wheel is defined as follows.

- The x-axis is defined in the drive direction of the wheel.
- The y-axis is defined in the perpendicular direction to the ground.

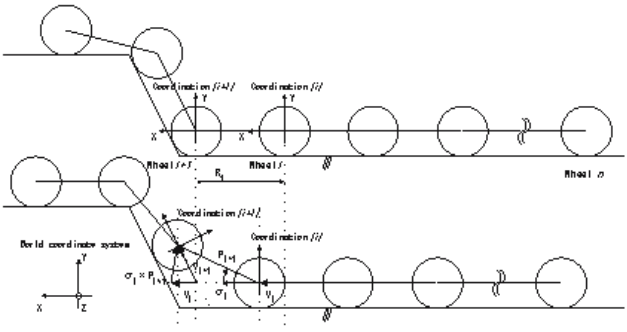


Fig. 2. Relationship between the velocity vector and the body

Thus, the x-direction ingredient of the velocity vector in the coordination {i} is derived as the control reference value. The control reference of the wheel $i+1$ (ω_{i+1}) is derived from (2) and (3).

$$\omega_{i+1} = \frac{\left\| {}^{i+1}\mathbf{v}_{i+1} \right\|_x}{r} + \left\| {}^{i}\sigma_i \right\|_z \tag{2}$$

$${}^{i+1}\mathbf{v}_{i+1} = {}^{i+1}_{i}\mathbf{R} \cdot {}^{i}\mathbf{v}_{i+1} \tag{3}$$

$\left\| {}^{i+1}\mathbf{v}_{i+1} \right\|_x$ is x ingredient of the wheel $i+1$ velocity vector, r is the radius of the wheel and ${}^{i+1}_{i}\mathbf{R}$ is the conversion matrix from the coordination $\{i\}$ to $\{i+1\}$. Thus, when the velocity vector and rotation vector of wheel i are defined as ${}^{i}\mathbf{v}_i$ and ${}^{i}\sigma_i$, the control reference of wheel $i+1$ is expressed as (4).

$$\omega_{i+1} = \frac{\left\| {}^{i+1}_{i}\mathbf{R} \cdot \left({}^{i}\mathbf{v}_i + {}^{i}\sigma_i \times {}^{i}\mathbf{P}^{i}_{i+1} \right) \right\|_x}{r} + \left\| {}^{i}\sigma_i \right\|_z \tag{4}$$

All wheels have grounded, therefore we can assume that each wheel grounds the plane. When the angle between the x-axis of coordination {i} and the one of coordination $\{i+1\}$ is α, the conversion matrix is derived as (5). α fulfills the (6).

$${}^{i+1}_{i}\mathbf{R} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$\left\| {}^{i+1}\mathbf{v}_{i+1} \right\|_y = 0 \tag{6}$$

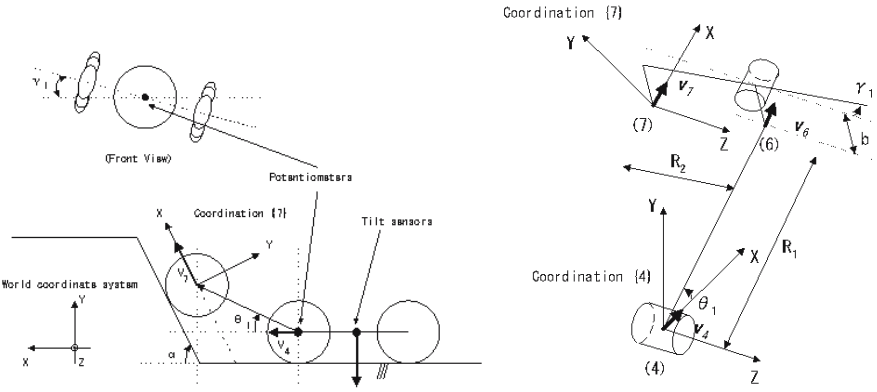
2.2 Adaptation to Control Reference

In previous section, we discuss the vehicle kinematics referring to the body configuration. In this section, we adapt it to our prototype vehicle. Our vehicle measures the change of body configuration using its attitude sensors and generates the wheel control reference with this information.

Our vehicle has two potentiometers on each passive joint and tilt sensors which are attached on the rear part of the vehicle body as shown in Fig. 3(a). We can measure the following angles using these sensors.

- The roll angle θ_1 and pitch angle γ_1 from potentiometers.
- The roll angle θ_2 and pitch angle γ_2 from tilt sensors.

Our developing vehicle has 7 wheels and all wheels has actuated. Figure 1(a) shows the definition of the wheel number (We display as wheel i: $i = 1 \cdots 7$), the coordinates, the length of each links, and the rotate speed of each wheel, respectively. $R1$ and $R2$ indicate the length of each links and $\omega_1, \ldots, \omega_7$ are the rotation velocity of each wheel.



(a) Side view (b) 3D View

Fig. 3. Coordination and parameters of our prototype

When the vehicle runs at v_0 in x-direction on the coordination {4}, the velocity vector of wheel 7 on the coordination {4} is derived as (7) by (1). The kinematic relationship among the wheels is shown in Fig. 3(b).

$$
\begin{aligned}
{}^4\mathbf{v}_7 &= {}^4\mathbf{v}_6 + {}^4\sigma_6 \times {}^4\mathbf{P}_7^6 = \left({}^4\mathbf{v}_4 + {}^4\sigma_4 \times {}^4\mathbf{P}_6^4\right) + {}^4\sigma_6 \times {}^4\mathbf{P}_7^6 \\
&= \begin{bmatrix} {}^4v_{7x} \\ {}^4v_{7y} \\ {}^4v_{7z} \end{bmatrix} = \begin{bmatrix} v_0 + \dot{\theta}_1\{-R_1 \sin\theta_1 + R_2 \cos\theta_1 \sin\gamma_1 - b\cos\theta_1 \\ \quad (1-\cos\gamma_1)\} \\ \dot{\theta}_1 \{R_1 \cos\theta_1 - R_2 \sin\theta_1 \sin\gamma_1 + b\sin\theta_1 (1-\cos\gamma_1)\} \\ \quad -\dot{\gamma}_1 (R_2 \cos\gamma_1 - b\sin\gamma_1) \\ -\dot{\gamma}_1 (R_2 \cos\theta_1 \sin\gamma_1 - b\cos\theta_1 (1-\cos\gamma_1)) \end{bmatrix}
\end{aligned} \tag{7}
$$

As the same, the velocities of wheel 1, 3 and 5 are derived.

$$
{}^4\mathbf{v}_1 = \begin{bmatrix} {}^4v_{1x} \\ {}^4v_{1y} \\ {}^4v_{1z} \end{bmatrix} = \begin{bmatrix} v_0 - \dot{\theta}_2\{R_1 \sin\theta_2 + R_2 \cos\theta_2 \sin\gamma_2 \\ \quad -b\cos\theta_2 (1-\cos\gamma_2)\} \\ \dot{\theta}_2 \{R_1 \cos\theta_2 + R_2 \sin\theta_2 \sin\gamma_2 - b\sin\theta_2 (1-\cos\gamma_2)\} \\ \quad +\dot{\gamma}_2 (R_2 \cos\gamma_2 - b\sin\gamma_2) \\ \dot{\gamma}_2 (R_2 \cos\theta_2 \sin\gamma_2 - b\cos\theta_2 (1-\cos\gamma_2)) \end{bmatrix} \tag{8}
$$

$$
{}^4\mathbf{v}_3 = \begin{bmatrix} {}^4v_{3x} \\ {}^4v_{3y} \\ {}^4v_{3z} \end{bmatrix} = \begin{bmatrix} v_0 + \dot{\theta}_2\{-R_1 \sin\theta_2 + R_2 \cos\theta_2 \sin\gamma_2 \\ \quad -b\cos\theta_2 (1-\cos\gamma_2)\} \\ \dot{\theta}_2 \{R_1 \cos\theta_2 - R_2 \sin\theta_2 \sin\gamma_2 + b\sin\theta_2 (1-\cos\gamma_2)\} \\ \quad -\dot{\gamma}_2 (R_2 \cos\gamma_2 - b\sin\gamma_2) \\ -\dot{\gamma}_2 (R_2 \cos\theta_2 \sin\gamma_2 - b\cos\theta_2 (1-\cos\gamma_2)) \end{bmatrix} \tag{9}
$$

$$
{}^4\mathbf{v}_5 = \begin{bmatrix} {}^4v_{5x} \\ {}^4v_{5y} \\ {}^4v_{5z} \end{bmatrix} = \begin{bmatrix} v_0 - \dot{\theta}_1\{R_1 \sin\theta_1 + R_2 \cos\theta_1 \sin\gamma_1 \\ \quad -b\cos\theta_1\,(1-\cos\gamma_1)\} \\ \dot{\theta}_1\{R_1 \cos\theta_1 + R_2 \sin\theta_1 \sin\gamma_1 - b\sin\theta_1\,(1-\cos\gamma_1)\} \\ \quad +\dot{\gamma}_1\,(R_2\cos\gamma_1 - b\sin\gamma_1) \\ \dot{\gamma}_1\,(R_2\cos\theta_1\sin\gamma_1 - b\cos\theta_1\,(1-\cos\gamma_1)) \end{bmatrix} \tag{10}
$$

The rotation vector of each wheel is derived using the roll and pitch angle as shown in (11) and (12).

$$
{}^4\sigma_5 = {}^4\sigma_7 = \begin{bmatrix} {}^4\sigma_{7x} & {}^4\sigma_{7y} & {}^4\sigma_{7z} \end{bmatrix}^T = \begin{bmatrix} \dot{\gamma}_1 & 0 & \dot{\theta}_1 \end{bmatrix}^T \tag{11}
$$

$$
{}^4\sigma_1 = {}^4\sigma_3 = \begin{bmatrix} {}^4\sigma_{3x} & {}^4\sigma_{3y} & {}^4\sigma_{3z} \end{bmatrix}^T = \begin{bmatrix} \dot{\gamma}_2 & 0 & \dot{\theta}_2 \end{bmatrix}^T \tag{12}
$$

As shown in (5), we assume that the obstacle is the α-degree slope about each wheel as shown in Fig. 3(a). The velocity vector and the control reference of wheel i is expressed in (13) and (14). The α-degree is defined in (15), because the x-axis is defined in the drive direction of the wheel and the velocity vector is parallel to the drive direction as (6). Our vehicle controls each wheel based on this control reference using PID based control system [7].

$$
{}^i\mathbf{v}_i = {}^i_4\mathbf{R} \cdot {}^4\mathbf{v}_i = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^4v_{ix} \\ {}^4v_{iy} \\ {}^4v_{iz} \end{bmatrix} = \begin{bmatrix} \cos\alpha \cdot {}^4v_{ix} - \sin\alpha \cdot {}^4v_{iy} \\ \sin\alpha \cdot {}^4v_{ix} + \cos\alpha \cdot {}^4v_{iy} \\ {}^4v_{iz} \end{bmatrix} \tag{13}
$$

$$
\omega_i = \frac{\cos\alpha \cdot {}^4v_{ix} - \sin\alpha \cdot {}^4v_{iy}}{r} + {}^4\sigma_{iz} \tag{14}
$$

$$
\sin\alpha \cdot {}^4v_{ix} + \cos\alpha \cdot {}^4v_{iy} = 0 \tag{15}
$$

3 Experiments

We verify the performance of our proposed method by two experiments. In experiment 1, we test our proposed method for passing over the step in forward direction and we verify the slippage and the rotation error which cause to reduce the overcoming performance. In experiment 2, we test that the vehicle runs in an imbalanced situation such as overcoming the step form slant direction.

3.1 Experiment 1

In experiment 1, the prototype vehicle passes over the step and we verify about two topics. One topic is the height of the step which the vehicle can climb up. The other topic is the slippage and the rotation error of the wheels

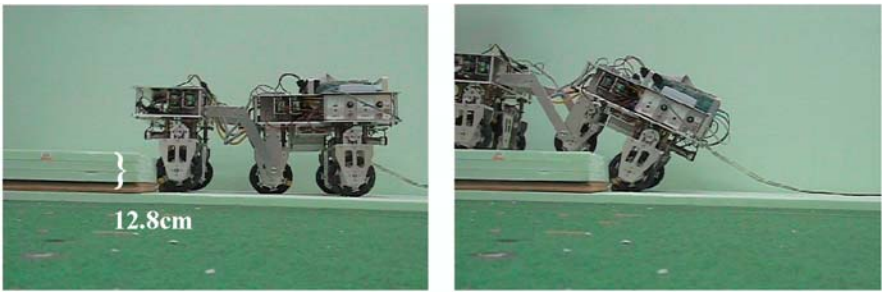


Fig. 4. Passing over the step

when the vehicle passes over 0.09 [m] height step. We compare our proposed method with standard method which does not refer to the body configuration. In both cases, PID based controller [7] for traction control is employed.

As the result of the experiment, the vehicle can pass over the 0.128[m] height step with our proposed method as shown in Fig. 4. With standard method, the vehicle can pass over the only 0.096[m] height step. When the vehicle passes over the 0.09[m], the slip ratio of the wheel decreases 53[%] and the disturbance ratio of the wheel ratio, which means the error ratio of the rotation velocity, decreases 27[%] by our proposed control method as shown in Table 1. The slip ratio [6] and the disturbance ratio [6] are calculated by (16) and (17), respectively.

$$\hat{s} = \frac{r\omega - v_\omega}{r\omega} \tag{16}$$

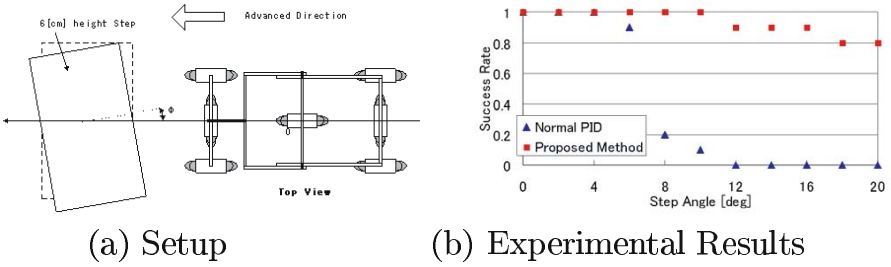
$$\hat{d} = \frac{\omega_{ref} - \omega}{\omega} \tag{17}$$

where ω is the rotation speed of the actuator and ω_{ref} is the reference value of wheel rotation velocity. r and v_ω indicate the radius of the wheel and the vehicle speed, respectively.

From these results, our proposed method reduces the slippage and the rotation error of the wheels. Furthermore, our method improves the vehicle's performance of the step-overcoming. Therefore, our proposed control method is effective for passing over the step.

Table 1. The slippage and the disturbance ratio

	Method	Front Wheel	Middle Wheel	Rear Wheel	Average
Slippage	Standard	16.8	15.9	14.4	15.7
Ratio	Proposed	12.8	12.0	10.4	11.7
Disturbance	Standard	11.6	9.1	11.4	10.7
Ratio	Proposed	6.2	5.1	7.0	6.1



(a) Setup (b) Experimental Results

Fig. 5. The step-overcoming from slant direction

3.2 Experiment 2

In experiment 2, the prototype vehicle passes over the step from slant direction as shown in Fig. 5(a). ϕ is the angle between the obstacle and the vehicle advanced direction. We change the step angle ϕ from 0 to 20 [degrees] by every 2[degrees] and we test the vehicle passes over the step ten times at each angle. We compare the result by our proposed method with the result utilizing standard PID controller, which doesn't consider the body shape.

As the result, with the standard method, the vehicle can cancel an imbalanced situation and overcome the obstacle only at 6[degrees] maximum as shown in Fig. 5(b). On the other hand, with our proposed method, the vehicle can pass over the obstacle at 20[degrees]. When the height of obstacle is 6[cm], maximum value of the obstacle angle ϕ is about 20 [degrees] which is mechanical determined. Therefore, we verify that our control method can adapt an imbalance and improve the vehicle mobility on rough terrain.

4 Conclusion

In this paper, we propose the vehicle control method according to change of vehicle's body shape. We discuss the kinematic model referring to the body configuration and adjusting method of the wheel control references when the vehicle passes over the rough terrain changing the body shape. Our proposed method can utilize for the vehicle which has passive suspension mechanism.

We verified the effectiveness of our proposed method by the experiments. Utilizing our proposed method, the slip ratio and the disturbance ratio reduce when the vehicle passes over the step. Not only in sagital plane but also in three dimensions, our method is effective. Suitable wheel control reference improves the step-climbing ability and its adaptation capability to the irregular terrain.

References

1. G. Campion, G. Bastin and B.D. Andrea-Novel. Structual Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. In: IEEE Trans. on Robotics and Automation, Vol. 12, No. 1, pp. 47–62, 1996.
2. M. Ichikawa. Wheel arrangements for Wheeled Vehicle. Journal of the Robotics Society of Japan, Vol. 13, No. 1, pp. 107–112, 1995.
3. H. Asama, et al. Development of an Omni-Directional Mobile Robot with 3 DOF Decoupling Drive Mechanism. In: Proc. of the 1995 IEEE Int. Conf. on Robotics and Automation, pp. 1925–1930, 1995.
4. Stone, H. W., Mars Pathfinder Microrover: A Low-Cost, Low-Power Spacecraft, Proceedings of the 1996 AIAA Forum on Advanced Developments in Space Robotics, 1996.
5. D. Chugo, et al. Development of omni-directional vehicle with step-climbing ability. In: Proc. of the 2003 IEEE Int. Conf. on Robotics & Automation, pp. 3849–3854, 2003.
6. K.Yoshida and H.Hamano. Motion Dynamic of a Rover With Slip-Based Traction Model. In: Proc. of the 2002 IEEE Int. Conf. on Robotics & Automation, pp. 3155–3160, 2001.
7. D. Chugo, et al. Vehicle Control Based on Body Configuration. In: Proc. of the 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, to appear, 2004.

Physically Variable Compliance in Running

Jonathan W. Hurst and Alfred A. Rizzi

The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh PA, 15213

Summary. This paper explores the role and utility of variable compliance in running behaviors. While it is well understood in the literature that leg compliance plays an important role in running locomotion, our conjecture is that mechanically adjustable leg compliance improves the efficiency and robustness of a running system. We claim that variable compliance can serve as an effective means for gait regulation, and can enable energy efficient locomotion over a wide range of terrains and speeds. We draw inspiration from a number of observations in biomechanics, and use qualitative arguments and a simulation of our variable compliance actuator to support the underlying ideas.

1 Introduction

Our long-term goal is to develop mechanisms and controllers that allow legged robots to run in a robust and efficient manner, over terrain that varies in geometry and dynamic properties. In this paper, we use a combination of literature review and qualitative and quantitative arguments to describe important aspects for implementing a successful running robot. We begin with a literature review showing that the Spring Loaded Inverted Pendulum (SLIP) model is a good representation of animal running as well as a useful framework for analysis and control. We then go on to assert, through literature and qualitative argument, that leg stiffness is a useful control parameter. Finally, we use quantitative arguments to show that a SLIP is most accurately and efficiently implemented using a mechanical spring, where the stiffness is tuned to a specific gait and terrain.

Our prototype Actuator with Mechanically Adjustable Series Compliance (AMASC) is an implementation of the ideas presented in this paper [1, 2]. A simulation of the AMASC, used as the knee actuation for a simulated running robot, is able to vary leg stiffness to influence its running gait. In addition, when the mechanical spring is tuned to match the desired leg stiffness, the simulation expends the lowest motor shaft power and is most energetically efficient.

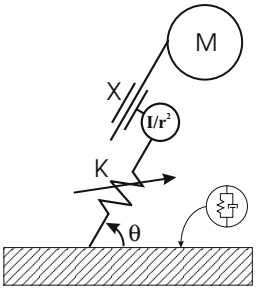


Fig. 1. The Spring Loaded Inverted Pendulum, with rotor inertia

2 Background

The SLIP model accurately describes the center of mass (CoM) motion of an animal in a running gait, regardless of the number of legs, the size of the animal, or the running gait employed [3–5]. Successful running robots also exhibit SLIP model behavior, such as the Planar Hopper, ARL Monopod II and CMU Bowleg Hopper, and most likely were designed specifically to do so [6–8].

2.1 Tuned Leg Stiffness

In general, carefully choosing leg stiffness is important, for a variety of reasons. Leg stiffness that is too high causes energetically wasteful ground deformations, high stresses in the body, and high acceleration of sensors or other sensitive components. Leg stiffness that is too low results in a long stance time and larger deflection of the leg, possibly reaching compression limitations of the leg. A long stance time is more expensive energetically, because the animal or robot must hold its weight against gravity for a longer period of time.

Tuned leg stiffness is also important for passive stability properties of legged locomotion. According to mathematical analysis of the SLIP model, for certain angles of attack, the spring-mass system becomes self-stabilized if the leg stiffness is properly adjusted and a minimum running speed is exceeded [9]. Similar results were observed in computer simulation of a running cockroach, showing passive stability in its running gait, and an optimal leg spring stiffness for maximum passive stability [10].

2.2 Using Leg Stiffness for Gait Control

Varying the leg stiffness is one good method for controlling a running gait. Only three terms are required to describe a cyclic running gait based on the SLIP model, and leg stiffness affects one or more of the three terms [11]. Control of a running gait using the three different terms (hopping height, leg

stiffness, and leg angle at touchdown) has been demonstrated experimentally [12].

Animals vary leg stiffness to control their running gait and alter their center of mass motion, within the constraints of the SLIP model. Researchers seek to better understand the role of compliance control in animal running through experiments limiting one parameter of the SLIP model. By restricting obvious control methods (such as hip angle), more obscure methods (such as leg stiffness) can be observed. Ferris and Farley restricted humans to hopping in place, removing hip angle as a control variable. When subjects were tested at constant ground stiffness but instructed to alter hopping frequency, subjects adjusted leg spring stiffness [3]. Another experiment forced runners to run at stride frequencies higher and lower than normal for a given running speed, thus artificially restricting hip angle, again observing that runners adjust leg stiffness [13].

Most biomechanists claim that animals prefer to change speed by changing stride length (perhaps by controlling hip angle at touchdown), while leg stiffness remains constant [3, 6, 15]. Others claim that step length changes little with speed for normal running on a hard surface, implying stiffness control of the leg [16]. The discrepancy may arise from testing at different running speeds; most experiments involve running at low to moderate speeds, where runners seem to prefer a stride length change over a leg stiffness change. Only in experiments with high-speed running do subjects maintain a relatively consistent stride length over a range of speeds. Common wisdom in the athletic running community is that faster running is achieved by increasing stride length, increasing stride frequency (likely achieved by increasing leg stiffness), or both [17].

2.3 Using Leg Stiffness for Disturbance Rejection

Discussion has thus far focused on control of a running gait on flat, hard floors, but leg stiffness control is also useful when running over terrain with varying stiffness. Without leg stiffness control, the center of mass displacement and ground contact time will change, affecting the gait significantly. When hopping in place, humans adjust leg stiffness to maintain constant global stiffness in response to changes in surface stiffness at a given hopping frequency [3, 18]. During unrestricted running over varying terrain, human runners compensate for ground surface changes by varying leg stiffness, and maintain a relatively consistent center of mass motion [19–21].

Maintaining a constant center of mass motion is beneficial, despite the fact that changing leg stiffness costs energy. Hopping higher than necessary requires more energy storage in the leg, leading to higher spring restitution losses. Hopping high also increases flight time, allowing trajectory errors to build. Alternatively, hopping too low increases the risk of hitting obstacles and requires faster leg swing recoveries, which may require high power at the hip. Although these reasons are largely speculative, animals do control leg stiffness

to maintain a consistent center of mass motion, indicating some benefits. Most current running robots have manually tuned leg stiffness, and are not capable of varying leg stiffness on the fly. However, these robots are not subject to disturbances in ground stiffness; therefore, the reduced efficiency caused by center of mass motion changes is not important. Many values of leg stiffness will result in successful (though not optimal) running, so a robot with no leg stiffness adjustment can still control its forward speed, hopping height, or stride length [19].

3 Implementation on a Robot

In this section, we discuss implementation strategies for robots. Discussion of the merits of specific robot behaviors, such as spring-like stiffness, does not explicity address the implementation of the behavior. An independent argument must be made for design choices which can affect the performance, efficiency, and ultimately the ability of a machine to exhibit the desired behavior. The implementation of variable stiffness in animals is still debated; co-contraction of antagonistic muscles results in a stiffness change, but neural control of muscles can also result in behavior similar to a spring stiffness.

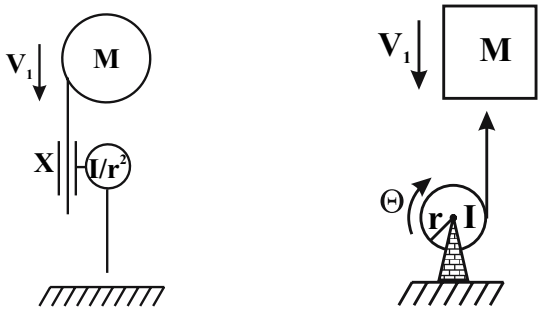
While the analysis in this paper is generalized and applies to a range of actuation technologies, it presumes the use of electric motors and springs. We also assume that energy efficiency is an important goal.

3.1 Electric Gearmotors and Inelastic Collisions

A simple design for a legged robot would utilize an electric gearmotor at each joint. Several groups have built bipedal robots using this design, and some intended to make the robots run as well as walk. The problem with this approach to running is that most of the kinetic and potential gait energy is lost with each hop to an inelastic collision with the ground.

A gearmotor-actuated running robot is represented in Fig. 2(a) as a spring-free, vertically-hopping representation of the SLIP model. The absence of a spring is an important distinction from a spring of infinite stiffness, because it represents the difference between an elastic collision and an inelastic collision. For the model to portray realistic SLIP model behavior, the foot should stick to the ground on impact without chattering, implying an inelastic collision. Spring behavior should be implemented behind the foot, either by the gearmotor or by a physical spring, so there is some stance time during which the robot can apply control forces. Another important distinction is that the reflected rotor inertia, represented by $\frac{I}{r^2}$, does not add to the overall mass of the system. The entire mass is represented by M, while $\frac{I}{r^2}$ may be any arbitary quantity, perhaps larger than M.

Because the rotor inertia and the robot mass are unrelated, the pin joint of the reflected rotor inertia may be moved to the ground without affecting the



(a) SLIP Model, without a spring

(b) The inertias separated - still the same collision model

Fig. 2. Figures representing the Spring Loaded Inverted Pendulum model, with the physical spring removed. The inertia of the motor is represented by I, the mass of the robot by M, the gear ratio on the motor by r, the velocity of the robot just before collision by V_1, and the position of the motor by X, or Θr

model, as shown in Fig. 2(b). The rotor begins at rest, and after collision, has some speed that matches that of the mass. If the kinetic energy just prior to impact is represented as T_0, the rotor inertia is represented as I, the effective gear ratio is r, and M is the robot's total mass (including motor mass), then the energy lost to an impact is:

$$T_{loss} = \frac{I}{Mr^2 + I} T_0$$

and the remaining energy, stored in the downward motion of the robot and the rotation of the motor, is:

$$T_{final} = \frac{Mr^2}{Mr^2 + I} T_0 \ .$$

If the effective inertia of the motor rotor ($\frac{I}{r^2}$) is the same as the robot's mass, then half the kinetic energy from flight will be lost to an inelastic collision with the ground. Even after the collision, any remaining energy must be converted through the motor and transmission inefficiencies, which are compounded when energy must pass into the system and then out. In effect, very little energy can be recovered. Most will be lost instantly during collision, much of the rest to motor inefficiencies.

3.2 Improving on the Basic Gearmotor

Minimizing rotor inertia or adding a spring in series between the motor and the leg will reduce the energy lost to inelastic collisions. Both methods are

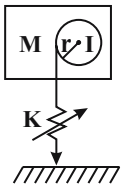
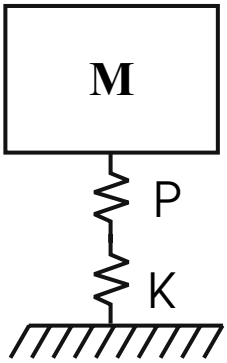
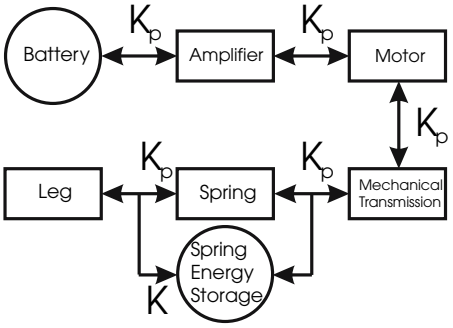


Fig. 3. A SLIP, identical to that shown in Fig. 1, but hopping vertically



(a) SLIP model, with software spring depicted and labelled P, hardware spring labelled K



(b) Energy flow diagram; K_p represents the energy path of stiffness behavior implemented through software control, K represents the mechanical spring stiffness.

Fig. 4.

used in force-control applications [22–25], which are similar in many ways to implementation of a spring rate. A series spring system such as that depicted in Fig. 3 has the added benefit of disconnecting the rotor inertia from the leg inertia, such that energy is lost to inelastic collision with only the mass of the leg. Our analysis shows that for a series spring system being used to implement a SLIP, performance and efficiency are maximized when the physical spring stiffness matches the desired spring stiffness.

For a series-elastic actuator with a physical spring rate that does not match the desired total spring rate, the motor must apply torques so the overall system exhibits the desired total spring rate. Given a zero-inertia rotor (the best-case scenario), a proportional controller will behave like a spring, creating two springs in series – a software spring and a physical spring, as shown in Fig. 4(a). This is relatively simple to analyze, and provides a conservative estimate of energy use and power output due to the assumption of no inertia. Therefore, further analysis will assume perfect force control of a massless rotor, providing an ideal software spring in series with a physical spring.

Power Density of a Series Spring System

In a cyclical system such as a hopping SLIP, energy is transferred from external sources (kinetic energy of motion, potential energy of height) to internal sources (physical spring energy or chemical battery energy) and vise-versa, repeatedly. This transfer of energy is represented in Fig. 4(b), where the energy may go into and out of the physical spring as an energy storage element (compression and extension), or through the physical spring as merely a power-transmission element (the spring translating, with no deflection). The power output will be divided between the software spring and the physical spring, depending on their stiffnesses.

If the series spring system is deflecting at some rate, the power output attributed to the software spring, P_{K_p}, is

$$P_{K_p} = \frac{K}{K_p + K} P(t) \ ,$$

where $P(t)$ is the total power output of both springs in series, K_p is the software proportional gain, and K is the physical spring constant. If the physical spring is perfectly tuned to match the desired stiffness, P becomes infinite, and it can be seen from the equation that the motor (exhibiting the software spring) exerts zero shaft power.

Because springs have higher power density than electric motors, it makes sense to design a system such that the physical spring transfers as high a proportion of the power as possible. A physical spring can have nearly infinite power density, depending on its stiffness; therefore, a comparison between the power density of a spring and that of a motor must be made in the context of an application. Choosing reasonable values for a hopping robot of leg stiffness $K = 5000\,\mathrm{N/m}$, hopping height of $h = 0.25\,\mathrm{m}$, and robot mass of $m = 30\,\mathrm{kg}$, the highest power output during stance is approximately $1\,\mathrm{kW}$ (RMS power is $680\,\mathrm{J}$) and the maximum work stored is about $75\,\mathrm{J}$. With an efficient fiberglass spring, such as those used on archery bows which have energy capacity around $1000\,\mathrm{J/kg}$, a $75\,\mathrm{g}$ spring can store the required energy and output the desired power. In contrast, a brushless, frameless motor that can output $600\,\mathrm{W}$ of continuous power (found through internet search) weighs approxmately $2.2\,\mathrm{kg}$, almost 30 times the mass of the spring. Adding the necessary electronics, batteries, and motor housing would add to the mass considerably. It may be that current electric motor technology cannot supply sufficient power density for a running robot. One study on the gearmotor-driven Honda humanoid robot found that the motors would have to be 27–54 times as powerful, without increasing weight, to make the robot run [26].

Energy Efficiency of a Series Spring System

Although the power requirements are the most compelling reasons to use physical springs, energetic efficiency is also improved through the use of tuned

physical springs. Given previous assumptions of a perfect software spring and an inertia-free rotor, spring efficiency e_k and motor efficiency e_p, the equation for energy returned is

$$E_{ret} = \frac{P}{2(K+P)} K x^2 e_k + \frac{K}{2(K+P)} P x^2 e_m \ .$$

If spring efficiency is higher than motor efficiency, it makes sense for the physical spring stiffness to be as close to the desired spring stiffness as possible. If our assumption of zero rotor inertia is false (and for any real system, it is), then the motor must transmit power to change the momentum of the rotor, and it will expend (and lose) more energy than in this idealized example.

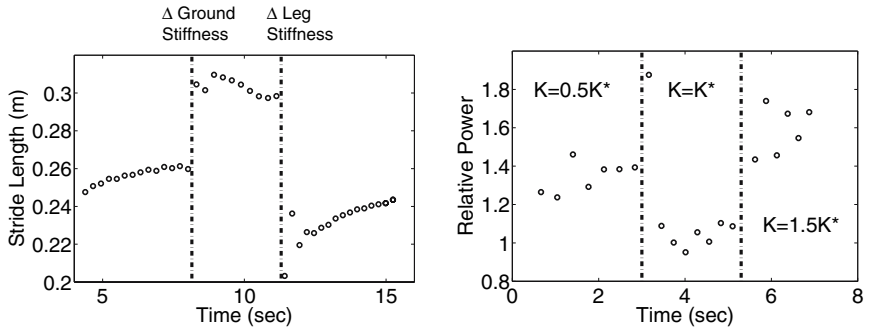
This effect of energy efficiency on a mis-tuned system has been demonstrated on humans in "Groucho Running." Human subjects were instructed to run with knees bent more than normal, thus consciously decreasing their leg stiffness. Oxygen consumption increased by as much as 50%, indicating a significant increase in energy use [27].

4 Simulation

Based on these ideas of maintaining a tuned physical spring, we designed and built an Actuator with Mechanically Adjustable Series Compliance (AMASC) [1, 2]. We also created a dynamic simulation of the AMASC, and verified the simulation through comparisons with benchtop tests of the prototype. We then used the simulated AMASC to actuate a simulated bipedal running robot, using Raibert-style controllers. Through testing of the simulation, we show that variable compliance can be used as a method of gait control, and also that a mis-tuned spring requires more shaft power than a properly tuned physical spring.

The first test involves a variable ground stiffness, and a controllable leg stiffness. After the robot has reached a steady-state running gait, the ground stiffness is changed. In Fig. 5(a), it can be seen that the stride length is affected. After several hops with the altered gait, the leg stiffness is changed, bringing the robot closer to its original stride length. This test is not optimized, and the leg stiffness adjustment is only approximate; but it does show that in this simulated system, which is based on the SLIP model and simulates a physical prototype, the leg stiffness can be used as a method of gait control.

The second test of the simulation involves standard running, with a constant leg stiffness K^*, but a changing mechanical spring stiffness K. While running with a mis-matched mechanical spring, the motor must exert additional power to create the desired knee behavior. When the mechanical spring matches the desired stiffness, power output by the motor is minimized. Assuming some motor inefficiencies, a graph of energy expenditure would follow the graph of the power output.



(a) Stride length vs. time, with changing commanded stiffness K*

(b) relative power vs. time, with changing physical stiffness K. Commanded stiffness, K*, remained the same.

Fig. 5.

5 Conclusion

Observations of animal behavior provide insights to possible control for robotic legged locomotion. One observation is that animals vary leg stiffness to control their running gait within the constraints of the SLIP model. Based on arguments put forth in this paper, we believe that robots would also benefit from this strategy. Control of leg stiffness can be used to vary the gait and influence the forward speed, hopping height, or stride length; it can also be used to maintain a consistent CoM motion in the presence of outside disturbances, such as a ground stiffness change. A properly tuned leg stiffness produces passive stability properties, and likely minimizes energy use. Most importantly, the leg stiffness behavior should be implemented primarily through mechanical springs, rather than through software control of the motors. It is far more energetically efficient, requires significantly lower power from the motors, and will result in lower shock loads to the entire robot.

The final goal is to be able to design and build robots capable of efficient and disturbance-tolerant running on variable terrain, mimicking much of the efficiency, agility, and robustness of animals. To achieve these goals, leg actuators of running robots should have mechanically adjustable series compliance, and the software control algorithms should be designed to appropriately tune the leg compliance in the full variety of running situations.

Acknowledgements

This work is supported in part by an NSF fellowship held by the first author and by The Robotics Institute of Carnegie Mellon University. Many thanks to Joel Chestnutt, for ongoing discussions and for his contribution to the software simulation tool.

References

1. Jonathan W. Hurst, Joel E. Chestnutt, and Alfred A. Rizzi. An actuator with physically adjustable compliance for highly dynamic legged locomotion. In *IEEE International Conference on Robotics and Automation,* 2004. (to appear).
2. Jonathan W. Hurst, Joel E. Chestnutt, and Alfred A. Rizzi. An actuator with mechanically adjustable series compliance. Technical report, Carnegie Mellon Robotics Institute, 2004.
3. Robert J. Full and Claire T. Farley. Musculoskeletal dynamics in rhythmic systems – a comparative approach to legged locomotion. In J. M. Winters and P. E. Crago, editors, *Biomechanics and Neural Control of Posture and Movement.* Springer-Verlag, New York, 2000.
4. R. Blickhan and R. J. Full. Similarity in multilegged locomotion: Bouncing like a monopode. *Journal of Comparative Physiology*, pp. 509–517, 1993.
5. Michael H. Dickinson, Claire T. Farley, Robert J. Full, M. A. R. Koehl, Rodger Kram, and Steven Lehamn. How animals move: An integrative view. *Science,* 288:100–106, April 2000.
6. Marc Raibert. *Legged Robots That Balance.* MIT Press, Cambridge, Mass., 1986.
7. M. Ahmadi and M. Buehler. The ARL monopod II running robot: Control and energetics. In *IEEE International Conference on Robotics and Automation,* pp. 1689–1694, May 1999.
8. Garth Zeglin and H. Benjamin Brown. Control of a bow leg hopping robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May, 1998.
9. Andrew Seyfarth, Harmut Geyer, Michael Gunther, and Reinhard Blickhan. A movement criteria for running. *Journal of Biomechanics,* 35:649–655, November 2001.
10. Robert J. Full, Timothy Kubow, John Schmitt, Philip Holmes, and Daniel Koditschek. Quantifying dynamic stability and maneuverability in legged locomotion. *Integrative and Comparative Biology,* 42:149-157, 2002.
11. William J. Schwind and Daniel E. Koditschek. Characterization of monopod equilibrium gaits. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation,* 1986–1992, Albequerque, New Mexico, April 1997.
12. Jessica K. Hodgins and Marc H. Raibert. Adjusting step length for rough terrain. In *IEEE Transactions on Robotics and Automation,* volume 7, 1991.
13. Claire T. Farley and Octavio Gonzalez. Leg stiffness and stride frequency in human running. *Journal of Biomechanics,* 29(2):181–186, 1995.
14. Thomas A. McMahon and George C. Cheng. The mechanics of running: How does stiffness couple with speed? *Journal of Biomechanics,* 23:65–78, 1990.
15. Claire T. Farley, James Glasheen, and Thomas A. McMahon. Running springs: Speed and animal size. *Journal of Experimental Biology,* pp. 71–86, 1993.
16. T. A. McMahon. The role of compliance in mammalian running gaits. *Journal of Experimental Biology,* 115:263–282, 1985.
17. Florence Griffith Joyner, John Hanc, and Jackie Joyner-Kersee. *Running For Dummies.* For Dummies, 1999.
18. Daniel P. Ferris and Claire T. Farley. Interaction of leg stiffness and surface stiffness during human hopping. *The American Physiological Society,* pp. 15–22, 1997.

19. Daniel P. Ferris, Kailine Liang, and Claire T. Farley. Runners adjust leg stiffness for their first step on a new running surface. *Journal of Biomechanics,* pp. 787–974, 1999.
20. Claire T. Farley, Han H. P. Houdijk, Ciska Van Strien, and Micky Louie. Mechanism of leg stiffness adjustment for hopping on surfaces of different stiffnesses. *The American Physiological Society,* pp. 1044–1055, 1998.
21. Daniel P. Ferris, Mickey Louie, and Claire T. Farley. Running in the real world: adjusting leg stiffness for different surfaces. *Proceedings of the Royal Society London,* 1998.
22. W. T. Townsend and J. K. Salisbury. Mechanical bandwidth as a guideline to high-performenace manipulator design. In *IEEE International Conference on Robotics and Automation,* volume 3, pp. 1390–1395, 1989.
23. Takeo Kanade and Donald Schmitz. Development of CMU direct-drive arm II. Technical Report CMU-RI-TR-85-05, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 1985.
24. J. De Schutter. A study of active compliant motion control methods for rigid manipulators based on a generic control scheme. In *Proceedings of the IEEE International Conference on Robotics and Automation,* pp. 1060–1065, 1987.
25. Gill A. Pratt and Matthew M. Williamson. Series elastic actuators. In *IEEE International Conference on Intelligent Robots and Systems,* volume 1, pp. 399–406, 1995.
26. Shuuji Kajita, Takashi Nagasaki, Kazuhito Yokoi, Kenji Kaneko, and Kazuo Taie. Running pattern generation for a humanoid robot. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation,* pp. 2755–2761, May 2002.
27. Thomas A. McMahon, Gordon Valiant, and Edward C. Frederick. Groucho running. *Journal of Applied Physiology,* 62:2326–2337, 1987.

Implementation of a Driver Level with Odometry for the LAURON III Hexapod Robot

J.L. Albarral and E. Celaya

Institut de Robòtica i Informàtica Industrial (IRI) UPC-CSIC
Llorens i Artigas 4-6, 08028 Barcelona (Spain)
albarral@iri.upc.es celaya@iri.upc.es

Abstract. A vision-based robot navigation system was designed as a general platform able to be used with wheeled and legged robots. We adapted our former hexapod robot Lauron III controller with a new driver level that provides all necessary behaviours, such as odometry, to interface it with the general navigation system, thus showing the independence of the navigation system from the specific kind of robot that actually performs the navigation task.

1 Introduction

In our present research project [2] the final goal is to develop a general purpose visual-based navigation system for navigation of autonomous mobile robots in unknown outdoor environments. The purpose of such a robotic navigation system is to provide a simple graphical interface to a non specialized user who can select a visual target in the scene currently captured by the robot's cameras, starting a navigation process to reach the target with no further user intervention. This general interface should be used for any type of mobile robot, no matter if it is a wheeled or a legged one.

In this work we first explain the navigation system and then the way of integrating our Lauron robot controller with this system. Next we present the specific behaviours implemented in Lauron controller to cope with the requirements of the navigation system: speed control, rotation control, block detection, and odometry.

2 Navigation System

In the central part of the system there are the Navigator module and the GUI module. In the GUI module the user selects a visual target for the robot to reach. Then the Navigator module creates a local map with all the landmarks

detected by the vision system and uses this map to order movement commands to the robot in order to drive it towards the selected target.

Another part of the system involves the Pilot and RobotDriver modules which remotely control the movement of the robot. The Pilot module, which receives commands from the Navigator module, implements the obstacle avoidance behaviour and sends the appropriate commands to the RobotDriver module, which finally controls the speed and direction of the robot movement through a wireless connection.

The last part of the system is the vision system which involves the Tracking System module and the Pan and Tilt Unit (PTU) module for the remote control of the robot's pan and tilt, and the FrameGrabber module for the remote image acquisition from the robot's camera. Finally various parallel Landmark Detector modules extract visual landmarks from the camera images and pass them to the Navigator and GUI modules.

The described system can be used for autonomous navigation of different types of robot, pan and tilt's, cameras and frame grabbers. Therefore the corresponding modules (RobotDriver, PTU and FrameGrabber) have to be specifically implemented although keeping compatibility with the rest of modules of the system.

The RobotDriver module outputs the following driving commands to the robot:

- Translational speed.
- Rotational speed.
- Rotation angle (translation remains halted until the ordered rotation is finished)

And also asks the robot for the following information:

- (X,Y) coordinates in a given global coordinate system (no Z coordinate is asked by present version, intended for flat ground)
- Robot heading angle in the same coordinate system
- Block status (active when an obstacle is blocking the robot's trajectory)

The PTU module outputs the absolute angle commands for the pan and tilt unit. And the FrameGrabber module periodically gets the images captured by the robot's camera to be processed by the system's Landmark Detector modules.

3 Implementation for Lauron III

In our case, we want to use the navigation system with the Lauron III hexapod robot and evaluate its performance in different navigation tasks.

As explained in [3] we had implemented a hierarchical walk controller for Lauron III based on Brooks' Subsumption Architecture [1]. In this controller

we divided the walking task in five levels: Balance, Adaptation, Force compliance, Walk, and Drive. With the implementation of the four lower levels of the control structure the robot was able to reliably walk on irregular terrain robustly adapting to ground conditions. User selectable parameters were available for the step length and height in case of straight trajectories, and also for the turn radius and step angle in case of circular trajectories.

The robot's top Drive level has now been implemented according to the navigation system requirements. Behaviours such as odometry, block detection, speed control and rotation control have been implemented.

3.1 Integration of the Lauron Controller with the Navigation System

The lower levels of control of the Lauron robot are implemented in MCA2 [4], a hierarchical, modular architecture developed at FZI (Karlsruhe). A possible choice for the integration of the existing control processes with the general navigation system would be the extension of the MCA2 controller with a top level module implementing the navigation system. However, this would make the system dependent of the specific control architecture used in this robot, and the portability of the navigator would be impaired. A better alternative is to maintain the navigation system and the MCA2 controller as independent processes that run in parallel and communicate with each other, thus maintaining a clean separation between them. Next we give a short explanation of the MCA2 architecture and explain how the integration with the navigation system has been done.

MCA2 is a modular C++ framework for building robot controllers where all the methods are realized by simple modules with standardized interfaces. These modules are connected via data transporting edges which is how the communication between the single parts of the entire controller architecture is managed. Every module is structured in the same way. In every direction (above and below) there is an interface to receive data (input) and to send data (output). Every module consists of two functions: Sense and Controle. These functions contain the actual functionality of a module. For clarity reasons modules can be arranged into module groups. Finally there is a global group that includes all other groups and modules, and which is managed by the executable program called Part.

The behaviour of Lauron's MCA2 program can be externally controlled through the control input interface of the global group, and its state can be externally checked through its sensor output interface. Luckily, the MCA2 framework provides a TCPConnection class which permits to access those interfaces of the global group from an external program running in a remote machine. Thus the integration is done in the following way: The user interface of the navigation system has a "Connect" pushbutton that, when activated, creates a TCPConnection object that establishes a connection from the user's machine to Lauron's IP address, and will be used by the navigation system

to send its motion, PTU, and FrameGrabber commands to Lauron to write the appropriate values (translational speed, rotational speed, turn angle, pan angle, tilt angle,...) in the corresponding control inputs of the MCA2 program. The TCPConnection object will also be used to read the necessary robot values (X and Y coordinates, heading angle, block status...) from the corresponding sensor outputs of the MCA2 program.

Image Management

Image processing is done off-board the robot for two reasons. First because it is a computationally intensive task that may exceed the resources of the on-board processor, and second, to make easier the adaptation of landmark detection algorithms already developed for other robot platforms used with the navigation system.

Images are transmitted to the external computer in the following way: Image frames captured by the camera are stored in a MCA2 blackboard defined in the robot's memory. The navigation system uses the established TCP-connection explained above to obtain information about the parameters of the Lauron's MCA2 blackboard. This information is used by the navigation system interface to create a MCABlackboard object (provided by the MCA2 framework) which creates a local blackboard and establishes a new connection directly to the remote Lauron's blackboard. Through this new connection, the FrameGrabber module of the navigation system will periodically update the local blackboard with the data from the Lauron's blackboard, thus allowing the navigator system to process the images obtained by the Lauron's camera. These images are also used for display in the interface to show the user what Lauron is currently seeing.

3.2 Speed Control

As explained before, the navigation system is defined to order translational and rotational speed commands. However, Lauron walk controller receives orders of step length for straight trajectories, or turn radius and step angle for circular trajectories. The bigger these step lengths/angles were, the higher translational/rotational speeds of the robot were obtained.

However, accurate speeds can not be granted by Lauron in unstructured terrain because local ground conditions and leg collisions with obstacles greatly affect step times.

Anyway, speed measurements were made in flat ground conditions for different step lengths/angles in order to use them in a new Speed Control behaviour which translates from the navigation system ordered speeds to Lauron step lengths and step angles.

3.3 Rotation Control

Rotation angle commands provided by the navigation system expect the robot to halt translation and to rotate in place the whole given angle before the translation continues. Again this command may be common for wheeled robots where rotation can be suddenly stopped at any given time providing accurate rotation movements. But in the case of legged robots a rotation accuracy problem arises because rotation can not be stopped while any leg is performing a step process.

In our case, the new available information of robot heading angle provided by the implemented Odometry behaviour has proven to be very useful. The new Rotation Control behaviour uses this information to start and stop the rotation movement while halting the robot translation. And the rotation accuracy problem has been solved by estimating the remaining rotation angle before each new step in order to perform an appropriate final step to exactly match the ordered rotation angle.

3.4 Block Detection

Wheeled robots usually have bumpers, infrared, or other kinds of short range sensors all around their body to prevent or detect collisions with obstacles. The detection of an obstacle in the path of a wheeled robot always implies a block situation since this kind of robots is not expected to pass over obstacles large enough to be detected by its sensors.

In the case of legged robots the situation is very different, since not all obstacles correspond to a blocking situation. Because of the flexibility given by the legged structure, most middle-sized obstacles can be overcome without the need to turn away from them. A legged robot may collide with obstacles with its legs and/or body, and both situations can be overcome by issuing higher steps or by raising the robot's body while passing those obstacles, respectively. Only big enough obstacles which the robot is not able to overcome will block its way.

In the Lauron walk controller, both obstacle overcoming behaviours are implemented: the Rebound behaviour detects leg collisions during the transfer phase of each leg by monitoring the force and current sensors of the implied joints, and reacts by reissuing the step at a higher position. A block situation is detected when, after a number of attempts, the leg has reached its highest permitted position and has not been able to avoid the collision.

Similarly, the Body Raise behaviour detects obstacles in the front of the body with an infrared sensor and adjusts the body height to keep a convenient distance with the obstacle. A block situation is detected when the body reached its highest permitted elevation but the obstacle is still detected.

3.5 Odometry

Odometry is a common feature in wheeled robots where the advanced distances or rotated angles are easily measured through the number of turns done by each wheel. But the problem arises in the case of legged robots where odometry is far more difficult to measure because many legs and many more motors are involved in the robot displacement, so the combination of their contributions is hard to deal with.

In the case of our Lauron hexapod robot we could solve this problem by taking profit of the fact that all body movements are directly done in the lowest Balance level by the combination of six independent behaviours which continuously adjust the position of the body with respect to all supporting legs. Three of these behaviours are exclusively involved in body translation along the robot X, Y and Z axes, while the other three are exclusively involved in body rotation around those same axes.

As the present navigation system version is only intended for flat ground environments, we have made an odometry approximation for only two degrees of freedom, instead of for the six degrees of freedom provided by Lauron's body movement. This approximation allows the new Odometry behaviour to only keep track of the contributions made by two of those Balance behaviours: the one involved in the body rotation around the Z axis for the measure of robot heading angle, and the one involved in the body translation along the robot X axis (advance direction) for the measure of global (X, Y) position.

Robot Heading Angle

The Z Rotation Balance behaviour is the one involved in the body rotation around the Z axis. This behaviour adjusts the position of the body by issuing appropriate simultaneous displacements to all supporting legs. These displacements have different lengths and directions for each supporting leg. The directions are perpendicular to the foot vectors (with origin at the center of the robot's body) and the lengths are the exact ones that make all supporting legs rotate the same angle around the center of the body, thus leading the body to rotate the opposite angle with respect to the supporting legs.

The odometry behaviour is continuously monitoring the displacements ordered by the Z Rotation Balance behaviour and calculating its corresponding rotation angle, which are accumulated from the beginning of Lauron task to keep track of the robot heading angle at each moment.

The calculation of the rotation angle γ is done as

$$\gamma = \text{argtan}(\|\Delta\nu\|/\|\nu\|)$$

where ν is the vector (x, y) of the previous leg position and $\Delta\nu$ is the displacement vector $(\Delta x, \Delta y)$ ordered by the Z Rotation Balance behaviour to this leg.

Robot (x, y) Position

The X Translation Balance behaviour is the one involved in the body translation along the X axis, which is the robot advance direction. This behaviour adjusts the position of the body by issuing the same simultaneous displacements to all supporting legs in the advance direction.

The odometry behaviour is also monitoring this behaviour displacement orders and uses them together with the calculated robot heading angle to calculate the (X, Y) position of the robot in a global coordinate system.

4 Conclusions

A vision-based robot navigation system was designed with the purpose of constituting a general platform able to be used with different robots, including wheeled and legged ones, provided they are endowed with a minimal set of sensorial and motor capabilities. We implemented the specific modules needed to interface the six-legged robot Lauron III with the system, thus showing the independence of the navigation system from the specific kind of robot that actually performs the navigation task.

Acknowledgements

This work has been partially supported by the Spanish Ministerio de Ciencia y Tecnología and FEDER funds, under the project DPI2003-05193-C02-01 of the Plan Nacional de I+D+I.

References

1. Brooks R.A. (1986) A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, vol. RA-2, No. 1, March 1986, pp. 14–23.
2. Celaya E., Torras C. (2002) Visual Navigation Outdoors: the ARGOS Project. Proc. of the 7th. International Conference on Intelligent Autonomous Systems (IAS-7), Marina del Rey, USA, March 2002, pp. 63–67.
3. Celaya E., Albarral J.L. (2003) Implementation of a hierarchical walk controller for the Lauron III hexapod robot. CLAWAR 2003, 6th. International Conference on Climbing and Walking Robots, Catania, Italy, September 2003, pp. 409–416.
4. Scholl K.U., Kepplin V., Albiez J., Dillmann R. (2000) Developing robot prototypes with an expandable Modular Controller Architecture. In: Proc. of the 5th. International Conference on Intelligent Autonomous Systems (IAS-5), Venedig, pp. 67–74, June 2000.

Local Positive Velocity Feedback (LPVF): Generating Compliant Motions in a Multi-Joint Limb

Axel Schneider, Holk Cruse, and Josef Schmitz

Department of Biological Cybernetics, University of Bielefeld, P.O. Box 10 01 31, 33501 Bielefeld, Germany

Summary. The concept of local positive feedback, which is found in the movement control system of arthropods, can be used to generate meaningful movements in closed kinematic chains. We introduce two simple joint constructions each of which shows a passive compliance. These joints are the basic requirement for using local positive velocity feedback (LPVF) on the single joint level. The underlying control principle of LPVF is derived. As a final benchmark we simulate a planar manipulator, which is equipped with compliant joints each controlled by LPVF. This setup is able to turn a crank at different velocities.

1 A Short Introduction to Simple Biological Solutions

Studies on different arthropod species have revealed that the biological control strategies for generating locomotion appear to be computationally much simpler than traditional engineering solutions, while at the same time being more efficient. It is reasonable to assume, for example, that swing or stance trajectories for walking are not generated through accurate calculations of the kinematics by the animal. Especially the stance movement comprises high complexity since stance movements of all legs on ground have to be coordinated in order to move the body forward in a straight manner. Bässler [1] and Schmitz [7] found indications that positive feedback might be the solution for such tasks, at least for the stick insect *Carausius morosus*. They reported that positive feedback (also termed reflex reversal) occurs in single joints during the stance phase of walking. Cruse and co-workers [2, 4] showed in a simple kinematic simulation of the stick insect that positive feedback in the body-coxa ("hip") and femur-tibia ("knee") joint produces a realistic stance phase. Moreover positive feedback supports the coordination of joints, even of different legs, when walking over irregular surfaces via coupling through the environment [4].

2 LPVF and the Compliant Motion Problem

Whenever systems with limbs work in a closed kinematic chain, pure kinematic control is not sufficient to prevent the different joints from generating undesired tensions. Tensions can occur when a single limb is used for tasks like turning a crank or when a walking machine touches the ground with its feet. In these cases, a combination of position and force control has to be utilised. Such a combined approach is commonly included in the superordinate concept of *compliant motion*. Early examples for these *hybrid* control systems were given by Raibert and Craig [6]. A classical solution for a compliant motion problem includes calculations for direct and inverse forms of both kinematics and dynamics. Solving these for several degrees of freedom per leg and the number of extremities requires considerable computational power. As an alternative we propose to use LPVF on the single joint level.

3 The Compliant Joint

The key idea of the LPVF-concept is the exploitation of the physical properties of the system's interactions with its environment. These interactions have to change the system state. In an insect, a moving limb can be part of a closed kinematic chain, e.g. a leg in stance phase. The resulting passive elongation of its muscles due to their elastic properties is one part of the above mentioned change in the system state. The muscle elongations in turn result from the rotatory deflections of the joints. When these passive rotatory deflections are measured they provide additional information about the real state of the system namely in this case the overall angle of the joint. Passive compliance generated by some elasticity is a key feature of any joint that is controlled by LPVF. Following this insight we developed two kinds of joints with elastic properties. These are shown in Fig. 1. In Fig. 1a the single hinge joint with its two adjacent segments is depicted. The joint itself is formed by a ball bearing connection between the two segments.

In Fig. 1b a servo motor is mounted on the lower segment. The motor's shaft protrudes through the ball bearing. A bracket is attached to the motor shaft's head. This bracket is connected to a counter bracket on the upper limb via two spiral springs. If the upper segment is fixed, a rotation of the motor shaft results in a rotation of the lower segment. Yet in a fixed motor shaft position, the spiral springs allow for passive bending of the system by outer torques. Due to the antagonistic design, the spiral springs can be mounted pre-stretched and therefore are able to react with a distinct restoring force even on a small deflection.

The second design is shown in Fig. 1c. Instead of spiral springs a single flexible spring steel filament is fixed to the motor shaft's head. Its other end also meets a counter bracket on the upper segment for torque transmission.

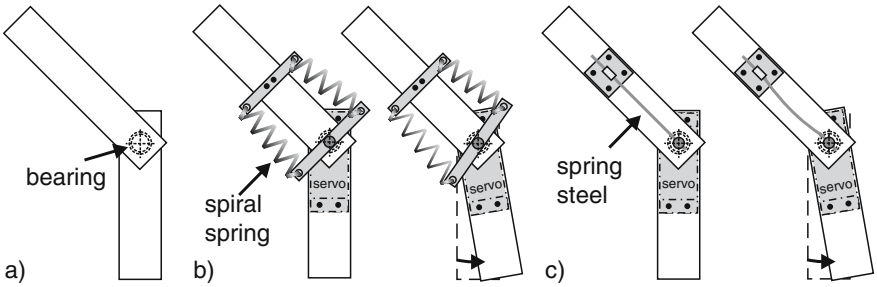


Fig. 1. Two different constructions for a compliant joint: (**a**) two segments connected with a ball bearing; (**b**) construction with two spiral springs; (**c**) construction with a spring steel filament

This version of a compliant joint is space saving and can therefore be used for joints very close to the body.

Both versions of the compliant mechanical joint approximate a biological joint with elastic properties as found in its driving muscles. They are equipped with a system of magnets and hall sensors which act as angle transducers. Before these constructions were used on a real robot they were tested in a simulation setup.

4 The Control Principle of LPVF

LPVF is based on the feedback of the joint velocity into the input of the driving servo motor. Figure 2 visualizes the detailed function principle. The servo motor of each joint acts as an integrator since it converts the angular velocity information at its input into angular positions α_s. The actual bending signal α_b is added on α_s to obtain information on the real joint angle α_j. This value is differentiated in order to convert the angular joint position into a joint velocity $\Delta\alpha = \dot{\alpha}$. The value is positively fed back into the integrator's input. This circuit shows an active compliant behaviour. A superimposed controller can be used to control the velocity of the joint. This was done for the crank experiment in Sect. 7.

5 Derivation of the dLPVF

The LPVF-controller depicted in Fig. 2 is analysed in more detail in this section. In order to derive a control algorithm, we describe the controller in the discrete time domain ($\mathcal{Z}$-domain) and term it dLPVF controller. Figure 3 shows two different interpretations of the controller idea developed so far. As it will be demonstrated in the next section, the controller in Fig. 3a bears an active position compliance behaviour, the one in Fig. 3b an active velocity compliance behaviour.

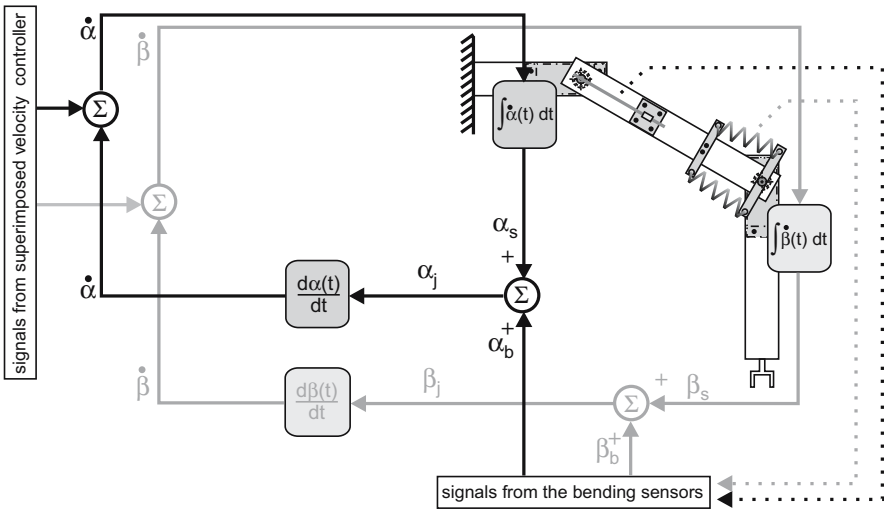


Fig. 2. LPVF-circuits for the joints of a planar manipulator. The *black solid lines* show the feedback circuit for the first joint α of a planar two segment manipulator. The *grey lines* describe the same circuit for the second joint β

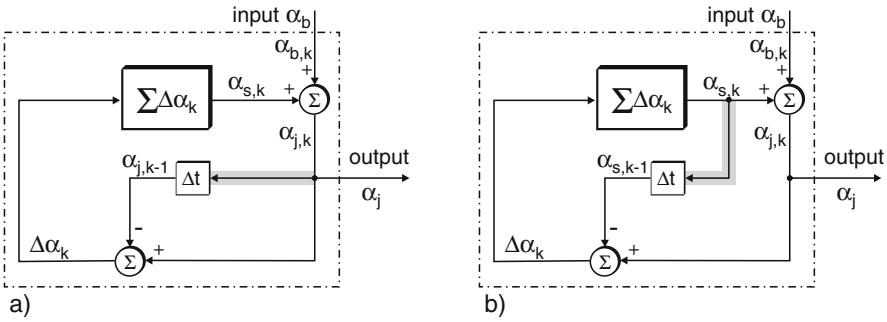


Fig. 3. The circuit diagrams of the discrete versions of LPVF (dLPVF): In (**a**) the most intuitive version of local positive velocity feedback is shown. This circuit generates an active position compliance of the associated joint. In (**b**) the second version of local positive velocity feedback is depicted. This circuit generates an active velocity compliance of the associated joint

5.1 dLPVF Controller with Active Position Compliance

Because of the decentralised control principle, the dLPVF controller in Fig. 3a can be modelled as a simple SISO-controller. The bending information α_b serves as its input and the joint angle α_j is the output. The system equations for the controller are shown in (1a) to (1c). Index k denotes the number of the time step $k\Delta t$ in the discrete time domain. Angles α_s, α_b and α_j are those introduced in Sect. 4.

$$\alpha_{s,k} = \alpha_{s,k-1} + \Delta\alpha_{k-1} \tag{1a}$$
$$\alpha_{j,k} = \alpha_{s,k} + \alpha_{b,k} \Leftrightarrow \alpha_{s,k} = \alpha_{j,k} - \alpha_{b,k} \tag{1b}$$
$$\Delta\alpha_k = \alpha_{j,k} - \alpha_{j,k-1} \tag{1c}$$

In order to get the discrete transfer function for the control circuit in Fig. 3a we insert (1c) in (1a) which yields:

$$\alpha_{s,k} = \alpha_{s,k-1} + \alpha_{j,k-1} - \alpha_{j,k-2} \tag{2}$$

By replacing $\alpha_{s,k}$ and $\alpha_{s,k-1}$ in (2) by the right hand side of (1b) we get:

$$\alpha_{j,k} = 2\alpha_{j,k-1} - \alpha_{j,k-2} + \alpha_{b,k} - \alpha_{b,k-1} \tag{3}$$

The $\mathcal{Z}$-transform of (3) is:

$$\begin{aligned} \alpha_j &= 2z^{-1}\alpha_j - z^{-2}\alpha_j + \alpha_b - z^{-1}\alpha_b \\ \Leftrightarrow \alpha_j &\left(1 - 2z^{-1} + z^{-2}\right) = \alpha_b \left(1 - z^{-1}\right) \end{aligned} \tag{4}$$

Using (4), the discrete transfer function reads as follows:

$$G(z) = \frac{\alpha_j(z)}{\alpha_b(z)} = \frac{z^2 - z}{z^2 - 2z + 1} = \frac{z^2 - z}{(z-1)^2} = \frac{z}{z-1} \tag{5}$$

For a closer inspection of the controller behaviour we analysed its pulse response. It is obtained by back transfomation of the transfer function given in (5) into the original (time) domain. The pulse response is shown in Fig. 4a. When the controller is excited at its input via a bending signal α_b, it only corrects for this value and relaxes the joint until there is no bending left. This can be described as an *active position compliance*.

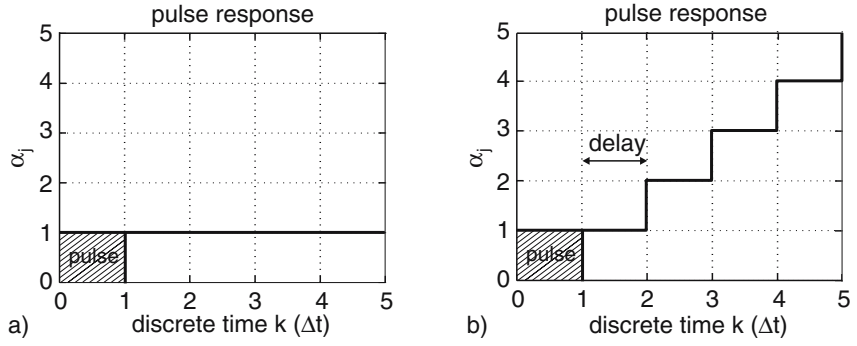


Fig. 4. The figure shows the pulse responses for the two different dLPVF circuits. In (**a**) the discrete pulse response of the active position compliance dLPVF controller is depicted. In (**b**) the discrete pulse response of the active velocity compliance dLPVF circuit can be seen. Note the delay of the response in (**b**)

5.2 dLPVF Controller with Active Velocity Compliance

The control circuit from Sect. 5.1 can be easily adapted to show an *active velocity compliance* behaviour. Velocity compliance in this context basically means that the dLPVF controller has to do two things at the same time. Firstly it has to relax the passive joint bending and secondly it has to pick up the velocity of the exciting signal and accelerate the joint to reach the same velocity. The underlying assumption is that the object which is responsible for the joint bending will move on with the same velocity in future. The circuit change is depicted in Fig. 3b. The system equations for this controller are set up in (6a) to (6c).

$$\alpha_{s,k} = \alpha_{s,k-1} + \Delta\alpha_{k-1} \tag{6a}$$

$$\alpha_{j,k} = \alpha_{s,k} + \alpha_{b,k} \Leftrightarrow \alpha_{s,k} = \alpha_{j,k} - \alpha_{b,k} \tag{6b}$$

$$\Delta\alpha_k = \alpha_{j,k} - \alpha_{s,k-1} \tag{6c}$$

In order to get the discrete transfer function for the controller we insert (6c) in (6a) which yields:

$$\alpha_{s,k} = \alpha_{s,k-1} + \alpha_{j,k-1} - \alpha_{s,k-2} \tag{7}$$

By replacing $\alpha_{s,k}$, $\alpha_{s,k-1}$ and $\alpha_{s,k-2}$ in (7) by the right hand side of (6b) we get:

$$\alpha_{j,k} = 2\alpha_{j,k-1} - \alpha_{j,k-2} + \alpha_{b,k-2} - \alpha_{b,k-1} + \alpha_{b,k} \tag{8}$$

The $\mathcal{Z}$-transform of (8) is:

$$\begin{aligned} &\alpha_j = 2z^{-1}\alpha_j - z^{-2}\alpha_j + z^{-2}\alpha_b - z^{-1}\alpha_b + \alpha_b \\ &\Leftrightarrow \alpha_j\left(1 - 2z^{-1} + z^{-2}\right) = \alpha_b\left(1 - z^{-1} + z^{-2}\right) \end{aligned} \tag{9}$$

Using (9), the discrete transfer function results in:

$$G(z) = \frac{\alpha_j(z)}{\alpha_b(z)} = \frac{z^2 - z + 1}{z^2 - 2z + 1} = \frac{z^2 - z + 1}{(z-1)^2} \tag{10}$$

The pulse response of the controller can be calculated by expanding the transfer function from (10) into partial fractions.

$$G(z) = \frac{z}{z-1} + \frac{1}{(z-1)^2} \tag{11}$$

The first summand of (11) again represents a step pulse. The second summand is a shifted straight line with unity slope. Since the system is linear the superposition of both leads to the pulse response shown in Fig. 4b. As a result the joint controller picks up the angular velocity of the excitation signal α_b and accelerates the joint to match this velocity in the following steps. For compensation of the delay displayed in Fig. 4b, the signal $\alpha_{b,k}$ was corrected adequately in order to get the results in the sections below.

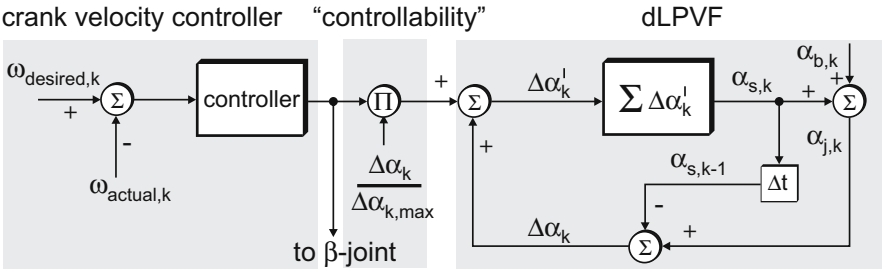


Fig. 5. The scheme of the control circuit (only α joint shown): On the left a negative feedback velocity controller is depicted. It receives the velocity error of the manipulator endpoint (e.g. the velocity of the crank in Sect. 7) as an input. On the right the LPVF circuit is shown. Since the velocity controller output is not suitable as a control information for a single joint, a mediating controllability measure (middle) scales the controller output

6 Velocity Control of a Multi-Joint Manipulator

In Fig. 5 the scheme of the control circuit is shown. The system consists of a negative feedback velocity controller depicted on the left hand side and the dLPVF controller derived in Sect. 5.2 depicted on the right hand side. The signal from the velocity controller cannot be fed into the LPVF-circuit directly, because it contains control information about the overall velocity of the manipulator during task execution. When using LPVF, the central velocity controller output can be regarded as a quality measure for the performance of the manipulator at an instant of time. Therefore the controller output is scaled by a joint velocity dependent *controllability* measure. The controllability value is zero when a joint reaches an inflection point during motion which means its velocity is zero. Thus the velocity controller has no influence on the joint movement in this phase. On the other hand the velocity controller has a large influence when the joint is moving with a considerable amount of its maximum possible speed.

7 Turning a Crank: Results from the Kinematic and Dynamic Simulation

For the evaluation of the control scheme we implemented a kinematic and a dynamic simulation of a planar manipulator with compliant joints, as described in Sect. 3, in Simulink 6 and SimMechanics 2.2 (The MathWorks Inc., 3 Apple Hill Drive, Natick, MA, USA). In both simulations the gripper of the planar manipulator (two segments: $l_1 = 197\,\text{mm}$, $l_2 = 174\,\text{mm}$) was connected to the handle of a crank (radius: $r = 80\,\text{mm}$; position: $x = 77\,\text{mm}$, $y = 197\,\text{mm}$) in order to form a closed kinematic chain. When the system was excited by a push, it started to turn the crank with the desired velocity. The

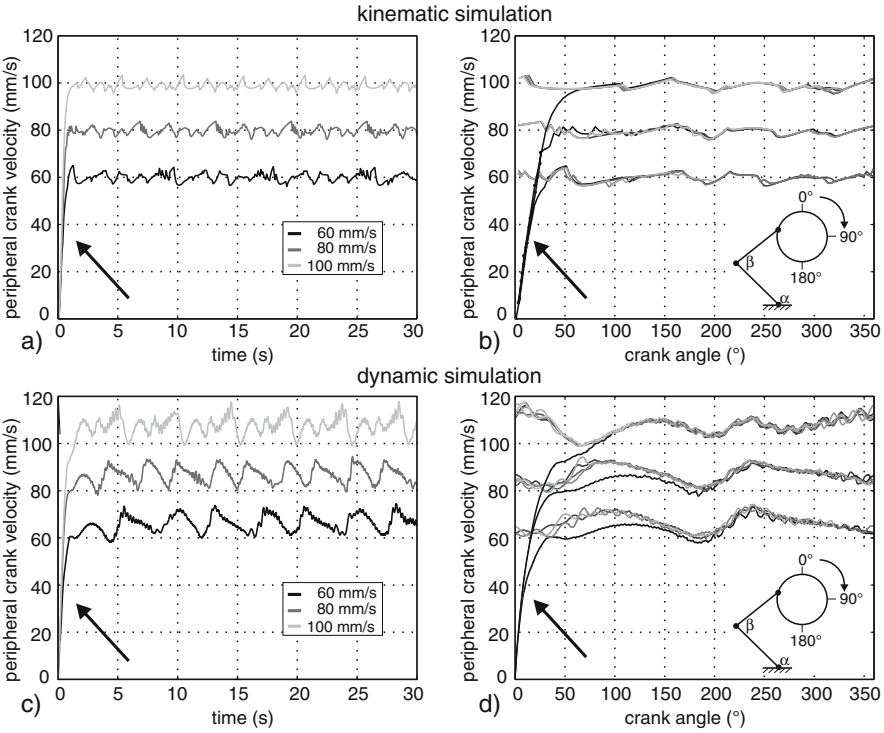


Fig. 6. Course of the crank velocity in the kinematic simulation (**a**) and (**b**) and in the dynamic simulation (**c**) and (**d**). The velocities and crank angles are given for clockwise rotations of the crank (see small sketches in (**b**) and (**d**)). In the left column the velocity of the crank is plotted over time whereas the right column shows the crank velocity plotted over the crank angle. Superimposed traces in shades of grey illustrate consecutive rotations of the crank. The crank was initially excited by an outer torque for 0.3 seconds (indicated by the black arrows). The crank velocity ranged from 60 mm/s up to 140 mm/s. Three velocities (60, 80, 100 mm/s) are depicted in the figure

results are shown in Fig. 6. In Fig. 6a and 6c the time course of the crank velocity is depicted. Figures 6b and 6d show the same data but this time plotted over the crank angle. Since the robot performed several rotations within the 30 seconds displayed in this figure, there are as many lines as rotations plotted for each velocity. Every new rotation is indicated by another shade of grey. Note the appearance of control errors which occur because the velocity controller was a pure p-controller and because the system reacts differently for every different posture during a crank turn. Instead of the p-controller a more sophisticated control scheme would further minimize the velocity errors.

8 Discussion and Conclusions

Compliant motion problems appear in different contexts as described in Sect. 2. Mason [5] developed the *Compliant Frame Formalism*, or *Task Frame Formalism* to specify such tasks in which (robotic) limbs act in closed kinematic chains. Most of the developed control approaches for compliant motions, as far as they are known to us, are either based on *hybrid control* [6] or on *impedance control* [3] and follow a central controller design rather than a decentral one. LPVF on a single joint level however is a decentral approach. It has been proven that this design solves the compliant motion task of turning a crank. Although the cranking process can only be handled by coordinated actions of the participating joints they can be controlled separately. The coordination is achieved by mechanical coupling through the environment. The same system can be used to generate useful stance trajectories in a multilegged robot during walking. Preliminary tests with a single leg on a treadmill were successful.

References

1. U. Bässler. Reversal of a reflex to a single motoneuron in the stick insect carausius morosus. *Biol. Cybernetics*, 24:47–49, 1976.
2. H. Cruse, C. Bartling, and T. Kindermann. High-pass filtered positive feedback for decentralized control of cooperation. In F. Moran, A. Moreno, J.J. Merelo, and P. Chacon, editors, *Advances in Artificial Life*, pp. 668–678, New York, 1995. Springer.
3. N. Hogan. Impedance control: An approach to manipulation: Part i – theory, part ii – implementation, part iii – applications. *ASME J. Dynam. Syst., Meas., Contr.*, 107:1–23, 1985.
4. T. Kindermann. Behavior and adaptability of a six-legged walking system with highly distributed control. *Adapt. Behav.*, 9(1):16–41, 2002.
5. M. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(6):418–432, June 1981.
6. M.H. Raibert and J.J. Craig. Hybrid position/force control of manipulators. *Transactions of the ASME*, 102:126–133, June 1981.
7. J. Schmitz, C. Bartling, D.E. Brunn, H. Cruse, J. Dean, T. Kindermann, M. Schumm, and H. Wagner. Adaptive properties of "hard wired" neuronal systems. *Verh. Dtsch. Zool. Ges.*, 88(2):165–179, 1995.

Motion Calculation for Human Lower Extremities Based on EMG-Signal-Processing and Simple Biomechanical Model

Christian Fleischer, Konstantin Kondak, Christian Reinicke, and Günter Hommel

Technische Universität Berlin
{fleischer,kondak,reinicke,hommel}@cs.tu-berlin.de

Summary. In this paper we present a way to calculate the motion of the human lower extremities online based on surface EMG signals emitted by the activated muscles. The EMG signal evaluation is directly integrated into the control loop of the model to allow for more flexible and spontaneous movements than with predefined trajectories. The algorithm implements a simple biomechanical model composed of bones and muscles. An additional stability controller modifies the torques in the ankle, knee and hip joints to keep the model in a stable pose. The proposed method will be part of a control system for an exoskeleton robot where the movement of the model will be interpreted as the intended movement of the operator. The performance of the presented method was investigated on the stand-to-sit movement.

1 Introduction

For many decades surface electromyography has been studied by many researchers in the medical and biomechanical fields to get a deeper understanding of muscles and how and when they are activated.

In recent years more and more studies have explored the relationship between single muscles and the complex movements of the human body, but most of these studies were focused on analysing disabilites, anomalies and how to track progress in rehabilitation. Only a few publications focused on using electromyographical signals in real-time to control biomechanical robots e.g. [1–3].

The main reason is the difficulty to map the EMG signal into the force a muscle is producing [4]. The approximated relationship itself is not too complex [5], but influenced by many different parameters. Some of these parameters only vary among different subjects, but some are even different from day to day. With the approach presented in [6] it is possible to calibrate the EMG-to-force relationship in a simple environment.

In this work we explain how calibrated muscle signals can be combined with motion controlling algorithms to achieve stable movements of a biomechanical model that reflects the intended movement of the operator. It is important to keep in mind, that this intention has to valid only for a short period of time in the future (fractions of a second). After that, the model can be synchronized with the reference system (synchronization is not done in this work to show the full effect of the model).

1.1 Why EMG?

Many different approaches for tracking the movement of a human being exist. Some use ultrasonic or visual sensors, some goniometers, accelerometers or other techniques. Every technique has its advantages and disadvantages, but all of the mentioned systems sense the current movement. With EMG signals it is possible to track the *intended* movement, which might differ from the current movement due to obstacles or lack of sufficient muscle-power. And if the EMG sensors are placed carefully, the intention should even be readable ahead of time. The reason for this is that EMG signals are detectable slightly before the movement is performed, because muscles take some time to produce the force after having received the activation signal.

2 Environment

In this section the environment is described in which the biomechanical model and the motion controller is embedded. Refer to Fig. 1 for an initial overview of the whole system.

As stated before, the basic idea is to let the *Human* interact with the *Mechanical System*. To achieve this, *EMG Signals* are collected from some

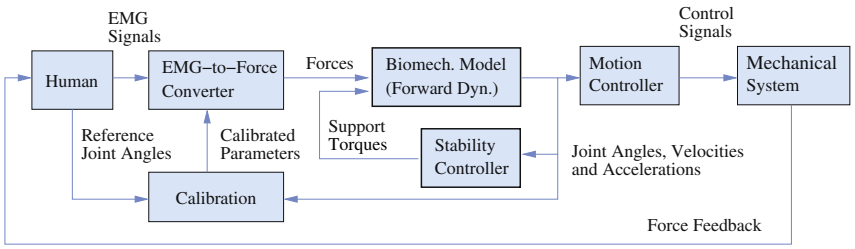


Fig. 1. Control scheme for a mechanical system attached to the human body: The EMG signals are captured from the subject, converted into forces and brought into the biomechanical model. Resulting joint angles and velocities from the model are passed to the motion controller to be executed by the mechanical system. The stabilty controller analyses the model state and computes supporting torques to keep a balanced pose. The calibration compares the joint reference angles with the angles computed by the model and modifies the EMG-to-force-parameters

of the muscles of the subject and converted to forces in the block *EMG-to-Force Converter*. The resulting *Forces* are fed into the *Biomechanical Model* that represents the human subject. Through forward dynamics computation accelerations for all body parts are calculated from the muscle forces and the current state of the model. After double-integrating those accelerations, the resulting joint angles, velocities and accelerations form the new state of the model and are interpreted as the desired movement of the human. The *Stability Controller* analyses the model state and calculates supporting torques for the ankle, knee and hip joint to bring the model back into a stable pose if necessary. The supporting torques are fed back to the biomechanical model to take effect.

The *Motion Controller* takes the desired movement and computes the control signals for the *Mechanical System*. Because of the connection between the human body and the mechanical system, the motion of the actuators of the mechanical system affect the human body (*Force Feedback*). The *Motion Controller* and the *Mechanical System* are not yet finished and not part of this work.

To be able to calibrate the parameters of the *EMG-to-Force Converter* the calculated *Kinematic Data* from the *Biomechanical Model* are compared with the *Reference Kinematic Data* taken from the human body with additional sensors (see Sect. 2.2). The computed parameters are then brought into the *EMG-to-Force Converter* again. Those additional sensors should be attached to the human with or without an exoskeleton to allow recalibration whenever necessary. The blocks *Biomechanical Model* and *Stability Controller* are the main focus of this work.

2.1 EMG Setup

The EMG signals are sampled with 1 KHz from DelSys 2.3 Differential Signal Conditioning Electrodes [7] with an inbuilt gain of $1000\frac{V}{V}$ and a bandpass filter from 20-450 Hz. The input data are rectified and then smoothed by a lowpass filter with a cutoff frequency of 5 Hz [8].

2.2 Reference System

The reference system is needed to capture the actual movement of the human limbs for the calibration step. We have used a reference system based on the two axis accelerometers ADXL210 from AnalogDevices Inc. [9].

As shown in Fig. 2 the orientation of the limb in the sagittal plane can be calculated by projecting the earth gravity field into the x- and y-axis of the sensor: $q_i = \arctan 2\,(G_{y\,i}/G_{x\,i})$. One sensor was placed on each of the torso, thigh and shank and as close as possible to the rotation axis of the joints to reduce the inertial acceleration resulting from limb movement (the error is small enough below $45\frac{deg}{s^2}$). Due to the nature of the sensors, there is only 0.5% temperature drift and a peak-to-peak noise below 2%.

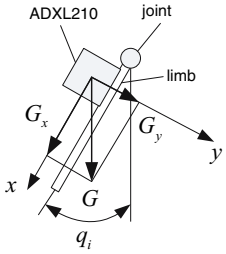


Fig. 2. Capturing the limb movement with two axes accelerometer ADXL210

2.3 Biomechanical Model

In literature a lot of information can be found about the anatomy of the human body: The properties of muscles, joints, tendons and tissues are available [10] and elaborated investigations have been done on biomechanical modelling of those [11]. Unfortunately, with increasing complexity of the model, more parameters have to be calibrated for every subject. Even though it is possible to create a complex model of the lower extremities [12], it is impossible to provide all those muscles in the model with properly recorded EMG values. This would result in an uncompensated defect of some muscles. So an equal layer of abstraction has to be applied everywhere.

For our experiments we have chosen a very simple model: It consists of the trunk, thigh, shank, foot and four muscles, as shown in Fig. 3 (both legs are modelled as one, since in our experimental setup they move as one). Each of those muscles in our model is representing a collection of muscles in the human body and does not necessarily have anatomical analogy. For the EMG signal acquisition we selected the muscles by their contribution to the most important movements of the leg: m. sartorius ($M0$), m. glutaeus maximus ($M1$), m. quadriceps femoris ($M2$), and m. semimembranosus ($M3$).

We are aware that this is a rough approximation, but since the goal is to provide input values for a control unit of a mechanical construction and

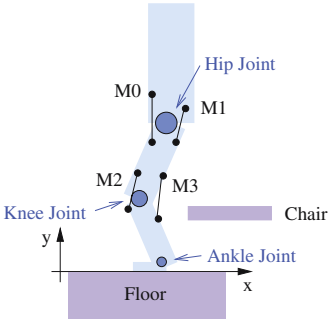


Fig. 3. The biomechanical model composed of trunk, thigh, shank, foot and four mucles: $M0$, $M1$, $M2$, $M3$

not pursueing clinical analysis, this seems to be an adequate simplification. If required, more muscles can easily be integrated into the model.

Each of those muscles is only represented by a contractile element. The passive elastic and viscous elements are summed up in the joints spanned by the muscle (in contrast to the Hill model) together with the friction of the joints. This simplification is similar to one presented in [13]. The overall frictions are assumed to be $5.0\frac{Nms}{rad}$ (hip joint), $1.0\frac{Nms}{rad}$ (knee joint) and $0.5\frac{Nms}{rad}$ (ankle joint). All values are inspired by [14] and were optimized by hand. The parameters of one muscle in our model are: the point of origin and insertion and the two parameters in the EMG-to-force function:

$$F_{EMG}(\phi_s, \phi_e) = \phi_s \left(1 - e^{-\phi_e x(t)}\right) \tag{1}$$

where $x(t)$ is the postprocessed EMG signal and ϕ_s, ϕ_e two muscle parameters. Since muscle origin and insertion are similar for the same average adult, offline optimization or calculation derived from the body dimensions can be used. Due to lack of space here, the model can be received from the authors on request. There are many other anatomical properties described in literature, but they are neglected for now.

Torso, thigh, shank and foot are modelled as cylindrical rigid bodies. The masses are calculated as fixed fractions of the total body weight ($m_{total} = 88\,\mathrm{kg}$) of the subject (figures can be found e.g. in [15]). The dynamic equations were derived using Kane's formalism [16], and have the following form:

$$\mathbf{M}(\mathbf{q})\,\dot{\mathbf{u}} = \mathbf{f}(\mathbf{q}, \mathbf{u}) + \mathbf{T}(\mathbf{q}, \mathbf{u}) \tag{2}$$

where

- $\mathbf{q} = (q_{ankle}, q_{knee}, q_{hip})^T$ is the vector of generalized coordinates (joint angles)
- $\mathbf{u} = (u_{ankle}, u_{knee}, u_{hip})^T$ is the vector of corresponding generalized velocities (with $\dot{\mathbf{q}} = \mathbf{u}$, the dot denotes the time derivative in a Newtonian reference frame)
- $\mathbf{M}(\mathbf{q})$ (matrix function) takes into account the mass distribution
- $\mathbf{f}(\mathbf{q}, \mathbf{u})$ (vector function) describes the influence of both inertial forces and gravity
- $\mathbf{T}$ is a vector of the generalized forces applied to the system. For the model considered, these are:
  - the forces produced by the muscles (F_{EMG}) multiplied with the nonlinear function $\mathbf{g}(\mathbf{q})$ (current system configuration and geometry of the muscles)
  - friction torques in joints (depending on $\mathbf{u}$)
  - supporting-torques vector $\mathbf{t_s} = (t_{ankle}, t_{knee}, t_{hip})^T$ that keeps the model in a stable pose (additional torques in the joints)
  - temporary contact force $\mathbf{f_c}$

For the experiments perfomed (see Sect. 4) an additional external force $\mathbf{f_c}$ is needed that must be applied to the hip to simulate the contact force with the chair when sitting down. A small P-controller simulates the effect of the chair contact force. If the y-coordinate of the hip y_{hip} moves below a certain value y_{chair} (height of the chair) the contact force $\mathbf{f_c}$ is applied to the hip to move the hip back on top of the chair and to the first point of contact with the chair in x (x_{cc}):

$$\mathbf{f_c} = \begin{cases} \text{with } ((x_{cc} - x_{hip}) * \alpha, (y_{chair} - y_{hip}) * \beta, 0)^T & \text{if } y_{hip} \leq y_{chair} \\ (0,0,0)^T & \text{otherwise} \end{cases}$$

where $y_{chair} = 0.45m$, $\alpha = 1.5 \times 10^5 \frac{N}{m}$, $\beta = 1.0 \times 10^5 \frac{N}{m}$ (experimentally determined). This controller might be substituted by an external sensor (either to get a boolean value for "contact"/"no contact" or the force value f).

An external floor contact force is not needed since the foot position is fixed and the reaction forces can be calculated implicitly through the model. The dynamic (2) were generated with the symbolic manipulation tool AUTOLEV [17]. The script for the system description and equatiopns generation can be received on request.

3 Stability Controller

The stability controller is activated whenever the recorded EMG signals are not sufficient to keep the model in a stable pose. f_{M1} (force of muscle group $M1$) is monitored and whenever $f_{M1} < 500N$ the stability controller is activated. In our experiments this is the case when standing upright. The controller is based on an approach presented in [18]. It takes the current state of the model $(\mathbf{q}, \mathbf{u}, \dot{\mathbf{u}})$ as and input and calculates the supporting-torques vector $\mathbf{t_s}$ that is applied to the joints. The stabilty condition is

$$x_{cm} = x_c \quad \text{with } x_c = \frac{x_{toe} + x_{heel}}{2} \tag{3}$$

Where x_{cm} is the x-coordinate of the center of mass of the whole body, x_{toe} the x-coordinate of the tip and x_{heel} of the heel of the foot. From the dynamic equations of the model x_{cm} can be rewritten as

$$x_{cm} = g(\mathbf{q}) \tag{4}$$

where g is a non-linear function in $\mathbf{q}$. Dual differentiation of (4) leads to

$$a_{cm} = \beta_0 + \beta_1 \dot{u}_{ankle} + \beta_2 \dot{u}_{knee} + \beta_3 \dot{u}_{hip} \tag{5}$$

where $\beta_0, \ldots, \beta_3$ are non-linear functions in $\mathbf{q}$ and $\mathbf{u}$. If $\dot{\mathbf{u}}$ is chosen in a way that (5) is satisfied the center of mass x_{cm} is moving towards x_c. This can be calculated by interpreting (5) as a plane in the acceleration space where

$\beta_0, \ldots, \beta_3$ are considered constants for a given moment in time (for a detailed explanation, see [18]) and $\dot{\mathbf{u}}$ as variables with arbitrary values. Projecting $\dot{\mathbf{u}}_{des}$ (necessary acceleration to reach a desired pose) onto the plane defined by β_1, β_2, β_3 and $\beta_0 - a_{cm}$ we get the support accelerations $\dot{\mathbf{u}}_\mathbf{s}$ and through inverse dynamic calculation the supporting torques $\mathbf{t_s}$. $\dot{\mathbf{u}}_\mathbf{d}$ is

$$\dot{\mathbf{u}}_\mathbf{d} = k * (q_{ankle,des} - q_{ankle}, q_{knee,des} - q_{knee}, q_{hip,des} - q_{hip}) \tag{6}$$

where $k = 100\frac{Nm}{s^2}$. For the calculation of the desired coordinates $q_{ankle,des}$, $q_{knee,des}$ (standing position) the hip is at $y_{hip} = 0.955m$ above the ankle and q_{hip} is calculated by solving (4) with $x_{cm} = x_c$.

4 Experiments

The experiments were started in an upright standing position of the subject with both legs parallel. The subject was sitting down on a chair in a natural way (see Fig. 4).

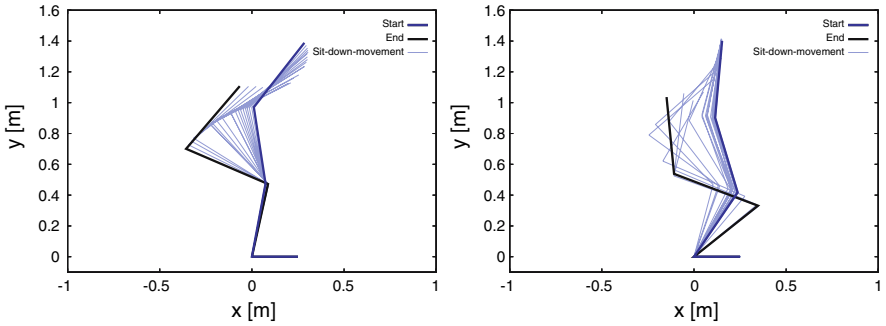


Fig. 4. The sit-to-stand movement calculated with the reference sensors (*left*) and the biomechanical model (*right*). In between a good correlation of the movement can be seen. Distortions at the start are due to different stability conditions and missing masses (head, arms) in the model. Because no muscle for bending the back or the ankle is recorded, the hip is not pushed back (and the back is straight). This also affects the resulting knee and ankle angles

During the experiment, the main force contributions from the modelled muscles come from the groups represented by $M1$ and $M2$. The only external forces affecting the motion was gravity and the chair contact force f_c (only temporary).

Figure 5 shows the angles in the ankle, knee and hip joints during the movement together with the muscle forces to show the interaction of the muscle forces with the modelled body parts.

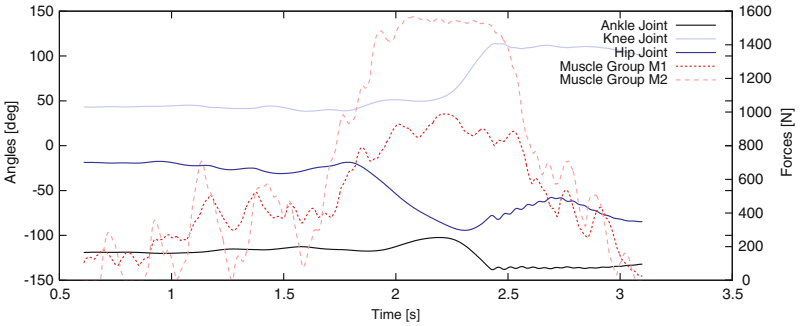


Fig. 5. Angles in the ankle, knee and hip joints during the movement are shown with the muscle forces here. Notice how the hip angle is changing when group $M1$ gets stronger. $M0$ is especially strong when the knee is bent. The ankle joint is only modified passively through the other joints

5 Discussion and Conclusion

As can be seen in this experiment, it is possible to calculate the intended movement of the subject with EMG signals. Although the presented movement here is quite simple compared to real-life situations, it should not be forgotton that for analysing the intentions of a subject the biomechanical model has to be valid only for a fraction of a second after synchronisation with the reference system. In the middle of the movement the model imitates the reference quite good. To use the algorithm as a versatile intention reader for more complex movements, a lot of work needs still to be done. More muscles have to be integrated (at the torso and shank) and the stabilty controller has to be more flexible. Also the issue of force feedback from the exoskeleton will be a major topic for further research. But the results so far look promising for future work.

References

1. S. Lee and G. N. Sridis, "The control of a prosthetic arm by EMG pattern recognition," *IEEE Transactions on Automatic Control*, vol. AC-29, no. 4, pp. 290–302, 1984.
2. O. Fukuda, T. Tsuji, H. Shigeyoshi, and M. Kaneko, "An EMG controlled human supporting robot using neural network," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1586–1591, 1999.
3. S. Morita, T. Kondo, and K. Ito, "Estimation of forearm movement from EMG signal and application to prosthetic hand control," in *Proceedings of the IEEE Int. Conf. on Robotics & Automation*, pp. 3692–3697, 2001.
4. J. V. Basmajian, and C. J. De Luca, *Muscles Alive: Their Functions Revealed by Electromyography.* Williams & Wilkins, 1985.
5. A. L. Hof, "EMG to force processing," Ph.D. dissertation, University of Groningen, 1980.

6. C. Fleischer, K. Kondak, C. Reinicke, and G. Hommel, "Online calibration of the emg to force relationship," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2004, p. about to appear.
7. *DelSys, Inc.*, http://www.delsys.com.
8. M. Thompson, M. Voigt, and M. de Zee, "Evaluation of lower extremity musculoskeletal model using sit-to-stand movement," in *19th congress of the International Society of Biomechanics*, 2003.
9. *Analog Devices, Inc.*, http://www.analog-devices.com.
10. N. Palastanga, D. Field, and R. Soames, *Anatomy and Human Movement. Structure and Function.* Butterworth Heinemann, 2002.
11. M. Epstein, and W. Herzog, *Theoretical Models of Skeletal Muscle: Biological and Mathematical Considerations.* John Wiley & Sons, Inc., 1998.
12. S. Delp, J. Loan, M. Hoy, F. Zajac, E. Topp, and J. Rosen, "An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 8, pp. 757–767, Aug. 1990.
13. R. Riener, "Neurophysiological and biomechanical modeling for the development of closed-loop neural prostheses," Ph.D. dissertation, Techn. Universität München, 1997.
14. M. Günther, "Computersimulation zur Synthetisierung des muskulär erzeugten menschlichen Gehens unter Verwendung eines biomechanischen Mehrkörpermodells," Ph.D. dissertation, University of Tübingen, 1997.
15. D. A. Winter, *Biomechanics and Motor Control of Human Movement.* John Wiley & Sons, Inc., 1990.
16. T. Kane and D. Levinson, *Dynamics OnLine: Theory and Implementation with Autolev.* Kane Dynamics, Inc., 2000.
17. *Kane Dynamics,Inc.*, http://www.autolev.com.
18. K. Kondak and G. Hommel, "Control and online computation of stable movement for biped robots," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 874–879, 2003.

Bifurcating Recursive Processing Elements in Neural Architectures for Applications in Multi-Dimensional Motor Control and Sensory Fusion in Noisy/Uncertain Environments

E. Del Moral and E. Akira

Polytechnic School of the University of Sao Paulo, Department of Electronic Systems Engineering, Cidade Universitaria – Av. Prof. Luciano Gualberto Sao Paulo, SP, CEP 05508-900 – Brazil
emilio_del_ moral@ieee.org, eakinto@lsi.usp.br

Abstract. This paper presents a neural architecture for the implementation of a robust mapping between multi-sensor information and the desired multi-dimensional motor control. Such architecture presents tolerance with respect to noisy or faulty stimuli, by means of coding and storage of information in assemblies of bifurcating recursive nodes that exhibit chaotic dynamics. An array of multiple time-dependent inputs, dealing with logical and analog variables, is sensed and drives the control system, which is implemented through the coordinated operation of a network of oscillators.

1 Introduction

Many of the research works [1–3] done on walking robots try to mimic the natural walking of animals (like Sony's AIBO, a dog robot) and humans [4]. These kind of natural walking and running are supposed to be the result of periodic rhythmic movements controlled by a set of (biological) neural networks in the spinal cord, forming a central pattern generator (CPG).

Another important issue in this subject is the smoothness on transition between gaits; [5] for example analyzes the stable trot gait. Works like [6] address research on the chaotic dynamic aiming a mathematical approach to some aspects of human brain. The key point for a CPG is the rhythm generator producing a periodical oscillation controlling the muscles; oscillations can be produced in several ways like the Zieliñska [7] that used a Van Del Pol oscillator.

This paper introduces the chaotic dynamics and bifurcation to address the problem of controlling robot gait, through the use of networks composed of

bifurcating recursive processing elements (RPEs), which are a type of recurrent neural network (similar to the Hopfield network) that explore chaos to perform its computation. The following section of the paper will present some aspects of this new approach on the subject, firstly reviewing some concepts on chaotic neural network and secondly presenting the structure of the robust to noise motor control unit.

Artificial neural networks [8] have been employed in a variety of applications in robotics due to, among other things, their nature of adaptive systems that can deal with changing environments, their properties as tools for the identification of non linear systems [9], their ability to implement sensor integration, and their ability to deal with noisy and faulty stimulation. In addition, neural networks also reveal their importance in this field when we bring to stage those neural architectures in which dynamics and the dealing with temporal information and motor control play important roles. This is the case of neural networks with feedback (recurrent networks), and it is also the case of neural architectures composed of nodes that exhibit bifurcation and chaotic dynamics like the ones addressed here, also known as chaotic neural networks [10–12]. These type of networks are particularly important because of their ability to produce a large variety of dynamical modalities (i.e. richness of bifurcations), what is an important element for the field of complex motor control. All these ingredients of neural networks (and chaotic neural networks in special) make them important tools for the implementation of robotic systems and mobile robots designed for uncertain environments.

In this study of networks composed of bifurcating recursive processing elements, we have in mind applications in which the following requirements coexist:

(a) the need for the generation of cyclic spatio-temporal patterns for multi-dimensional motor control
(b) the need for the sensing of digital variables, analog quantities, and static and time-dependent inputs
(c) the need for planned mapping of these sensed environment in actions through the definition of appropriate multi-dimensional motor control
(d) the need for associative memory and / or generalization mechanisms, in order to allow the dealing with faulty and noisy sensed data

2 Building Sensory-Motor Systems with Bifurcating Artificial Neural Networks

The RPEs (Recursive Processing Elements) architectures here studied are composed of coupled logistic elements whose discrete time state variable can oscillate with different periodicities and different phases of cycling [9, 11, 12]. We can look at these networks of logistic RPEs as assemblies of coupled oscillators, which have the role of controlling independent motor units that have to

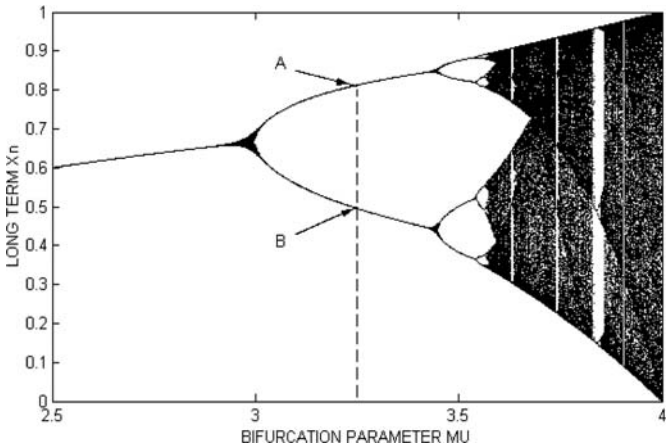


Fig. 1. The bifurcation diagram for the logistic recursion: $x_{n+1} = p.x_n.(1 - x_n)$

operate in a coordinated fashion, so to promote specific movements of interest. That is the case of multi-legged robots, where the coordinated operation and the relative phase of action of the motor units have a key role in the effective reaching of movement tasks. Each node of the Coordinated Multi Oscillator Module described ahead corresponds to an oscillator based on the logistic map. Figure 1 shows the bifurcation diagram for the logistic recursion, whose mathematical description is given in the figure caption as well ahead in (1). In that diagram, the horizontal axis represents the value of the bifurcation parameter p, and the vertical axis represents the values visited in the long term by the state variable x. The marked points A and B exemplify the specific period-2 cycle that emerges when p is 3.25. With different choices of p, we can reach different amplitudes of oscillation, and this together with the phase of oscillation between values A and B allow for the generation of diverse spatio-temporal patterns for motor control.

The bifurcating logistic nodes of the network addressed in our studies (responsible for the control of N motor units) are arranged to operate in periodic modes, and they interact with each other so to promote collective coordinated operation. Each one of the N nodes is a recursive element in which the state variable evolves in discrete time through (1), i.e., the logistic map. The coupling between nodes i and j is defined based on a Wij connection matrix, which drives a phase locking mechanism and promotes selected synchronization of the network elements. Equations (2) and (3) reflect such a synchronization driving mechanism.

$$xi_{n+1} = R_{pi}(xi_n) = pi_n \cdot xi_n \cdot (1 - xi_n) \tag{1}$$

$$di = -(xi - k_1) \cdot \left(\sum_j W_{ij} \cdot (x_j - k_1) \right) + k_2 \tag{2}$$

$$\Delta pi = di \cdot c - d \tag{3}$$

The mechanisms described in (1) to (3) are able to promote planned synchronism between subsets of logistic oscillators, according to the specific needs of coordinated operation of multiple motor elements which are under the control of the network. With that, we are able to implement a module composed of coupled RPEs, named here Coordinated Multi Oscillator Module, or simply CMOM. Such a module is based on the concept of attractor networks, where desired spatio-temporal modalities are embedded in the architecture. Thus, motor units can be controlled so to operate accordingly to a specific collective modality desired by the application. For the programming of the CMOM neural network, the connection matrix Wij is defined through a calculation that blends the outer product correlation matrix of the desired relative phase patterns (similarly to the programming of a Hopfield network [8, 9]) and normalizing factors that account for non homogeneous patterns of amplitudes of the oscillators.

The specific task to be performed by the CMOM architecture depends on the sensed environment: the command of its N motor units is implemented by additional modules through the definition of assemblies of sensory nodes, which can continually interact with the CMOM subsystem. These modules, with multiple sensory units of varied forms, are named here MSUs (Multiple Sensors Units), and they are responsible for the mapping of the sensed variables into compatible stimulation patterns for the CMOM subsystem. For that compatibility between nodes of the MSUs and nodes of the CMOM, we explore adequate features of the logistic recursion. Its phase of oscillation can be used either for the coding of binary variables or for the coding of the sign of bipolar analog variables; the distance between values A and B illustrated in Fig. 1 can be used for the coding of magnitudes in analog variables. In this way, binary inputs can be coded through phase of cycling of the sensory nodes, while analog inputs can be coded through the amplitude of oscillation. In addition, time-dependent inputs are naturally modeled by the activity of the oscillator-like sensory nodes.

With this homogeneous approach for the representation of external information (MSUs) and internal variables (CMOM), we are able to use the same coupling mechanism described in (2) and (3) for the communication between MSU modules and the CMOM module. This is represented in Fig. 2, which shows an illustrative example of MSU + CMOM arrangement. The Coordinated Multi Oscillator Module - CMOM network – represented in the figure is formed by only two logistic oscillators, L1 and L2, and the Multiple Sensory Units Module – MSU network – has only two sensors as well, Se1 and Se2. In real applications, the sizes f the networks are larger than in this illustrative diagram.

Notice that the time-dependent sensory nodes give room for the coordinated operation of independent assemblies of CMOM oscillators, since they make possible the sensing of exterior multi-dimensional spatio-temporal struc-

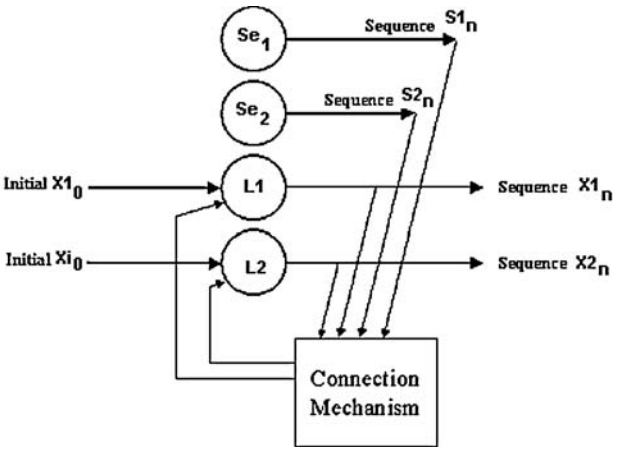


Fig. 2. An illustrative example of the CMOM + MSU architecture. The L's represent logistic nodes of the CMOM module, and the Se's represent the sensory nodes of the MSU module

tures, being therefore a potential mechanism for cooperation of multiple autonomous robots. This is also an important tool for modular design.

3 Characterizing Robustness to Noisy Sensing Inputs

The associative memory nature of the CMOM module allows for robustness of the generated multi-dimensional motor control with respect to noisy or faulty control being originated by the MSU modules – Multiple Sensory Units Modules.

Figure 3 shows a simple case of characterization of robustness to noise: the CMOM module is being used to generate multi-dimensional oscillating patterns of bipolar binary type (non-analog nature); the sensed inputs, either static or time-dependent, are also of bipolar binary nature. The input noise and the output error are measured in terms of normalized Hamming distances of the input and output binary patterns with respect to the ideal (noiseless) operation. The level of noise in the input stimuli here has to be understood as the percent of binary sensors that are being flipped due to noise, with respect to one of the situations of ideal noiseless stimuli. More concretely a situation of noiseless stimuli corresponds to an input pattern that has some meaning to the sensory-motor system, and therefore was previously stored in the Wij connection matrix of the CMOM neural network. With respect to the output error of the CMOM module represented in the vertical axis, it corresponds to the percent of bipolar motor units that will be out of the proper phase required for the proper movement associated to the current input. As can be seen from the plots, we have a good robustness of the output patterns with respect to stimuli with up to 30% of noise.

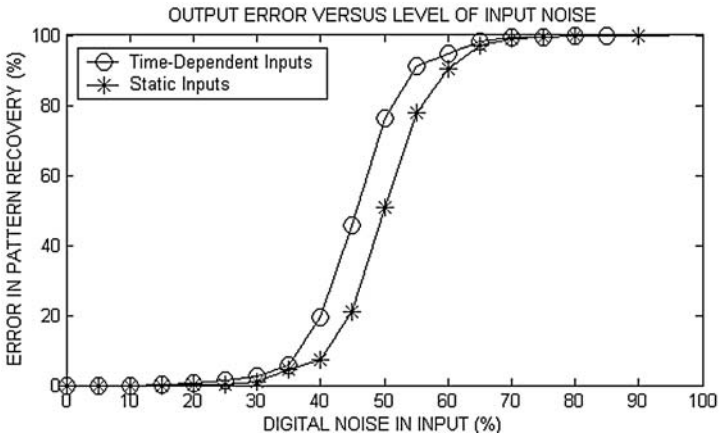


Fig. 3. Characterization of the noise robustness of the associative memory architecture implemented in the CMOM module

Although the building blocks of the MSU modules are like logistic map oscillators (as represented in Fig. 2), for which the techniques for phase locking, collective storage, and fusion and processing of information were developed, the input and output variables in a general application have their specific ranges for magnitude and average values. In order to promote the matching of these application features with the specific features of the logistic oscillators, interface modules to deal with binary inputs, continuous bipolar analog inputs, and continuous bipolar analog outputs are needed. They essentially have to implement the coding of analog and binary variables based on phase and amplitude of oscillations (distance between values A and B), as described in Sect. 2 of the paper. With that, the conversion between reasonably general classes of inputs and the particular forms of data representation and information storage proper to the RPEs networks operation is achieved. This diversity of inputs is represented in Fig. 4. In addition, the figure presents the inclusion of an additional block to the structure MSU + CMOM: the block for the processing of "command patterns", which allows context switching through the redefinition of the Wij matrix of the CMOM module.

Acknowledgements

We would like to thank the University of Sao Paulo (Brazil), FAPESP, FINEP and CNPq, for supporting this work.

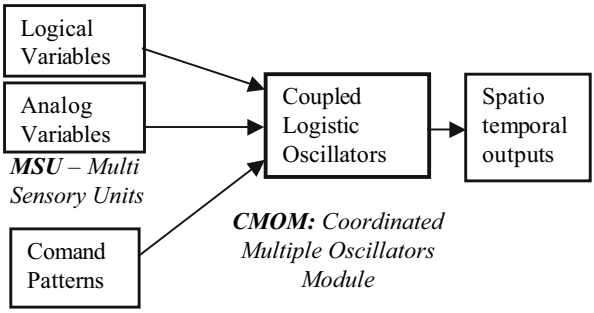


Fig. 4. The whole arrangement of the CMOM+MSU architecture. The blocks with logical and analog variables represent a multi-sensor system. The spatiotemporal outputs control motor units that have to work in coordination

References

1. Nakada K., Asai T., Amemiya Y. (2003) An Analog CMOS Circuit Implementing CPG Controller for Quadruped Walking Robot. In: Proc. of the 2nd International Symposium on Adaptive Motion of Animals and Machines, Kyoto
2. Ijspeert A.J., Billard A. (2000) Biologically inspired neural controllers for motor control in a quadruped robot. In: Proc. of the International Joint Conference on Neural Network
3. Lewis M.A. (1992) Genetic Programming Approach to the Construction of a Neural Network for Control of a Walking Robot. In: Proc. on IEEE International Conference on Robotics and Automation
4. Azevedo C., Poignet P., Espiau B. (2004) Articial locomotion control: from human to robots. Robotics and Autonomous Systems, vol 47, Issue 4, pp. 203–223
5. Karazume R., Yoneda K., Hirose S. (2002) Feedforward and feedback dynamic trot gait control for quadruped walking vehicle. Kluwer Academic Publishers
6. Tsuji S., Ueta T., Kawakami H., Aihara K. (2002) Bifurcation of Burst Response in Amari-Hopfield Neuron Pair with a Periodic External Force. Trans. of The Institute of Electrical Engineers of Japan - Special Issue Paper, vol 122-C Number 9
7. Zieliñska T. (1996) Coupled oscillators utilised as gait rhythm generators of a two-legged walking machine. Biological Cybernetics. 74, pp. 256–273
8. Haykin S. (1999) Neural Networks: a Comprehensive Foundation. 2nd edition, Prentice Hall, Upper Saddle River, NJ
9. Farhat N., et al. (1994) Complexity and Chaotic Dynamics in Spiking Neuron Embodiment, in Adaptive Computing, Mathematics, Electronics, and Optics. SPIE Critical Review, vol CR55 pp. 77–88, SPIE, Bellingham, Washington
10. Aihara K., Takabe T., Toyoda M. (1990) Chaotic Neural Networks. Physica Letters A, vol 144(6,7), pp. 333–340
11. Del-Moral-Hernandez E. (2003) Neural Networks with Chaotic Recursive Nodes: Techniques for the Design of Associative Memories, Contrast with Hopfield Architectures, and Extensions for Time-dependent Inputs. Neural Networks, Elsevier, vol 16, pp. 675–682

12. Del-Moral-Hernandez E., Lee G.Y. and Fahat N. (2003) Analog Realization of Arbitrary One Dimensional Maps. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol 50, pp. 1538–1547
13. Del-Moral-Hernandez E. (2003) A Chaotic Neural network Architecture for the Integration of Multi-Sensor Information and the Production of Coherent Motor Control: An Architecture with Bifurcating Recursive Processing Elements. 13th International Symposium on Measurement and Control in Robotics – Toward Advanced Robots: Design, Sensors, Control and Applications, ISMCR'03, pp. 101–106, Madrid

The Effectiveness of Energy Conversion Elements in the Control of Powered Orthoses

S.C. Gharooni[1], M.O. Tokhi[1], and G.S. Virk[2]

[1] University of Sheffield, UK;
[2] University of Leeds, UK

Abstract. The purpose of this study is to analyse and measure humanoid bipedal energy as a constraint in development of humanoid walking robots. For this purpose a biped model based on anthropometric data of human is developed to simulate human walking. Total energy in humanoid biped walking including kinetic energy of all segments in translation and rotation as well as potential energy are measured and calculated using Visual Nastran and Matlab software packages. The total energy in the biped is compared with total external power exerted by the controller during walking. Two control strategies in swing phase, closed-loop PID and open-loop fuzzy logic controller are investigated, and the latter has been found to perform much better in this specific context.

Key words: Control, energy conversion, powered orthosis, stored energy

1 Introduction

One of the main problems encountered in the development of an independent assistive walking system such as powered orthosis is the energy requirement for powering the device, see Virk et al. [2]. Powering and actuation for bio-robotic walking in orthoses have been investigated in terms of available prototypes and the current robotic technology and reported recently in the literature. The power requirements and sources for orthoses have been identified and studied. This includes the potential technology and future development of the sources of power such as fuel cell and nano-combustion engines. The details are published in Virk [3]. Researchers have also explored mechanically powered systems. In these systems, devices such as lightweight knee-ankle braces are fitted with a DC servo motor and a motor-actuated drum brake coupled to the knee joint with a ball screw so that the device can power or provide controlled

* Corresponding author: Dr. M.O. Tokhi, Department of Automatic Control and Systems Eng., University of Sheffield, S1 3JD, UK o.tokhi@sheffield.ac.uk

damping at the knee. The orthosis can be modular and designed to slip into a pair of trousers, see Popovic et al. [1]. However these types of orthotic systems are bulky because of the NiCd rechargeable batteries and the motor unit and so they are not practical to use.

The main restrictions for development of a powered orthosis are actuator size and weight; they must be easy to carry and easy to don and doff. Another issue is the power source such as battery, with important parameters being its size and weight, and how long it will last on one change.

In principle, there are significant conceptual differences between a biped robot and human walking. The biped robot, with its limbs full of sensors and electric motors, is normally very complex and heavy. Its power consumption is large, and the batteries used may be drained within a few minutes. Van der Linde has compared the robot's power consumption with that of a human; biped such as the Honda robot uses sixteen times as much power to move as a human does. Despite this efficiency of human gait in comparison to a biped robot, human movement is still inefficient as in cyclical movement, where energy is constant, human muscles suffer from fatigue and become tired after some exercise.

2 Humanoid Model

A realistic humanoid model was designed based on a typical human body height and weight specification, see Winter [4]. The model, based on simple geometrical shapes, was developed within the Visual Nastran (VN) software environment, so as to enable its motion control from VN as well as from Matlab/Simulink. The control strategy for the humanoid walking cycle was developed, for control of the trunk balance as well as walking phases. The simulation highlights the major dynamic and control issues of the walking cycle.

The main characteristics of the developed humanoid model are: total mass $M = 70\,\text{kg}$ and leg length 42 cm, thus height $L = 42/0.24 = 175\,\text{cm}$ (Winter, 91). Thus considering the model height as 175 cm, each segment height was calculated. Table 1 shows the segments weight when the total body weight is

Table 1. Weight and height of body segments

Segment	Weight (kg)	Height (cm)
Foot length	1.015	26.6
Shank	3.255	43
Thigh	7.000	42.9
Head & Neck	5.810	31.85
Trunk	34.79	50.4

70 kg. The centre of mass and inertia have also been considered in the model as further specifications of body segments.

A suitable control strategy is synthesised within Matlab/Simulink so as to track a set of reference trajectories for the leg segments, and to balance the trunk position throughout the walking cycle. The walking cycle is very important to simulate the humanoid model. Human walking comprises three basic phases or stages, namely, swing phase, stance phase and. double-support phase. These are shown in Fig. 1. For each phase a suitable control action was uses so as to maintain humanoid gait stability.

3 Energy Analysis

Energy is defined as the capacity of an object to do work. There are two main types: potential and kinetic energy. Potential energy is the energy that an object possesses as a consequence of its position and/or state. The two main forms of potential energy are gravitational potential energy and strain/elastic energy.

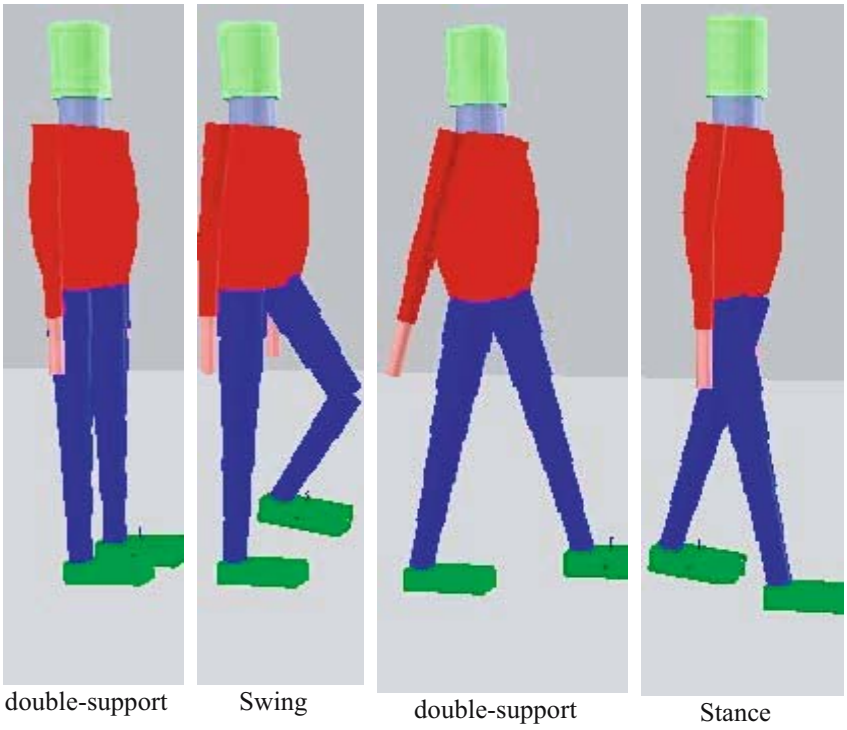


Fig. 1. Different walking stages of humanoid model

Energy measurement and analysis include five interrelated concepts that are used to describe the energy performance of a locomotion system:

- Energy storage,
- Energy return,
- Total energy,
- Dissipated energy, and
- Efficiency.

In order to evaluate the system performance, first the total external power given to the system should be identified. The external power given to a system at each joint is defined as:

$$P_i = \boldsymbol{\tau}_i \times \omega_i$$

where P_i represents power input, $\boldsymbol{\tau}_i$ the applied external torque and ω_i the angular velocity of joint i. The energy at the joint can be calculated by integration of power over period of time as:

$$E_i = \int_{t_0}^{t} P_i dt$$

The energy stored in the system is the sum of the total potential energy due to gravitational forces and strain/elastic energies. If the efficiency of the system is 100% the total input energy should be equal the total amount of stored and kinetic energy at the end of a proposed period.

4 Conversion of Energy During Gait

In cyclical movement potential energy (derived from height) and kinetic energy (derived from speed) can alternate in a practically perfect cycle. This results in a very efficient mechanical motion. This motion is similar to swing motion on a swinging robe or a simple pendulum. Humans and bipeds both use this principle during the stance and swing phases of gait. Thus, walking may be considered as an inverted swing, but a number of aspects make walking considerably more difficult than swinging, including switching between legs, progressing the foot, stability, pushing off, and linked movements. In this study the effectiveness of energy conversion in the swing phase of gait in powered orthoses is investigated.

Human gait after the first step is (starting from a stationary position) cyclical movement in which the centre of mass moves horizontally if walking is on level ground. This movement results in periodic swing and stance phases of the two legs. During walking the centre of mass for the trunk goes up and down slightly but the major movement is in the direction of walking not against gravity. In this process as the centre of mass goes up kinetic energy is stored in the body as potential energy and movement results due to this potential energy.

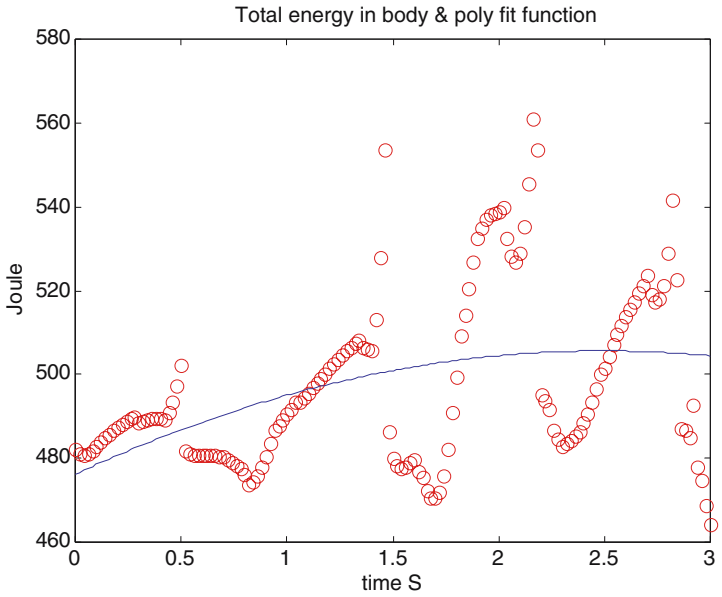


Fig. 2. Energy inside biped (*circles*) and polynomial cure fitting (*line*)

An insight into energy analysis in biped reveals that the total energy in the system after a few steps will remain constant, as shown in Fig. 2. This implies that no further energy is required for walking on level ground (assuming no losses). This is because during level walking, the centre of mass moves horizontally and no work is done against the gravity.

5 Energy Analysis During Swing Phase of Biped

Progressing the foot is an important cyclical movement during gait. During this phase the centre of mass of lower extremity periodically moves up and down while the foot advances in the direction of walking to prepare for weight acceptance. Figure 3 shows the leg modelled as a planar, two-segment linkage of rigid bodies. The parameters a, b, c, m_1 and m_2 represent the centre of thigh mass, thigh length, centre of shank mass, thigh and shank mass taken from a Visual Nastran biped model. The lower extremity is assumed to consist of two segments;

- The thigh as the upper segment with lumped mass m_1 located at distance a from the hip joint.
- The shank and foot as the lower segment, which is modelled as one rigid with lumped mass m_2 positioned at a distance c from the joint.

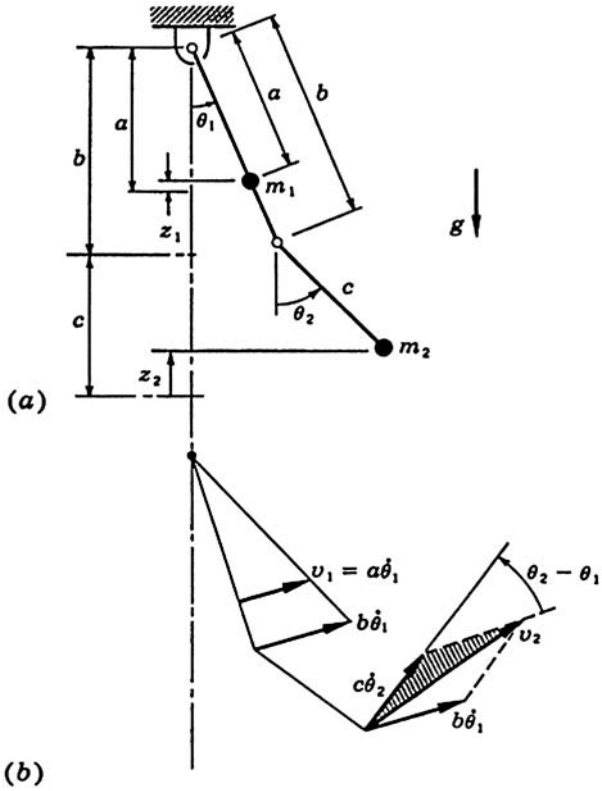


Fig. 3. (**a**) Double pendulum model (**b**) Velocity diagram

These two segments, which freely swing relative to each other, are modelled as a double pendulum. The hip and knee angles θ_1 and θ_2 are the outputs of the model.

The potential and kinetic energies are define as:

$$\text{Potential energy} = m_1 g z_1 + m_2 g z_2$$
$$\text{Kinetic energy} = \frac{1}{2} m_1 \nu_1^2 + \frac{1}{2} m_2 \nu_2^2$$

In order to express these in terms of the generalized coordinates, geometric relations are used as:

$$z_1 = a\,(1 - \cos\theta_1)$$
$$z_2 = b\,(1 - \cos\theta_1) + c\,(1 - \cos\theta_2)$$

From the diagram and the kinematic relations are:

$$\nu_1^2 = a^2 \dot{\theta}_1^2$$
$$\nu_2^2 = b^2 \dot{\theta}_1^2 + c^2 \dot{\theta}_2^2 + 2bc\dot{\theta}_1 \dot{\theta}_2 \cos\,(\theta_2 - \theta_1)$$

The above equation uses the cosine law applied to the shaded triangle in Fig. 3(b). The energy equations then become:

$$T = \frac{1}{2}\left(m_1 a^2 + m_2 b^2\right)\dot{\theta}_1^2 + \frac{1}{2} m_2 c^2 \dot{\theta}_2^2 + m_2 bc\dot{\theta}_1\dot{\theta}_2 \cos\left(\theta_2 - \theta_1\right)$$
$$V = \left(m_1 a + m_2 b\right) g\left(1 - \cos\theta_1\right) + m_2 cg\left(1 - \cos\theta_2\right)$$

The above energy analysis method was applied to the swing leg model in Visual Nastran. The potential and kinetic energy in one swing leg was obtained with an open loop and a closed-loop control strategy in the swinging leg. To evaluate system performance, the total external power and energy given to the leg to perform swing is calculated. Energy in the system has two parts; potential energy and kinetic energy. Moreover, the total energy (potential plus kinetic) will remain constant as long as no energy is given to or taken from the system. Thus, if the system is 100% efficient then the total external energy given to the system should equal the total energy inside the system.

6 Results

The energy equations developed in the previous section were applied to the Visual Nastran biped model and potential and kinetic energy in one swing leg plotted for an open-loop control strategy comprising a fuzzy logic controller (FLC) and a close-loop control strategy comprising a PID controller. The results showed different efficiencies for the same system with the two control algorithms with and without a conversion energy element (spring). An external elastic element on the knee joint can drive natural and desired trajectory until mid swing. In open-loop conversion element in the knee joint was selected so that the trajectory of plant was achieved from elastic energy stored in the spring and half of the trajectory from mid-swing was controlled with open-loop FLC. The results of energy analysis in FLC open-loop control are shown in Fig. 4.

Figure 5 shows the result with closed-loop, using PID control. Note that in certain parts of the trajectory (A and B) the power is negative. Negative power implies that energy is dissipated in the system by working against the actuators. This is because the closed-loop control system forces the plant to follow the desired trajectory. For this reason the total amount of input energy in closed-loop is more than the energy inside the system as compared with the open-loop system. This is the price paid for tracking the given trajectory. It is evident that optimal and natural trajectory can save more energy.

The same trajectory was used with the FLC approach. The external elastic element on the knee joint drives natural and desired trajectory until mid swing. Then fuzzy control can drive the end of swing phase with minimal energy. With PID control the energy inserted into the system was 15 times more than the energy inside the system. With FLC the energy applied to the system was substantially small. In addition, low energy for swinging leg allows less muscle fatigue which results in prolonged walking.

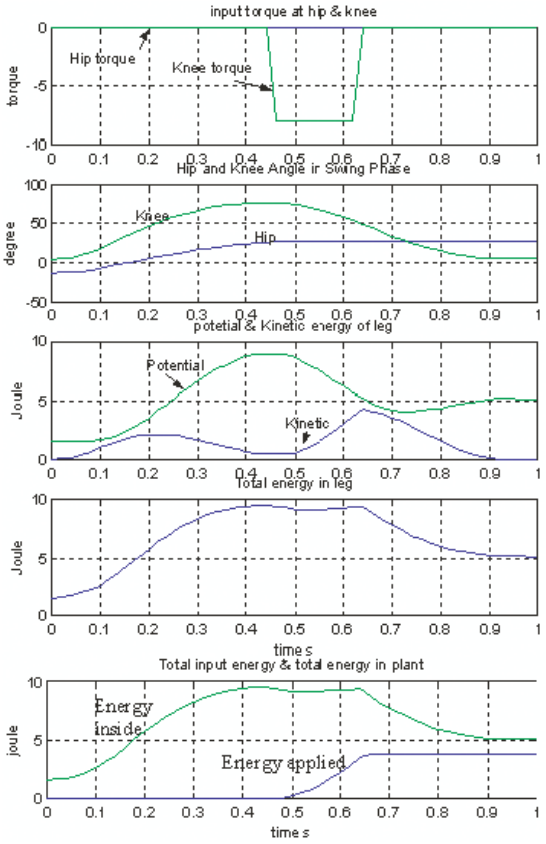


Fig. 4. Energy profile with FLC

7 Conclusions

An investigation into the effectiveness of energy conversion element has been presented in the context of bio-walking orthoses. A biped model has been developed using the Visual Nastran software package and used for evaluation of the concept of energy conversion in this study. Two control strategies, namely an FLC in open-loop configuration and a PID in closed-loop configuration have been developed and tested with an elastic element at the knee joint. It has been noted that with the closed-loop PID control strategy the energy inserted into the system was 15 times more than the energy inside the system. With FLC the energy applied to the system was substantially small. The latter is important as low energy for swinging leg allows less muscle fatigue and thus prolonged walking.

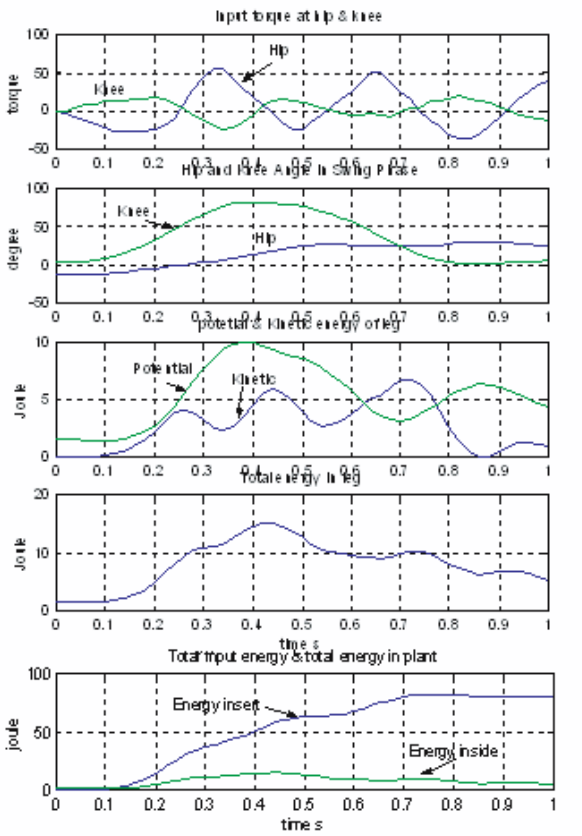


Fig. 5. Energy profile with PID control

Acknowledgment

This work was supported by the EPSRC Research Grant: GR/R10981/02.

References

1. Popovic, D. and Schwinlich, L. (1987) Hybrid powered orthoses. In *Advances in external control of human ex-tremities IX*, pp. 95–104.
2. Virk G.S., Bag S.K., Wright P.J., Gharooni S.C., Smith S.A., Sheth R., Tylor R.I., Bradshaw S., Tokhi M.O., Jamil F., Swain I.D., Chappell P.H. and Allen R. (2004) Powering and Actuation for Bio-robotic Walking Orthoses. In *35th Int. Conf. ISR*, Paris, France.
3. Virk, G.S. (2000). *Bio-Robotic Walking Orthoses – Phase 1, Appendix C.* S13/06, EPSRC Research Proposal.
4. Winter, D.A. (1991) *The biomechanics and motor control of human gait: normal, Elderly, and Pathological, 2nd Edition*, Waterloo University Press.

Kinematical Behavior Analysis and Walking Pattern Generation of a Five Degrees of Freedom Pneumatic Robotic Leg

G. Muscato and G. Spampinato

Dipartimento di Ingegneria Elettrica Elettronica e dei Sistemi,
Viale A. Doria 6 Catania Italy
gmuscato@diees.unict.it
gspampi@diees.unict.it

Abstract. The work presented provides an accurate analysis of the kinematical behavior and a brief description of the mechanical structure of an anthropometric robotic leg. The prototype realized has five degrees of freedom and three articulations actuated by five pneumatic pistons. Two pistons actuate the hip articulation, and other two move the ankle articulation, while the knee articulation has only one degree of freedom. To control the joint trajectories a two level control algorithm is implemented, in which each piston position is controlled by one piston control board, while the PC generate the joint trajectories once sampled the piston position and operated a proper kinematical inversion. Moreover, in order to ensure a more easy trajectory generation, a three-dimensional control strategy has been implemented directly into the operative space. The first walking patter trajectories have been also generated and some experimental results are reported.

1 Introduction

The high level of mobility, and the high number of degrees of freedom allows potentially biped robots to adapt and move upon very unstructured sloped terrains. Moreover, the interest of the scientist community in studying biped locomotion represents an optimal platform to point out new control strategies for complex non linear mechanical systems, where different control methodologies can be designed. In particular, pneumatic actuators, that have been adopted in this paper, represents an alternative solution with respect to the more used electric motors. In this work new control strategies and architectures have been specifically designed in order to improve the performance of the articulations movements.

Fig. 1. The five degrees of freedom mechanical structure

2 Mechanical Structure Overview

The design of the shape and the dimensions of the single parts that compose the robotic prototype are human inspired, so the dimensions of the links and the articulation movements, are similar to the corresponding biological ones [1]. The mechanical structure of the robotic inferior limb is made up of four links, corresponding to the pelvis, thighbone, shinbone, and foot, jointed by five degrees of freedom. In particular the knee joint has one rotational degree of freedom, while the ankle joint and the hip joints, have two rotational degrees of freedom. In Fig. 1 the prototype overall structure is shown.

Due to the anthropometric properties of the prototype, the biological movements of the articulations are reproduced. Each articulation of the prototype is actuated by pneumatic pistons. In particular two pistons actuate the ankle articulation, the third moves the knee articulation, while the last two pistons drive the hip articulation. In this way the hip and the ankle articulations have two degrees of freedom each, while the knee articulation is moved by one degree of freedom.

The ankle and the hip articulations are made up by a cardanic joint, that provides both roll and pitch degrees of freedom, so the two degrees of freedom are coupled by the two pneumatic pistons that actuate the articulation. In other words, a single movement of a single degree of freedom involves the motions of both pneumatic pistons and vice versa. In the Fig. 2 the mechanical structures of the articulations are shown.

The pneumatic pistons used to actuate the mechanical joints of the prototype are more similar to the biological muscles, than electric actuators [4].

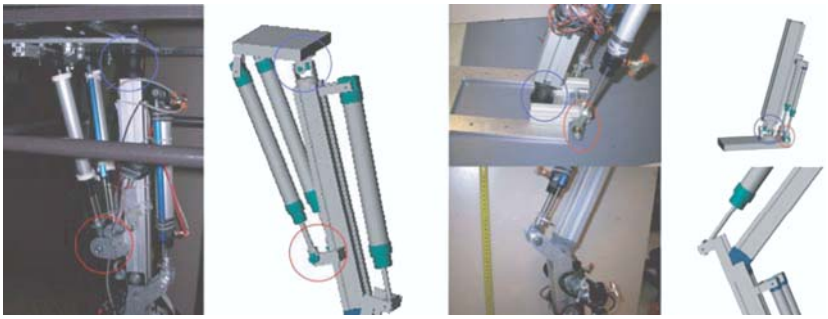


Fig. 2. Joint articulations view

Moreover the pneumatic pistons are actuated in both directions, so the articulation structure does not require the typical antagonistic scheme typical of the biological joints. As shown in other works [5, 6], the profit in using pneumatic muscles actuators, deriving from the McKibben artificial muscles, for designing and realizing biped robots has been proved.

3 A Dynamical Kinematic Inversion Algorithm

Once the piston position has been measured, the global control algorithm has to invert the kinematic relations for each joint in order to deduce the joint angles pitch and roll. The goal is to achieve a procedure to found the couple pitch-roll degrees of freedom once known the two actuator lengths. In some previous works [7] some static geometrical relations have been found between the piston lengths and the couples of pitch and roll working angles. Moreover, using the artificial neural networks, this geometrical maps become enough accurate and reliable. In this section a dynamic algorithm for solving the inverse kinematical problem is described through an appropriate Jacobian matrix.

The standard methods from the direct kinematic are used to obtain the piston length from the roll-pitch joint angles. The inverse relation can be found once known the Jacobian matrix specifically found for each articulation geometry. Looking at the structure of the hip and ankle joints, a more general geometric structure can be taken into account as a sort of general joint type, from whose the hip and ankle joint can be derived. This type of joint is shown in Fig. 3, in which the a, b, c and d segments defines the joint.

The Jacobian matrix that has to be found puts into relation the time variations of the piston lengths with the time variations of the pitch-roll degrees of freedom, as shown by (1).

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = J(\theta, \alpha) \cdot \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \end{bmatrix} \tag{1}$$

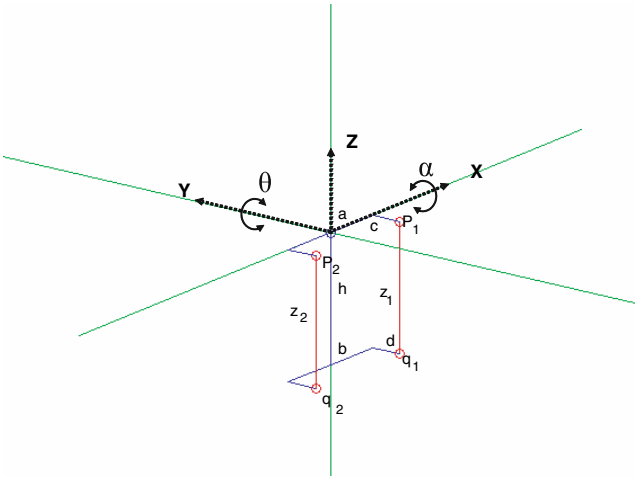


Fig. 3. General geometric structure for the hip and ankle joints

Then, according with the direct kinematics, the two inferior contact points of the pistons with the articulation q_1 and q_2 can be derived from (2).

$$q_1 = R_\theta \cdot R_\alpha \cdot \tilde{q}_1 \qquad q_2 = R_\theta \cdot R_\alpha \cdot \tilde{q}_2 \tag{2}$$

The piston length indicated with z_1 and z_2 can be derived from relations (3).

$$\begin{cases} {z_1}^2 = (P_1 - q_1) \times (P_1 - q_1) \\ {z_2}^2 = (P_2 - q_2) \times (P_2 - q_2) \end{cases} \tag{3}$$

In order to find the Jacobian Matrix, the relations (3) have to be derived with respect of the time, where the first derivative of q_1 and q_2 can be obtained deriving the (2) relations.

$$\begin{aligned} \dot{q}_2 &= [\dot{R}_\theta \cdot R_\alpha] \times \tilde{q}_2 + [R_\theta \cdot \dot{R}_\alpha] \times \tilde{q}_2 \\ &= [S(-\omega_y) \cdot R_\theta \cdot R_\alpha] \times \tilde{q}_2 + [R_\theta \cdot S(\omega_x) \cdot R_\alpha] \times \tilde{q}_2 \\ &= [S(-\hat{\omega}_y) \cdot R_\theta \cdot R_\alpha] \times \tilde{q}_2 \cdot \dot{\theta} + [R_\theta \cdot S(\hat{\omega}_x) \cdot R_\alpha] \times \tilde{q}_2 \cdot \dot{\alpha} = A_2 \cdot \dot{\theta} + B_2 \cdot \dot{\alpha} \\ \dot{q}_1 &= 7[\dot{R}_\theta \cdot R_\alpha] \times \tilde{q}_1 + [R_\theta \cdot \dot{R}_\alpha] \times \tilde{q}_1 \\ &= [S(-\omega_y) \cdot R_\theta \cdot R_\alpha] \times \tilde{q}_1 + [R_\theta \cdot S(\omega_x) \cdot R_\alpha] \times \tilde{q}_1 \\ &= [S(-\hat{\omega}_y) \cdot R_\theta \cdot R_\alpha] \times \tilde{q}_1 \cdot \dot{\theta} + [R_\theta \cdot S(\hat{\omega}_x) \cdot R_\alpha] \times \tilde{q}_1 \cdot \dot{\alpha} = A_1 \cdot \dot{\theta} + B_1 \cdot \dot{\alpha} \end{aligned} \tag{4}$$

Finally the Jacobian matrix obtained is shown by relation (5).

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = J(\theta, \alpha) \cdot \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} -\frac{P_1 - q_1}{z_1} A_1 & -\frac{P_1 - q_1}{z_1} B_1 \\ -\frac{P_2 - q_2}{z_2} A_2 & -\frac{P_2 - q_2}{z_2} B_2 \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \end{bmatrix} \tag{5}$$

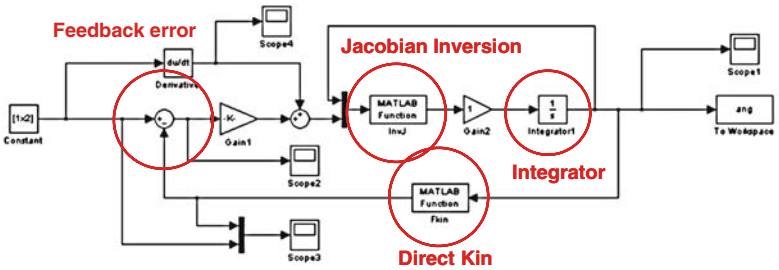


Fig. 4. Block diagram for the Jacobian inversion method

Once found the right Jacobian matrix, this can be used into a feedback integration method for inverting the kinematical problem like, for example, the inversion of the Jacobian method [2].

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \end{bmatrix} = J(\theta, \alpha)^{-1} \cdot \left(\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} + K \begin{bmatrix} e_{Z_1} \\ e_{Z_2} \end{bmatrix} \right) \tag{6}$$

A block diagram of the implementation of this method is shown in Fig. 4.

As shown in Fig. 5, the algorithm implemented achieves a good approximation of the inversion problem solution in six integration steps. The solution obtained differs from the right solutions with an error within the range $[-0.0001\ 0.0001]$, that is the five hundred times smallest than the static map approach.

4 Control Architecture Overview

To control the overall robot posture and motion stability, a decentralized two level control architecture has been implemented, in order to provide the

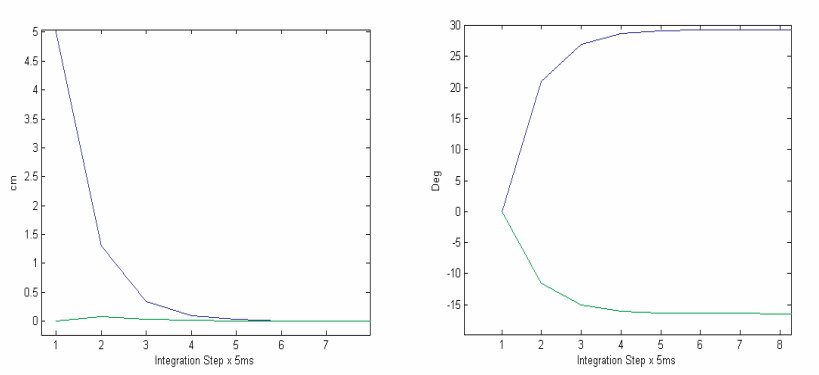


Fig. 5. Dynamical behavior of the algorithm

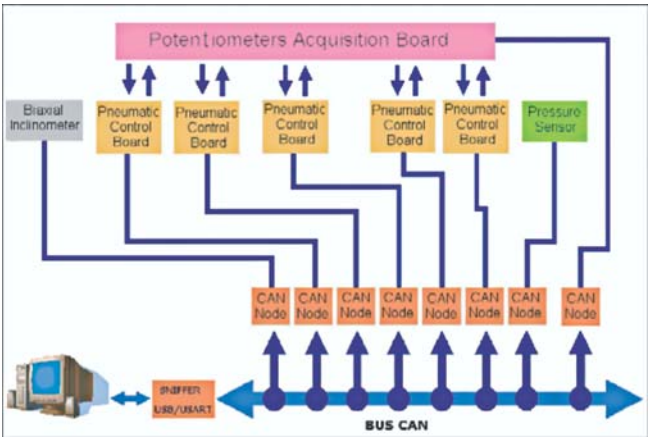


Fig. 6. Control architecture schematic

right joint trajectories [9]. The global control algorithm runs inside a standard PC, that represents the system supervisor. The PC is also connected to each sensor and actuator placed onboard the robot via a CAN communication bus. Moreover the control architecture as well as ensuring proper data rate, provides modularity and other classical advantages of using a fieldbus. The complete architecture block diagram is shown in the Fig. 6.

The piston control boards acts a position control algorithm for each piston of the prototype, that is, each control board operates on a single pneumatic piston apart from the other controllers. So there are five different piston control boards acting at the same time five different piston position controls. In this way the supervisor has to coordinate the motion of the five pneumatic piston giving on line the right position set point to each controller board, in order to guarantee the right prototype posture and stability.

The CAN (Controller Area Network) fieldbus has been used to connect the sensors ad the actuators onboard the robot with the standard PC. The 2.0 B active version of the CAN protocol is implemented at 1 Mb/sec. The most important advantage that came from using a communication bus is the possibility to synchronize piston control cards by sending piston position set points simultaneously in a single frame; In this way each CAN node receives its own set point at the same time, and works simultaneously with the other control boards increasing the stability of the global trajectory.

Despite of the previous version of the control architecture shown in previous works [8], a data acquisition board has been specifically designed and connected to the bus in order to decrease the computational load and the time resources. In particular this block performs the piston data acquisitions. In other words this node acquires the five piston position and send these information to the supervisor after a proper data request. In this way only two

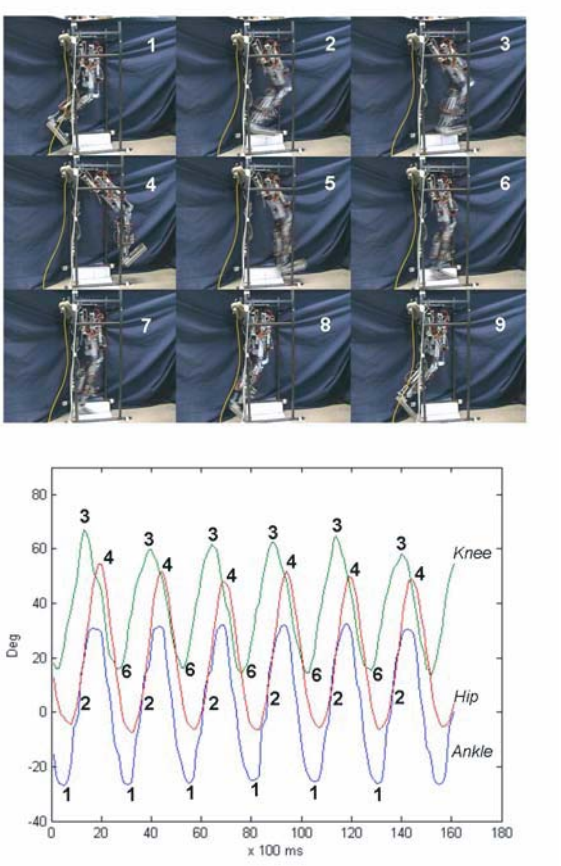


Fig. 7. Walking cycle pattern

frames have to be exchanged with the supervisor and the data acquisition module in order to acquire the overall piston positions.

An example of global articulation trajectory generation for the whole structure is shown in the Fig. 7, where a single walking cycle is reproduced. The hip, knee and ankle trajectories are shown, in which the sequence numbers 1–4 indicate the *Swing Phase*, and the sequence numbers 6–9 indicate the *Single Support Phase*. In particular the stage 5, in which it is supposed that the *Double Support Phase* occurs, connects the two phases.

In order to realize a balanced single-support phase walking pattern, a set of parabolic trajectories were also given to the robot directly in the three dimensional environment. In other words, the global control algorithm inverts the kinematic relations between the foot three-dimensional position and the hip and knee joint position, and provides to the robot the right joint trajectories in order to place the foot position in the desired three-dimensional point. In

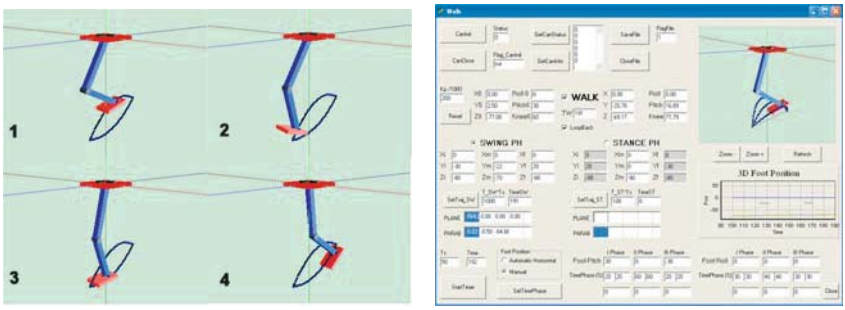


Fig. 8. Walking pattern generated into the three-dimensional environment

order to develop a proper joint trajectory sequence, it is necessary to take into account some basic properties of the human walking, as the gait repeatability in each step and the symmetry of the motions during the swing. Moreover, during the single-support phase, the COP position moves from the hill to the toe of the sole. In this way, the biped locomotion mechanism acquires stability and speed in the movements. All this has been accomplished dividing the walking trajectory into two different phases: the swing phase defined by a parabolic trajectory and the stance phase.

In particular, the stance phase is also divided into three different phases, in which the foot assumes a positive, parallel and negative pitch position with respect to the horizontal plane. In this way the COP migration from the heel to the toe of the foot is assured. All this is summarized and represented in Fig. 8.

- *Swing Phase*: Parabolic interpolation from the end to the beginning of the stance phase.
- *First part of the Stance phase*: The foot heel is in contact with the ground while the pitch moves towards the horizontal position.
- *Second part of the Stance phase*: The foot remains parallel to the ground.
- *Third part of the Stance phase*: The foot pitch moves from the horizontal position towards a negative direction in which only the foot toe remains in contact with the ground.

The walking pattern generated has been reproduced by the robot in the sequence shown in Fig. 9. The length step size is about 60 cm long while the foot maximum height is about 30 cm from the ground.

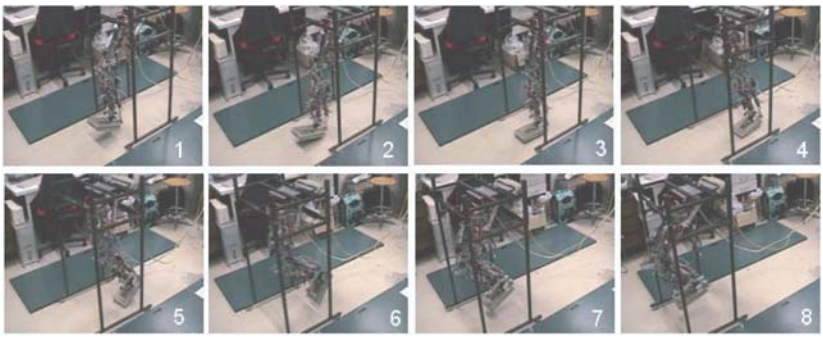


Fig. 9. A single step walking cycle

5 Conclusions

An anthropometric robot leg has been designed and realized. An accurate analysis of the kinematic behaviour of the ankle and knee articulation has been carried out and described, as well as the dynamic model. The control architecture has also been described, together with the Fieldbus communication system. The main difference between the current research and other work in the area is the use of pneumatic pistons to actuate the articulations. This means a small power weight ratio but also a complex control algorithm to maintain static posture. Otherwise they give a high level of compliance and response quickness for dynamically stable tasks like jogging or jumping.

References

1. Sardain P., Rostami M., Bessonnet G. (1998) An anthropomorphic biped robot: dynamic concepts and technological design. IEEE Trans. Syst., Man, Cybern., part A: Systems and Humans, pp. 823–838, vol. 28, No. 6, November, 1998
2. Modeling and Control of Robot Manipulators. Sciavicco – Siciliano, McGraw-Hill, 1996
3. Vukobratovic M., Borovac B. (2004) Zero Moment Point – Thirty five years of its life. International Journal of Humanoid Robotics, Vol. 1 (2004) pp. 157–173
4. Verrelst B., Van Ham R., Vermeulen J., Lefeber D., Daerden F. (2003) Concept of Combining Adaptable Passive Behaviour with an Active Control Structure Using Pleated Pneumatic Artifcial Muscles for the Bipedal Robot LUCY. International conference on humanoid Robots Humanoids2003, Karlsruhe and Munich, Germany, October 2003
5. Artrit P., Caldwell D.G. (2001) Single support balancing of a pneumatic biped. Proc. Int. Conf. Climbing and Walking Robots, pp. 835–842
6. Davies S.T., Caldwell D.G. (2001) The Biomimetic Design of a Robot Primate using Pneumatic Muscle Actuators. Proc. Int. Conf. Climbing and Walking Robots

7. Muscato G., Spampinato G. (2004) A Human inspired Five Dof robotic Leg: Kinematical Model and Control Architecture Overview. Submitted to "International Journal of Robotics and Research" March 2004
8. Guccione S., Muscato G., Spampinato S. (2003) Modelling and control strategies for bipedal locomotion – experiments with an anthropometric robotic leg. 6th International Conference on Climbing and Walking Robots CLAWAR 2003, pp. 517–526, Catania (Italy), 17–19 September 2003
9. Muscato G., Spampinato G. (2004) A Pneumatic Human inspired robotic Leg: Control Architecture and Kinematical Overview. Submitted to "International Journal of Humanoid Robotics" April 2004

Artificial Potential Based Control for a Large Scale Formation of Mobile Robots*

Wojciech Kowalczyk and Krzysztof Kozlowski

Institute of Control and Systems Engineering, Poznan University of Technology, ul. Piotrowo 3a, 60-965 Poznan, Poland
wk@ar-kari.put.poznan.pl
krzysztof.kozlowski@put.poznan.pl

Summary. This paper presents control method based on a set of artificial potential functions. The kind of used artificial potential depends on a particular objective: avoiding collisions between robots and keeping them in the ordered formation, executing task by the formation (e.g. moving formation into desired position), building formation and avoiding collisions with obstacles. In this paper we expand existing framework proposing attraction area potential function which allows to build formation and repulsion potential function which allows to avoid collisions with obstacles. Main advantage of presented approach is that it is easy scalable because control is based on general rule that determines behaviour of all robots and control is entirely distributed. Presented solutions are illustrated by simulation results.

Key words: formation of robots, mobile robots, building formation, artificial potentials

1 Introduction

Multiple robot coordination became intensively investigated area of robotics in last few years. This intensive study caused remarkable growth of knowledge of distributed systems and interactions between their autonomous subsystems. There are two main reasons of great development in this field. First is that contemporary industry requires more and more flexible, efficient and cheaper solutions. Second reason is availability of low-cost components that are necessary to build distributed and easy-scalable multi-agent solutions.

A large majority of multi-robot coordination methods developed in last years belongs to one of specified classes of approaches: virtual structure approach, [3, 6], behavioral approach [8, 11] and leader follower scheme [1, 2] (often treated as a combination of first two approaches). Each of them has

*This work was supported by university grant DS 93-121/04.

some advantages and weaknesses. There exist some solutions with characteristic features of more then one approach [7]. Detailed comparison of approaches to multi-robot coordination is presented in [12, 13].

Paper is organized as follows. In section two we describe artificial potential based control. I Subsect. 2.1 we present artificial potential function that avoid collisions between robots of the formation and keeps robots in order. In Sect. 2.2 we present attraction area artificial potential function. In Sect. 2.3 we introduce artificial potential that avoid collisions with obstacles.

In section three simulation results are given. Example of building ordered formation with robots scattered in the environment and formation motion in environment with obstacles are presented.

2 Artificial Potential Based Control

In this section we describe artificial potential based control. In paper [14] this method was used to control formation using virtual leaders. Attraction area and repulsion to the obstacles are expansion of concept proposed in that paper.

Assuming that all robots are fully actuated the dynamics of a single robot is given as follows:

$$\ddot{x}_i = u_i \ , \tag{1}$$

where x_i is position vector of the i-th robot, $x_i \in R^2$, u_i – control force vector on the i-th robot, $u_i \in R^2, i = 1, 2, \ldots, N$, N – number of robots.

In control method based on artificial potentials the role of particular robot in the formation is not explicitly specified by the high-level planner. There are no specially recognized robots. Only general rule that determines behaviour of all robots of the formation is specified. This feature is an important advantage in the case of huge formation of robots (dozens and more robots).

Control based on artificial potentials assumes that single robot is fully actuated. This is a very strong assumption. It requires robots to be in fact fully actuated or special technique must be used to transform underactuated robot into fully actuated (it can be done using feedback linearization [15, 16]). The strength of this is that formation control can be implemented separately from motion control and is not depended on architecture and properties of robots.

2.1 Interactions Between Robots

Artificial potentials around robots keep formation in order and avoid collisions between them. Artificial potentials cause forces between robots. Strength of these forces depend on distances between formation members. Forces cause repulsion when robots are too close; when they are too far forces cause attraction, but when distance exceed specified limit attraction forces are reduced to zero.

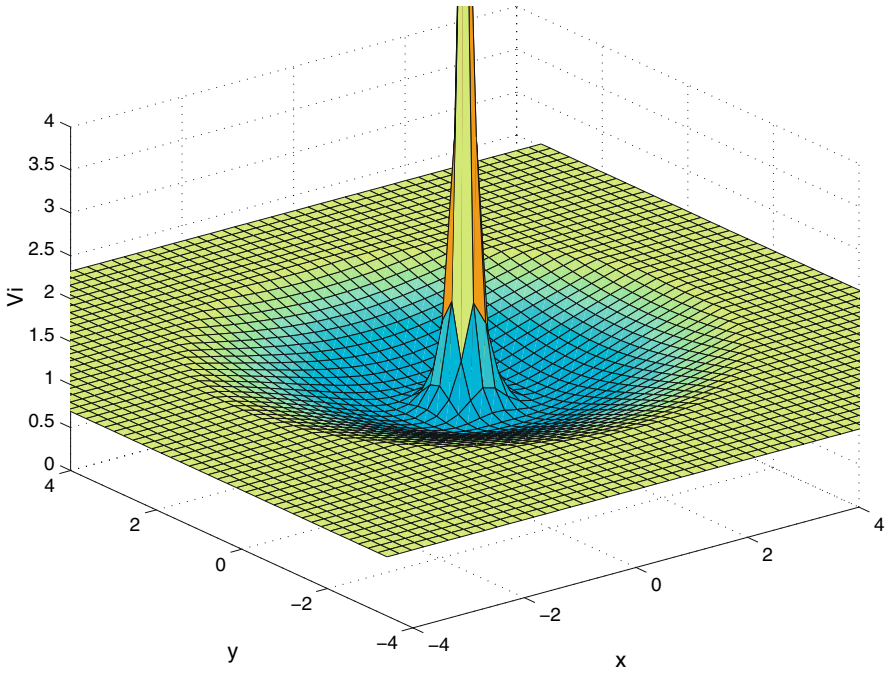


Fig. 1. Artificial potential around robots

Let denote $l_{ij} = x_i - x_j$, $l_{ij} \in R^2$ – distance between i-th and j-th robot. Artificial potentials between robots are given by the following equation:

$$V_I = \begin{cases} \alpha_I(\ln(l_{ij}) + \frac{d_0}{l_{ij}}) & 0 < l_{ij} < d_1 \\ \alpha_I(\ln(d_1) + \frac{d_0}{d_1}) & l_{ij} \geq d_1 \end{cases} . \tag{2}$$

Parameter d_0 designates equilibrium point between attraction and repulsion along line between robots, d_1 designates limit of distance between robots for which attraction force decay, α_I is the factor that determines slope of the artificial potential around robot.

Force f_I between robots caused by artificial potential V_I is given by the following equation:

$$f_I = \begin{cases} \alpha_I(\frac{1}{l_{ij}} - \frac{d_0}{(l_{ij})^2}) & 0 < l_{ij} < d_1 \\ \alpha_I(\frac{1}{d_1} - \frac{d_0}{(d_1)^2}) & l_{ij} \geq d_1 \end{cases} . \tag{3}$$

2.2 Attraction Area

Attraction area artificial potential can be used when robots are distributed in the environment and they are expected to build ordered formation in desired

position. The shape of attraction area is a circle with radius r. The value of the artificial potential in attraction area is constant and grows outside the circle.

Attraction area artificial potential can be used before execution of complex task that require robots to be initially in ordered formation.

Artificial potential that acts on robots is given by the following equation:

$$V_a = \begin{cases} 0 & 0 < s_i < r \\ \alpha_a (s_i - r)^{\frac{3}{2}} & s_i \geq r \end{cases}, \tag{4}$$

where α_a is the factor that determines slope of the artificial potential outside attraction area, s_i is the distance between i-th robot and the center of the attraction area.

Artificial potential function given by (4) is presented in the Fig. 2.

Artificial potential given by (4) causes the following force:

$$f_a = \begin{cases} 0 & 0 < s_i < r \\ \frac{3}{2}\alpha_a \sqrt{s_i - r} & s_i \geq r \end{cases}. \tag{5}$$

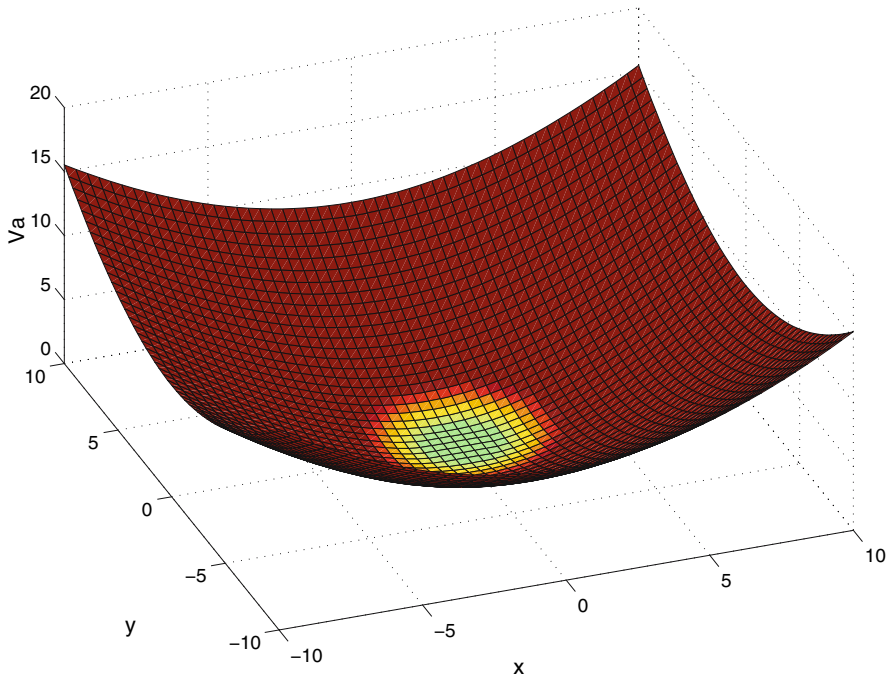


Fig. 2. Artificial potential in attraction area and around it; potential is constant in the circle in the middle and grows with distance

2.3 Repulsion to the Obstacless

To avoid collisions with obstacles we introduce repulsion potentials that act close to the obstacles and decay when distances to them grow.

Artificial potential that acts on robots when they are near to the obstacle is given by the following equation:

$$V_o = \begin{cases} not\ defined & 0 < g_i < r \\ \frac{\alpha_o}{g_i - r} + g_i \frac{\alpha_o}{(q-r)^2} & r \leq g_i < q \\ \alpha_o\left(\frac{2q-r}{(q-r)^2}\right) & g_i \geq q \end{cases}, \tag{6}$$

where r is the radius of the forbidden area that surrounds obstacle, q is the radius of the area where repulsion acts (repulsion decays outside), g_i is the distance between the i-th robot and the center of the forbidden area and α_o determines slope of the artificial potential around forbidden area. The condition $q > r$ must be fulfilled.

The value of the artificial potential in the forbidden area is not defined. Considerable slope of the potential function near border of the forbidden area avoid robots to get inside.

Artificial potential function given by (6) is presented in the Fig. 3.

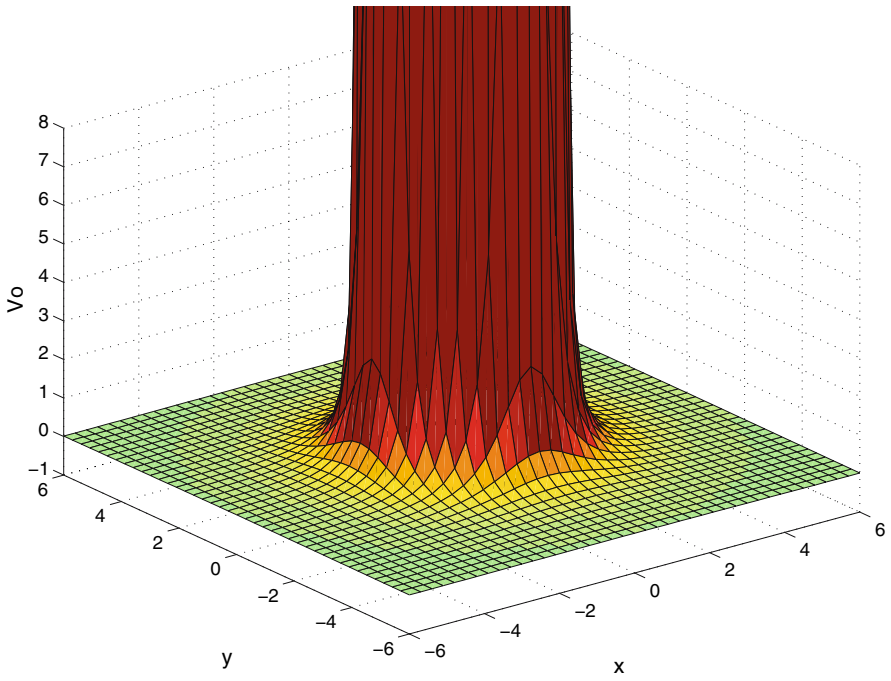


Fig. 3. Artificial potential around obstacle; inside forbidden area artificial potential is not defined, close to the obstacle it has considerable value and quickly decay when distance to the obstacle grows

Artificial potential causes the following forces:

$$f_o = \begin{cases} not\ defined & 0 < g_i < r \\ -\frac{\alpha_o}{(g_i - r)^2} + \frac{\alpha_o}{(q-r)^2} & r \leq g_i < q \\ 0 & g_i \geq q \end{cases} . \tag{7}$$

The control for i-th robot that combines repulsion to the obstacle, attraction area and interaction between robots artificial potentials is given by the following equation:

$$u_i = -\sum_{j=1, j\neq i}^{N} \nabla_{x_i} V_I(l_{ij}) - \nabla_{x_i} V_a(s_i) - \sum_{k=1}^{M} \nabla_{x_i} V_o(g_{ik}) - K\dot{x}_i \tag{8}$$

$$= -\sum_{j=1, j\neq i}^{N} \frac{f_I(l_{ij})}{\|l_{ij}\|} l_{ij} - \frac{f_a(s_i)}{\|s_i\|} s_i - \sum_{k=1}^{M} \frac{f_o(g_{ik})}{\|g_{ik}\|} g_{ik} - K\dot{x}_i \ , \tag{9}$$

where g_{ik} is the distance between the i-th robot and the center of the forbidden area around k-th obstacle, $\nabla_{x_i} V_I(l_{ij})$ is the gradient of the potential around j-th robot that acts on i-th robot, $\nabla_{x_i} V_a(s_i)$ is the gradient of the potential of attraction area that acts on the i-th robot, $\nabla_{x_i} V_o(g_i)$ is the gradient of the potential of obstacle that acts on the i-th robot, K is a positive definite matrix of linear dumping coefficients, M-number of obstacles.

When multiple obstacles exist in the environment the choice of the repulsive artificial potential must be done carefully. In some cases obstacles that are close to the other must be 'merged' and surrounded with common artificial potential. Improper choice of artificial potentials around obstacles can lead to local minima of artificial potential.

3 Simulation Results

In this section we present simulation results for control of the formation consisting of nineteen robots.

In Fig. 4 simulation results for nineteen robots that initially are scattered in the environment is presented. Used control combine interaction between robots and attraction area artificial potentials to execute task. Finally robots build ordered formation around the point $(3.0, 3.0)$.

In Fig. 5 formation of nineteen robots that drives between two obstacles is presented. Clearance between obstacle is narrower then the width of the formation. Initially formation is positioned around the origin of the coordination frame. Finally robots reach attraction area located around the point $(3.0, 3.0)$.

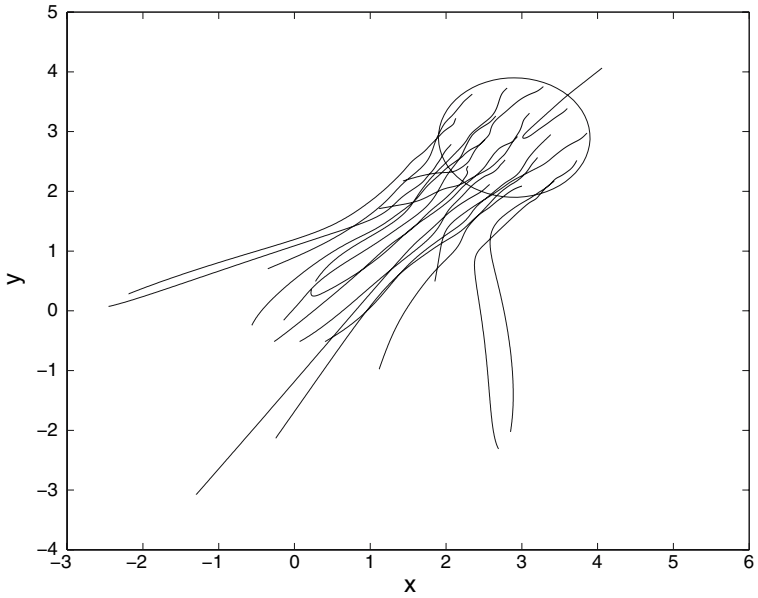


Fig. 4. Nineteen robots (initially scattered) with desired attraction area

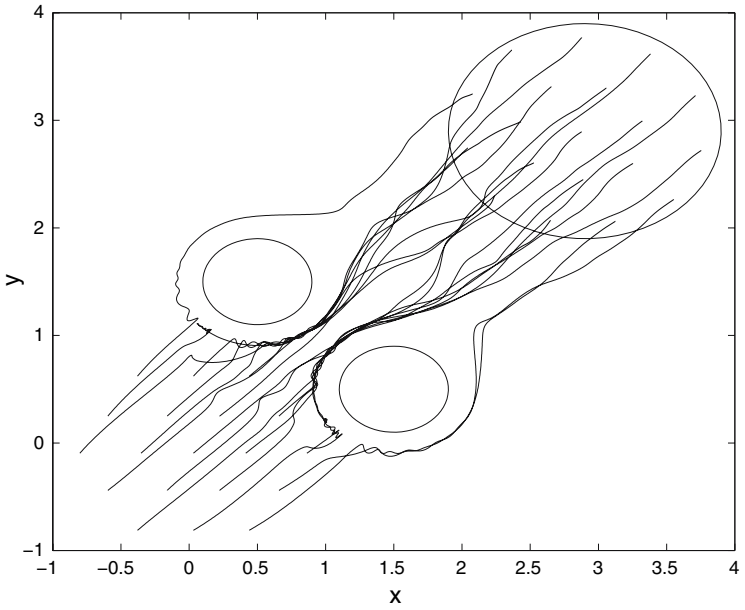


Fig. 5. Formation of nineteen robots drives between two obstacles

4 Conclusion

Attraction area and repulsion to the obstacle artificial potential functions are extensions of control framework based on artificial potentials. Proposed approaches can be used to perform motion to the desired area, to build ordered formation in the desired area and to avoid collisions with the obstacles during task execution. Main advantage of the presented approach is that control is entirely distributed. Second important advantage is that the method is easy scalable because behaviour of all robots of the formation is determined by one common rule.

References

1. R. Fierro, A. K. Das, V. Kumar, J. P. Ostrowski: *Hybrid control of formation of robots*, Proceedings of the 2001 IEEE International Conference on Robotics and Automation, pp. 3672-3677, Seoul, Korea, May 21–26, 2001.
2. J. Spletzer, A. K. Das, R. Fierro, C. J. Taylor, V. Kumar, J. P. Ostrowski, *Cooperative localization and control for multi-robot manipulation*, GRASP Laboratory – University of Pennsylvania, Philadelphia, 2001.
3. M. Egerstedt, X. Hu, *Formation constrained multi-agent control*, Proceedings of the 2001 IEEE International Conference on Robotics and Automation, pp. 3961–3966, Seoul, Korea, May 21–26, 2001.
4. P. Ogren, M. Egerstedt, X. Hu, *A control Lyapunov function approach to multi-agent coordination*, Optimization and Systems Theory, Royal Institute of Technology, Stockholm, Sweden, Division of Applied Sciences, Harvard University, Cambridge, 2001.
5. W. Kang, N. Xi, A. Sparks, *Formation control of autonomous agents in 3D workspace*, Proceedings of the 2000 IEEE International Conference on Robotics and Automation, pp. 1755–1760, San Francisco, CA, April 2000.
6. Kar-Han Tan, M. A. Lewis, *Virtual structures for high-precision cooperative mobile robotic control*, Computer Science Department, University of California, Los Angeles, 1997.
7. B. J. Young, R. W. Beard, J. M. Kelsey, *Coordinated control of multiple robots using formation feedback*, IEEE Transactions on Robotics and Automation, Vol. XX, No. Y, 2000.
8. J. M. Esposito, V. Kumar, *A formalism for parallel composition of reactive and deliberative control objectives for mobile robots*, Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, 2000.
9. J. R. Lawton, B. J. Young, R. W. Beard, *A decentralized approach to elementary formation maneuvers*, Proceedings of the 2000 IEEE International Conference on Robotics and Automation, pp. 2728–2733, San Francisco, CA, April 2000.
10. H. Yamaguchi, *A Cooperative Hunting Behavior by Mobile Robot Troops*, ICRA '98, pp. 3204–3209, Leuven, Belgium, May 1998.
11. H. Yamaguchi, *A Cooperative Hunting Behavior by Mobile-Robot Troops*, The International Journal of Robotics Research, Vol, 18, No. 8, pp. 931–940, September 1999.

12. W. Kowalczyk, *Multi-Robot Coordination*, Proceedings of The Second International Workshop on Robot Motion and Control, pp. 219–223, Bukowy Dworek, Poland, 2001.
13. W. Kowalczyk, K. Kozlowski *Coordination Strategy for Formation of Mobile Robots*, Proceedings of The 9th IEEE International Conference on Methods and Models in automation and Robotics, pp. 789–796, Miedzyzdroje, Poland, 2003.
14. P. Ogren, E. Fiorelli, N. Leonard, *Formation with a Mission: Stable Coordination of Vehicle Group Maneuvers*, Optimization and Systems Theory, Royal Institute of Technology, Stockholm, Sweden, Mechanical and Aerospace Engineering, Princeton University, Princeton, USA, 2002.
15. B. Young, R. Beard, *A decentralized approach to elementary formation manouvers*, IEEE Transactions on Robotics and Automation, Princeton University, 2002.
16. J. R. Lawton, R. W. Beard, B. J. Young, *A decentralized approach to formation maneuvers*, IEEE Transactions on Robotics and Automation, pp. 933–941, Vol. 19, Number 6, December 2003.

Mobile Mini Robots for MAS

M.W. Han, P. Kopacek, B. Putz, E. Sshierer, and M. Würzl

Institute for Handling Devices and Robotics, Vienna University of Technology
e318@ihrt.tuwien.ac.at

Summary. For practical tests concerning future industrial Multi-Agent-Systems (MAS) cheap intelligent, autonomous agents are necessary which should carry out its task in cooperative, coordinated, and communicative way with other agents. The group behavior of agents as well as the behavior of a single agent should be explored. The behavior of a single agent depends on the accuracy and dynamics of the vision system as well as the robot itself. In this paper the development of a vision system for a vision-based Hardware-MAS and of two mini robots "Roby-Go" and "Roby-Run" are described.

1 Introduction

Approximately 15 years ago the keyword "Multi-Agent-Systems (MAS)" as introduced. Multi-agent systems have emerged as a sub-field of AI that attempts to implement both an unified theory and design principles for constructing complex systems with multiple agents and their coordination in dynamical environments. At that time, computer scientists used MAS as an application example of distributed artificial intelligence. Nowadays MAS are closely related to the design of distributed systems for autonomous agents to be used in manufacturing and robotics applications. Until now, most of the realized MAS have been dealing with software-agents.

Usually our robot is a two-wheel driven mobile robot. It consists of two wheels, two DC motors, a micro controller, a communication module, a battery pack and other parts. It is a very good example for a mechatronic system. The behavior and efficiency of such a robot depends on the mechanical construction, control algorithm, and the performance and accuracy of the vision system. The vision system is responsible for the calculation of exact object's positions based on the pictures from CCD-Camera above the environment, which should be observed.

The Institute for Handling Devices and Robotics (IHRT) has developed two robots named "Roby-Go" and "Roby-Run". This paper is dealing with

Fig. 1. Robot Roby-Go

the development of these mobile mini robots which can be used as a cheap test bed for intelligent, cooperative robots and as a special application robot soccer.

2 The Mini Robot – "ROBY-GO"

The mobile mini robot at IHRT – "Roby-Go" (Fig. 1) is a two-wheel driven mobile robot with a simple, compact, robust and modular construction. Roby-Go can reach a maximum speed up to 2.5 m/s. Each of two wheels is connected by a gear with a DC motor. Each DC motor receives a command value – desired speed – as input a PWM (Pulse-Width-Modulation) signal generated by the micro controller.

Roby-Go consists of two parts; the mechanical part and the electronic part. The electronic part has a modular and open architecture and consists of the microcontroller board and the power- and communication-board.

As a controller a C167-LM from Infineon (Siemens) is used. The C167 is a 16-Bit CMOS micro controller (25 MHz) with a on Chip CAN module. It contains a CAN bus interface, offering possibilities to connect several micro controller boards for different tasks, like sensors, etc. This micro controller has 4 channel PWM units and 2 k Bytes internal RAM. The used DC-Motors provide a 2 channel digital encoder signal. This signal is used as a reference signal to control the speed of the wheels. The used micro controller has furthermore a special digital decoder unit to detect those encoder signals.

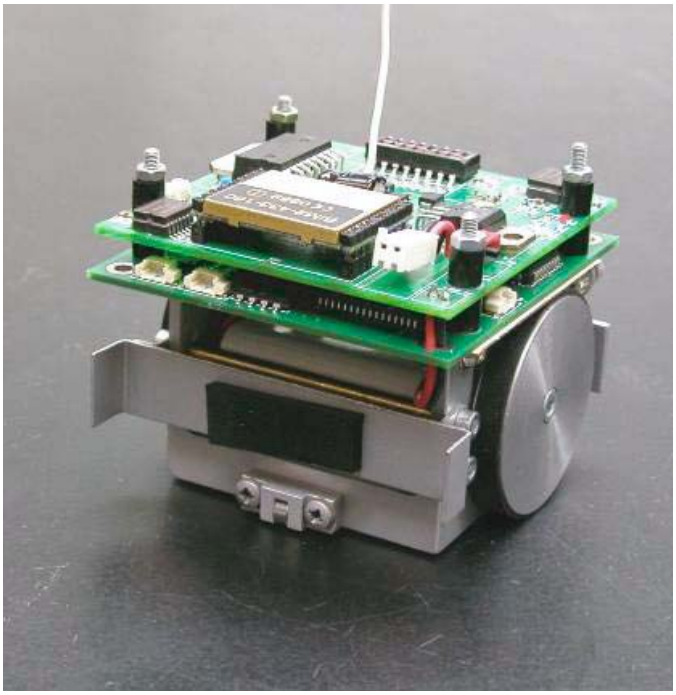


Fig. 2. Robot Roby-Run

The mechanical part consists of two DC motors connecting with a gear. In order to avoid the slippage the wide wheel and as the "tire" a ping-pong rubber is used.

3 New Generation of Mobile Mini Robot – "ROBY-RUN"

For tests of future MAS a faster moving robot is necessary. There are three main facts. One fact is that a sampling time of even 20ms is much too less to control a robot driven by DC – motors. The second fact is that there is time delay between the detection of the position and carry out the motion commands from the host computer. Additionally the exactness of the robot's position depends on the accuracy of the vision system.

According to our experiences a MAS robot should have the following features:

– Fulfill his task as exactly as possible
– Stay on its given trajectory
– Signal if the power supply is empty (lamp)

– Have an easy way to change the battery
– low-lying center of gravity
– separate power electronic from controlling electronic
– broad wheels
– with roller bearings
– a singe stage gear
– a modular and open architecture
– Reach a speed up to 3 m/s
– acceleration of more than 6 m/s2
– tolerance accuracy less than ±0.5 cm/m

A mini robot consists of the following parts:

– Power supply
– Two DC – motors with digital encoder
– Single stage gear
– Two wheels
– Micro controller for controlling the DC – motors
– Power electronic
– Radio module to receive tasks

By all these facts the development of a multifunctional robot was the goal.

According to the experiences mentioned above a new mini robot was developed. This mini robot is a two wheels driven robot, distinguished by his simple, compact and modular construction. It has two PWM controlled DC – motors. Furthermore out of the modular architecture the possibility of adding very easy sensors to the robot is given.

4 Structure of Roby-Run

The robot consists of a mechanical – the motion unit and electronic part including sensors. In case several robots are working a DIPswitch is included for the identification of the robot. The motion unit itself is connected by some kind of bus with the 'remaining part' of the robot. For controlling the motion unit velocity, angular velocity and commands (Go, Stop, Reset) will be provided. The motion unit is done as compact as possible in order to save room for other electronic parts like sensors.

The robot is seen as a smart actor fitting in the OSI layer model. It represents a complete control loop getting its demand value from a layer above. This layer above is in this step of development implemented on the control program at the host computer, but the idea is to move layer by layer to the mobile robot.

4.1 Mechanics

The mechanical system of the mobile platform consists of the following parts:

- Two wheels with special tire
- Two one stage gears
- Two DC – motors with magnetical two channel encoders
- Two side panels
- Two front panels
- Battery and battery mounting
- Special mounting for the electronic

Dimensions		
Length	75	mm
Width	75	mm
Height without electronic	49	mm
Height with electronic	75	mm
Wheel diameter	48	mm
Wheel width	8	mm
Wheel base	67	mm
Weight of robot		
Whithout batteries	340	g
Whith batteries	450	g
Specification of PWM controlled DC Motors		
Minimotor Type	2224 06 SR	
Output power	4.05	W
Speed up to	8000	rpm
Stall torque	21.2	mNm
Encoder resolution	512	ppr
Single stage gear		
Gear ratio	253	
Movement data		
Maximum speed	3	m/s
Maximum acceleration	6	m/s^2

4.2 Electronics

The mobile mini robot is connected via a radio module to its superior control unit. The electronic is built up in an open architecture. The motion unit controls the motors to follow a desired trajectory. This trajectory (as well as other demand behaviours like acceleration and etc.) has to be transferred to the motion unit. Transferring data to the motion unit or any other unit is in principle done via a bus system. As bus system the CAN bus is used. The CAN bus is very common in the automotive industry, but of course any other bus system which fulfils the real time criteria in a certain way is possible. Also

a hierarchical bus system with different busses for different tasks is thinkable. The bus of the mobile unit can be connected to another bus system for example by radio connection. This means that such a mobile platform fits into a layer model and the motion unit is part of a networked control system (Fig. 3).

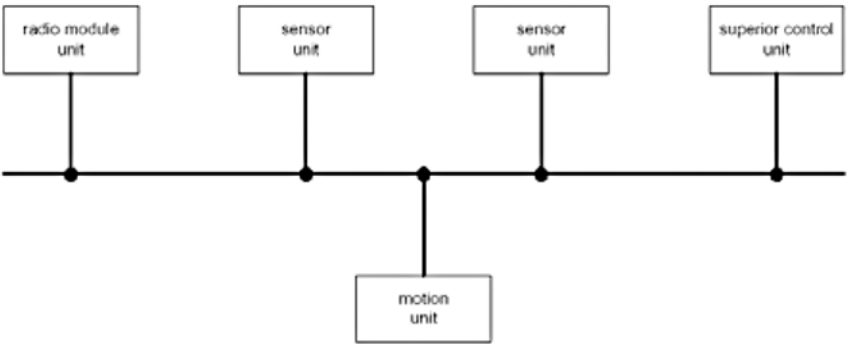


Fig. 3. Networked Control System

For the design of a mobile mini robot the considerations above mean that it consists of a motion unit and a connection via radio to its superior control unit. Remaining to the described system the radio module should be connected to a micro controller, which selects and processes the incoming information. Afterwards a bus provides the processed information to the motion unit (Fig. 4).

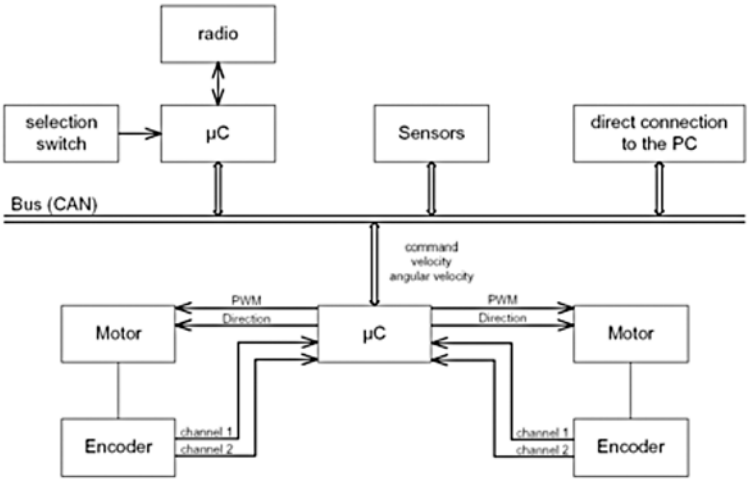


Fig. 4. Bus System for a Mobile Platform

The electronic part of the mini robot is almost the same as the of Roby-Go. The main additional features are:

– Implementation of acceleration sensors
– Shift the intelligence on the robot

The task of the micro controller board is to control both DC motors and analyze the radio data. The second board uses some I/O pins of the micro controller board and contains the implementation. Due to the fact, that the motors need much more electric current than the micro controller could provide a driver is required and a dual full bridge driver is installed. For the radio communication a special module is used, which can be plugged in this board. For selecting a robot, 8 DIP-switches are integrated. Additionally two LEDs signal if the robot is turned on, if enough power supply is available and if the radio communication works properly. The robot is supplied with 10.8 V. A potential transformer transforms this voltage to the 5V that the micro controller needs.

Power Supply		
Battery pack consists out of 9 NiMH cells with 1.2 V and 700 mAh.		
Operating times up to	40	min
Reloading time	15	min
Microcontroller Core		
C167CR-LM Infineon		(Siemens)
programmable by serial port.		
Contains CAN bus interface		
– Possibility of connecting several μC boards for different tasks.		
– Debugging of robots during operation		
Power and Communcication Board		
– Motor driver L298		
– Radio module by Radiometrix, UK BiM418F, BiM433F, Rx2 or Rx3		
Contoller		
Optional: PID, Neuro PID, or advanced algorithms		
Sampling Time	1	ms
Further features		
Acceleration sensors		

5 Vision System

The software package of a robot system consists of three parts, namely the vision system-, the communication- and strategy-module. The software package is necessary to control the robots. Vision system means detection and

recognition algorithms to handle the movement of multiple mobile robots in a dynamic, noisy environment. All calculations related to the vision system are processed in the host computer. The task of the vision system is to catch the position of the objects. The colours are used as major cue for object detection. In many cases the camera delivers 30 to 60 either half or full pictures to the frame grabber. Analog CCD-Cameras have different time intervals between two pictures. The time interval is therefore not constant. Digital cameras have not such a problem, but they are relatively expensive. The old IHRT Vision system consists of an analog colour-CCD-Camera and frame grabber which generates a frame each 33 ms and an image processing module which detects and tracks all the objects in the field of vision of the camera.

The new IHRT vision system uses a digital camera. Each robot is marked with colours. Because of the inherited noise, false detection occurs. It is eliminated by filtering off the "salt and pepper" noise. The speed of vision system plays an important role. There is a time delay (approx. 60 ms) between taking a picture with the camera, image processing, generation if motion commands, sending commands via radio frequency, reaching the motion commands and a moving robot. In the space of time between taking visual information and carrying out the command the robot is moving in somewhere. Assume that the robot moves at a maximum speed of 3 m/s and the time delay takes 60 milliseconds, the difference between robot's position when the camera takes picture and robot's position when the robot starts to move is almost 18 cm. Furthermore the time between new data provided by vision system is not constant.

Therefore it is necessary to have a fast as well as an exactly working vision system. The requirements for the new vision system were:

– full picture, not half-picture
– progressive camera instead of interlaced
– higher resolution
– higher refresh rate
– lower system load of the host computer
– extensibility of the hardware in future

It was not suitable to work with the analog system anymore. Therefore a new digital vision system was developed. Comparing to the previous analog vision system the digital vision system is

– 5 times faster than analog system
– exact and
– easy by setting of the colours in different lighting conditions.

The disadvantage is that the digital vision system is very expensive. One additional remark is through the implementation of the bi-directional communication the robot can send the level of the battery voltage which is important to check the actual voltage of the battery. During operation it is not possible

to measure battery's voltage. Therefore each robot sends the its voltage of the battery to the host computer.

6 Conclusions

In this paper the development of two mobile mini robots and vision system at the Institute for Handling Devices and Robotics of Vienna University of Technology are described. Until now the whole commands were sent by the host computer, but the robot becomes more and more intelligent, it means the intelligence is shifted to the robot step by step. Another approach is to make the robot autonomous through the implementation of different kinds of sensors, like CMOS-Camera, etc. At the moment a new robot "Roby-Speed" is developed, which is more intelligent, smaller and faster than "Roby-Run".

References

1. Aleksic, K., Han, M.W. and Kopacek, P.: (2001) "Prediction Algorithms for Multi-Agent Systems". In: Proceedings of the 10th IFAC Symposium on Information Control Problems in Manufacturing – INCOM 2001, Vienna, pp. 153–157
2. Han, M.-W., Novak, G.: "Multi-Agent-Systems and Production". In: Preprints of the 7th International Workshop on Computer Aided Systems Theory and Technology 1999 – EUROCAST'99, Sept. 29th–October 2nd, 1999, Vienna – Austria, pp. 189–192.
3. Han, M.-W., Novak, G.: Multi-Robot-Systems in Entertainment – Robot Soccer. In: Preprint of the 1st International IFAC Workshop on MULTIAGENT SYSTEMS IN PRODUCTION – MAS'99, Dec. 2–4, 1999; Vienna, Austria.
4. Han,M.-W., Kopacek P., Putz, B., Würzl, M., and Schierer, E. "Robot soccer a first step to 'edutainment' ". In the Proceedings of RAAD'03, 12th International Workshop on Robotics in Alpe-Adria-Danube Region, Cassino May 7–10, 2003
5. Kopacek, P.: (2002), "Multi Agent Systems for Production Automation with Special emphasis to SME's. Report, IHRT, Vienna University of Technology (in German)
6. Kopacek, P.; Han, M.-W.; Novak, G.: "Control Tasks in Robot Soccer". In: Proceedings of the 32nd International Symposium on Robots – ISR, Seoul, 19–21. April 2001, pp. 756–759.

Two Neural Approaches for Solving Reaching Tasks with Redundant Robots

J. Molina-Vilaplana, J.L. Pedreño-Molina, and J. López-Coronado

Department of Systems Engineering and Automation, Technical University of Cartagena, Campus Muralla del Mar. C/ Dr Fleming S/N 30202, Cartagena, Murcia, Spain

Abstract. In this paper, two solutions for learning to reach targets in 3D space with redundant robots are presented. Fist of them is based on Hyper Radial Basis Functions networks (HRBF) that learn to compute a locally linear approximation of the Jacobian pseudoinverse at each robot joint configuration. The second one, is an alternative solution that is inspired in human ability to project the sensorial stimulus over the motor actuators on joints, sending motor commands to each articulation and avoiding, in most phases of the movement, the feedback of the visual information. These models have been implemented on a visuo-motor robotic platform for reaching and grasping applications. The obtained results allow to compare the robustness and accuracy capabilities of both neural models for reaching and tracking objects as well as to give a solution when redundant robots are considered.

1 Introduction

In recent years, the interface between neuroscience and robotics is carried out by Neurorobotics research. Neurorobotics tries to investigate, model and validate neural dynamics underlying successful performance of tasks such as reaching and grasping actions by humans, and transfer validated neural algorithms into the design of control algorithms acting on bio-inspired robotic platforms. Two different neural models for achieving accurate reaching and tracking tasks are presented in this paper. The main idea is to give biologically-inspired solutions for solving the inverse kinematics of robots, without the knowledge of the internal physical properties of the robot arm, such as joint lengths and rotation and translation thresholds of each joints. Algorithms giving that kind of solutions have the advantage of avoiding the continuous calibration of the system and simultaneously to be independent from the considered robotic platform.

The mapping from kinematic plans in external coordinates to internal robot coordinates is the classic inverse kinematics problem, a problem that arises from the fact that inverse transformations are often ill-posed in the sense that the information in the data is not sufficient to reconstruct uniquely

the mapping at points where data are not available. One of the simplest assumptions is that the mapping is smooth. Techniques exploiting smoothness constraints in order to transform an ill-posed problem into a well-posed one are well known under the term of regularization theory [1]. In [5] it was shown that the solution to the approximation problem given by regularization theory can be expressed in a class of multilayer networks that the authors called Hyper Basis Function networks. The HRBF model used in this paper as a first solution to the problem of learning the inverse kinematics in redundant robotic systems, constitutes a particular case in the general scheme of regularization networks. This solution requires the continuous use of feedback visual information during both the training and performance phases of the system. The second solution proposed in this paper, avoids this need of continuous visual feedback producing movements in an anticipatory feedforward way. The workspace of the robot arm is autonomously divided in small structures of learning cells. The proposed model aims at the idea of solving the accuracy sensory-motor coordination by means of two neural networks whose interconnection allows the anticipatory behaviour of the model. These neural networks are based on Adaptive Resonance Theory (ART) [3], and the AVITE model (Adaptive Vector Integration to End Point) [2].

The neural models proposed in this work are able to combine visual, spatial, and motor information for reaching objects with a robot arm. The proposed architectures have been implemented in an industrial robot arm and capabilities of robustness, adaptability, speedy, accuracy have been demonstrated for reaching tasks.

2 The Neural Network Models for Reaching

The two neural models that have been developed and implemented in this paper are depicted in following subsections.

2.1 The HRBF Model

As [4] have previously proposed, the mapping from the desired movement direction vector Δx to the joint rotation vector $\Delta\theta$ is generated in the model according to the equation:

$$\Delta\theta = A\,(\theta) \cdot \Delta x \tag{1}$$

The elements $a_{ij}(\theta)$ of the matrix $A(\theta)$ are the outputs of the HRBF network, where index i refers to the actuator space dimension and index j refers to the 3D workspace. These elements are calculated according to the following equations:

$$a_{ij}\,(\theta) = \sum_{k} g_k\,(\theta)\, w_{kji} \tag{2}$$

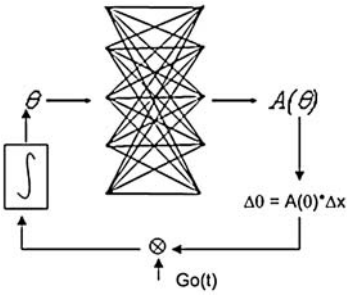


Fig. 1. HRBF model performance scheme. $Go(t)$ is a time varying speed-controlling signal (see text for details)

where w_{kij} are the network weights, the $g_k(\theta)$ are Gaussian functions, each of them with associated (and learnable) parameters corresponding to the mean and variance of the kth basis function along dimension l of the input space (joint space, Fig. 1). An action–perception cycle is induced by random joint velocity vectors $\Delta\theta^R$ (where the subscript R denotes the random movements). These joint rotations are carried out from certain joint configurations denoted by θ, that is the input to the HRBF network. Random joint rotations induce spatial displacements of the end–effector, displacements (denoted by Δx) that are measured by the stereohead vision system. The HRBF network computes an approximation of the Jacobian pseudoinverse that is used to calculate an estimation ($\Delta\theta$) of the random joint movements ($\Delta\theta^R$). An error function is constructed to derive the dynamics of the network parameters during the learning by gradient descent.

In order to allow both voluntary control of movement duration and generation of realistic velocity profiles, a movement-gating $Go(t)$ signal has been used. This signal multiplies the joint rotation commands computed by the system Fig. 1 prior to their integration at the joint angle command stage. When the $Go(t)$ signal becomes positive, end-effector movement rate is proportional to its magnitude multiplied by $\Delta\theta$.

2.2 The ART-AVITE Model

This second alternative has been designed based on how the human system projects the sensorial stimulus over the motor joints, sending motor commands to each articulation and avoiding, in most phases of the movement, the feedback of the visual information. In this way, the proposed neural architecture autonomously generates a learning cells structure based on the adaptive resonance theory, together with a neural mapping of the sensory-motor coordinate systems in each cell of the arm workspace. It permits a fast open-loop control based on propioceptive information of a robot and a precise grasping position in each cell by mapping 3D spatial positions over redundant joints.

This interconnection is based on the self-organizing Adaptive Resonance Theory (ART) [3], and the AVITE model (Adaptive Vector Integration to End Point) [2]. By one side, the self-organizing ART algorithm has the ability of solving the stability–plasticity dilemma for the competitive learning phase, allowing an anticipatory behaviour. By other side the supervised AVITE neural model permits to map the spatial-motor positions in each learning cells. As results, the proposed work is capable to combine visual, spatial, and motor information for reaching objects by using a robot arm, tracking a trajectory in which the close-loop control is only carried out in each learning cell of the workspace. The goal of the first stage of this neural model is to carry out an adaptive distribution of the robot arm workspace. After several random movements of the robot the 3D workspace is divided into small cells in whose centres the precise position of the robot joints are well known, by means of the propioceptive information and one previous learning stage. By means of a second learning stage, one neural weight map is obtained for each cell. Due to the lineal nature of the AVITE model, the spatial-motor projection is quickly computed and few steps close-loop control are required for accurate reaching tasks. The AVITE model projects the difference vector (DV), in visual coordinates between the current and desired position of the end-effector, over the incremental angular positions of the robot arm. Thus, the 3D visual distance inside the winner cell is reduced with high precision and fast operation.

3 Results

The implementation of both reaching approach has been simulated over the virtual robot simulator developed by DISA Valencia, Spain [6] with an industrial ABB robot. The base, elbow and shoulder joints have been considered for the experimentation. For both strategies, 600 trials have been considered; An hyperbolic function Go(t) has been generated for the HRBF algorithm and 23 learning cells have been autonomously generated for the ART model in the second reaching strategy. In order to evaluate the behaviour of both models, for reaching of static objects, the target and the end-effector initial positions have been {100, 900, 1000} and {600, 850, 1000}, respectively. The obtained results are represented in Figs. 2 and 3.

Experiments for reaching objects when perturbations exit in their position, have been simulated for both neural models. For this case, the target has been displaced form {1000, 500, 900} to {700, −900, 400}. Evolution of targets and 3D end effector positions are represented in Figs. 4 and 5.

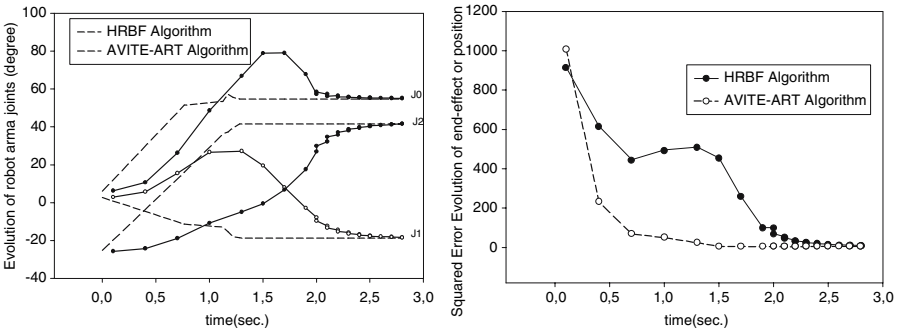


Fig. 2. Error and robot arm joint evolutions for a reaching task

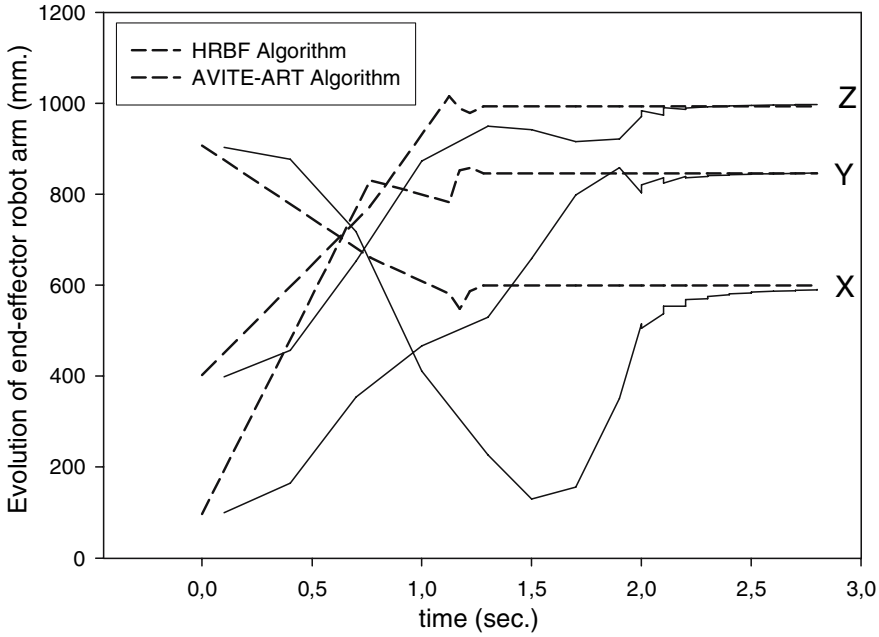


Fig. 3. Comparison of end-effector evolution in both reaching strategies

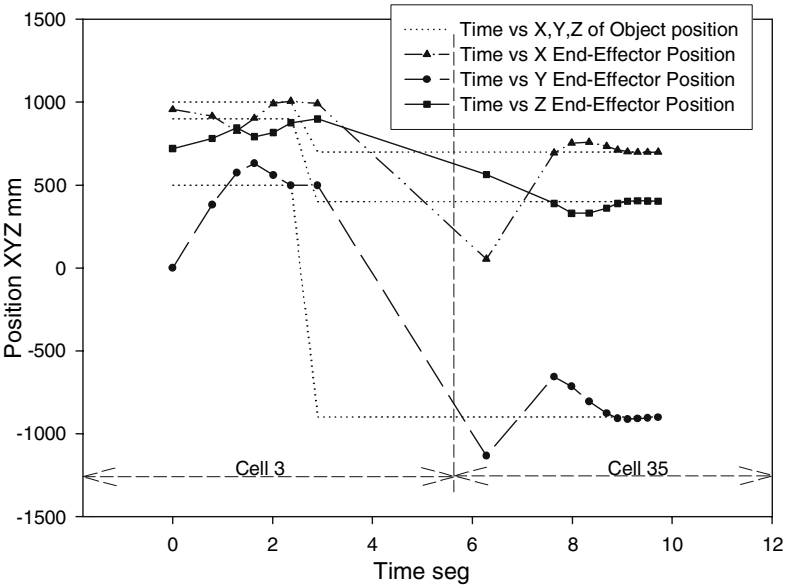


Fig. 4. Spatial position evolution by applying the AVITE-ART model

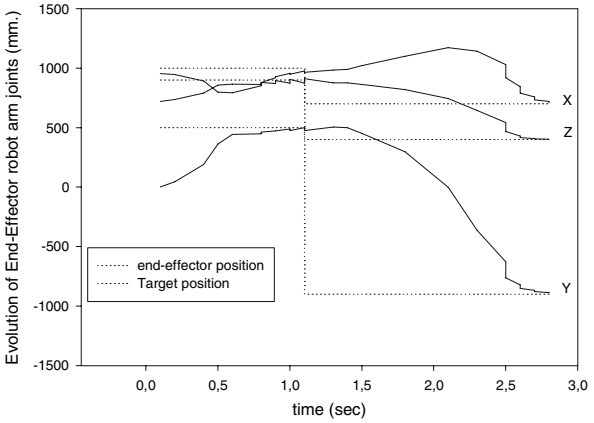


Fig. 5. Spatial position evolution by applying the HRBF model

4 Conclusions

In this paper, two robust visuo-motor architectures applied to redundant robots for reaching tasks, have been implemented by means of the virtual robot simulator, developed by DISA Valencia, Spain [6] with an industrial ABB robot, and their results are analyzed. The combination of the robustness and accuracy of the HRBF model and the fast computing for the ART-AVITE model, together with the integration of both neural networks for the learning

and performance phases, gives solutions for reaching applications when the precision is a required parameter. Several configurations and sceneries have been carried out for reaching a small sphere, by means of colour-based visual algorithm and, in all the experiments, the minimum error has been found for a reduced number of movements for the robot arm.

Acknowledgements

This work was supported in part by the SENECA Fundation (Spain) PC-MC75/00078/FS/02, and the European Community, PALOMA Research Project, ref. IST 2001-33.73

References

1. M. Bertero (1986). Regularization methods for linear inverse problems. In Inverse Problems, G. Talenti (Eds). Lecture Notes in Mathematics, Vol. 1225, pp. 52–112.
2. D. D. Bullock, S. Grossberg (1989) 'VITE and FLETE: Neural modules for trajectory formation and tension control. In W. Hershberger (ed.): Volitional Action, pp. 253–297. Amsterdam, North-Holland.
3. G. Carpenter, S. Grossberg (1987) ART 2: Selft-organization of stable category recognition codes for analog input patterns. Proceedings of the IEEE First International Conference on Neural Networks. Vol. 2, pp. 727–736.
4. F.H. Guenther, D. Micci-Barreca (1997). Neural models for flexible control of redundant systems. In P. G. Morasso & V. Sanguinetti (Eds): Self-Organization, Computational Maps, and Motor Control. Elsevier, North Holland Psychology series, pp. 383–421.
5. T. Poggio, F. Girosi (1989). A theory of networks for approximation and learning. AI Memo No. 1140, Massachusetts Institute of Technology.
6. Virtual Robot Simulator. Departamento de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia. November 2002.

Design and Implementation of Force Sensor for ROBOCLIMBER

H. Montes, S. Nabulsi, and M. Armada

Control Department, Industrial Automation Institute, Madrid, Spain
hmontes@iai.csic.es

Summary. In this paper, strategies of force measurement in ROBOCLIMBER legs are presented. Diverse strategies in the placement of force sensor have been realized. Initially, finite element analyses of the mechanical structure of the walking robot legs are made. By way of the results of these analyses, specific positions are selected for strain gages placing, with the objective of take measurement of the force contact between the foot and the soil. Comparisons between different placements of the force sensor are realized and after that, two different positions of strain gages placement were selected. Afterwards, several experiments are carried out with the force sensors implemented on the two positions and the obtained results are evaluated.

1 Introduction

Initially, the industrial applications of robotics were focused in the realization of simple tasks without external sensors (pieces handling robot manipulator). However, rapidly, external sensors, like feedback element, were utilized in order to improve the performance tasks of the robots. This gave place to the development of several control strategies employing different types of sensors for the environment information feedback. For example, when the robot manipulator works in contact with the environment, then the primary objective of control is regulate the forces and torques that the manipulator exerts on the environment, while the position is controlled in those directions for which the environment does not impose restrictions [1, 2]. First investigations related to force analysis of mechanical manipulator can be found in [3]. Some other related developments to the measurement of forces on the end-effector of the manipulator can be found in [4, 5], and other work regarding force measurement at different parts of the robot (e.g. legs, feet, etc) can be found in [6, 7].

At present, there are different and new kinds of sensors that are used in automation and robotic systems making possible that the robots can realise themselves innovative complex tasks, even in environment less structured and

unknown. Sensors that can measure force, torque, or pressure usually contain an elastic member that converts the physical magnitude to a deflection or strain. In this approach strain gages have been employed, because these transducers constitute the most commonly ones used for force measurement. In these sensors, strain produces a change in electrical resistance of the strain gages pasted to an elastic element.

In the last two decades, the force based control has been intensively researched for walking robots. With different force control strategies it is possible to realise optimisation of walking robot design, avoidance of the risk the foot slippage, investigation of force distribution, smoothing of the robot's motion, enhancement of energy efficient, and identification of mechanical properties [8, 9].

In this paper are compared the strain gages arrangement located on top of each robot leg and on each foot axis of the robot in order to measure the contact forces with the ground. If the robot is on a flat surface, the forces that are measured are in the direction of the "z axis". The sensors calibrations are presented and validated through an adequate methodology. With the sensor validated, it is possible to make several types of force control.

2 Force Sensor Implementation in ROBOClIMBER

A measurement is the experimental determination of the magnitude of a physical quantity upon comparing it with the corresponding patron unit of measuring. However, the majority of the measurements are carried out indirectly exploiting the physical effects in adequate devices of measuring: transducers or sensors. Often, the physical quantity to be measured is converted by a sensor to an electric signal that could be amplified and registered.

In this approach the physical measured quantity is the force by means of deformations of element elastic sensor. For this reason are very common use strain gages, implemented of satisfactory way, for the measurement of the forces in a mechanism subjected to any type of strain. The force sensor used with strain gages has the configuration of Wheatstone full bridge. With this configuration the force sensor reach the highest sensibility when the strain gages are used. The Wheatstone bridge convert the variation of the strain gages resistance, subjected to tension or compression, to output voltage, of such form that the pair of strain gages that are opposed varies their resistance of mode inverse that the other pair of strain gages, in the Fig. 1(a) is shown the configuration of complete bridge. Figure 1(b) show the deformation of each pair of strain gages subject to some strain (tension or compression depend upon the configuration).

ROBOCLIMBER is the walking robot used to realise the experiments with force sensors. ROBOCLIMBER [10–12] is a quadruped walking and climbing robot whose development is funded by the EC under a Growth/Craft RTD

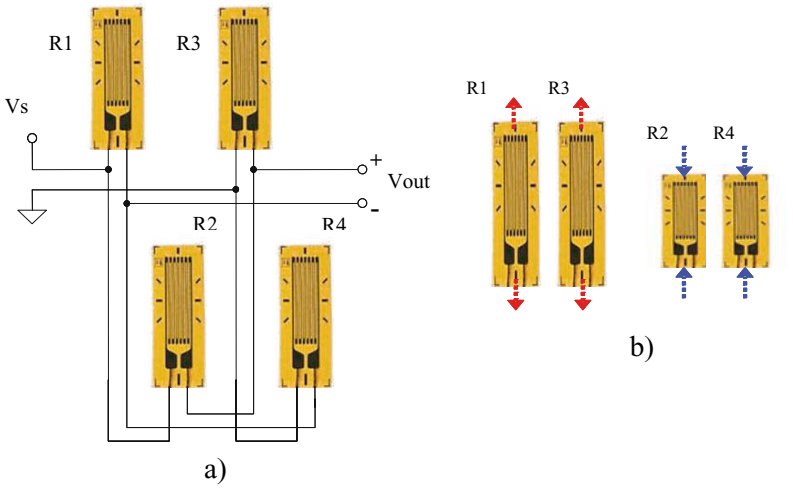


Fig. 1. (**a**) Wheatstone full bridge with strain gages; and (**b**) deformation of strain gages subject to tension or compression

project, where the objective is to develop a tele-operated service robotic system that will perform consolidation and monitoring tasks on rocky slopes. ROBOCLIMBER must work by climbing rough mountain slopes with inclination from 30° to almost 90°. Under these conditions, the robot has to be hold by steel ropes and helped to be pulled up from the top of the mountain.

ROBOCLIMBER has cylindrical leg configuration [11, 12] and it is hydraulically powered and controlled by PID system based on the IAI-CSIC developed control electronic board. Each leg has 3 DOF, one rotation and two prismatic joints. Figure 2 shows a photo of ROBOCLIMBER and the kinematics parameter of the plant view.

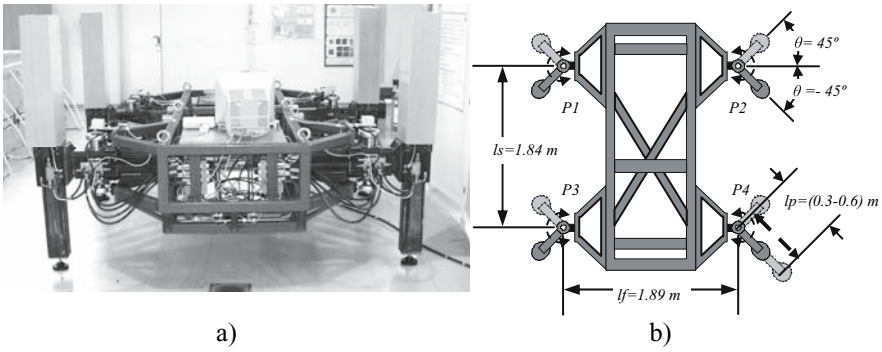


Fig. 2. (**a**) ROBOCLIMBER at IAI-CSIC lab; and (**b**) ROBOCLIMBER kinematics parameter (*top view*)

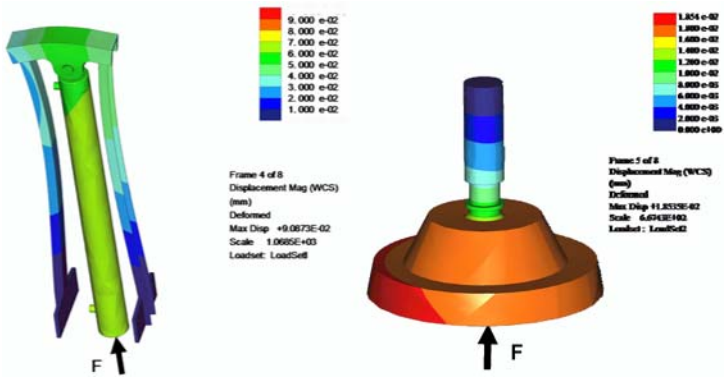


Fig. 3. Finite element analysis in ROBOCLIMBER leg, (**a**) on the leg structure; and (**b**) on the foot

In order to prepare ROBOCLIMBER with force sensing capabilities a previous element analysis (FEA) with Pro/Mechanica modulus of Pro-Engineer® software was realised. The strain was calculated when several loads were applied over the robot foot. Figure 3 shows this analysis with 7500 N of force.

In ROBOCLIMBER legs several configurations of force sensor with strain gages have been implemented. The objective was comparing them for to select the arrangement with minor function of cost. In this paper, two different placement of force sensor are shown. The first configuration was implemented on the top of the robot leg and de second was applied on the foot bar. So, when the robot legs contact ground the force sensor in each leg is able to detect contact and to measure force, so it is possible, for example, to calculate the robot centre of pressure in real time. With this approach control strategies aimed to keep the centre of pressure inside support polygon of the robot can be implemented.

Force sensors implemented are shown in Fig. 4. Strain gages are placed like a Wheatstone full bridge (see Fig. 1(a)), only in the first configuration (strain gages on the top leg) the real functioning is like half bridge because strain gages R2 and R4 are of reference. This strain gages (reference) are placed on unstrained zone and active strain gages (R1 and R3) are pasted on most strained area obtained by previous finite element analysis.

3 Calibration Procedure and Comparison of Force Sensor

The force sensors were calibrated with a reference instrument which characteristic are shown in Table 1. At IAI-CSIC lab several calibrations test were realised for each robot leg in different working conditions.

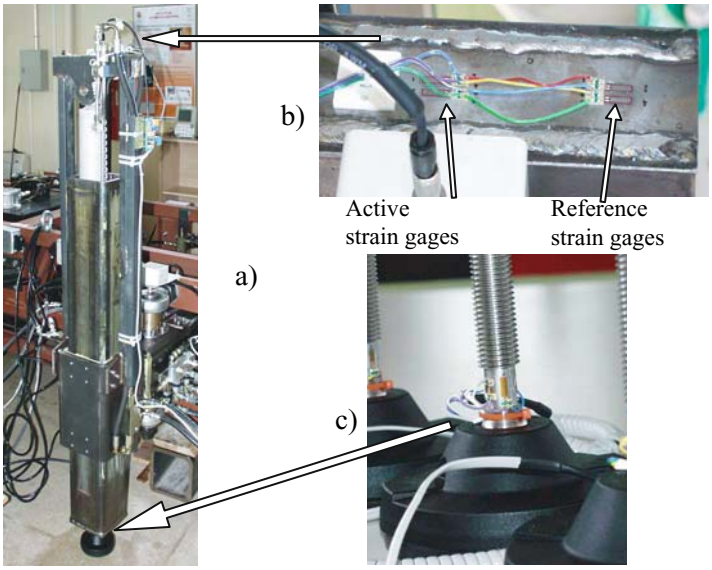


Fig. 4. (**a**) ROBOCLIMBER leg; (**b**) force sensor mounted on the top of the leg; and (**c**) strain gages on the foot bar like classical pillar load cell

Table 1. Reference instrument charasteristic

Maximal Measurement	Scale	Standard Deviation	Uncertainty
3000 Kg.	1 Kg.	0.1722 Kg.	0.5992 Kg.

The calibration process consists in the calculation of the calibration matrix that relates the voltage measured in each one of the force sensors placed in each leg of the robot and the measured forces by the reference instrument. This function expressed like: $\mathbf{F} = (1/\nu_s)\mathbf{K}\mathbf{v} + \mathbf{b}$ where, $\mathbf{K}$ is the calibration matrix, $\mathbf{v}$ is the vector of the difference of voltage registered by the sensor, ν_s is the power supply of the strain gages bridge and $\mathbf{b}$ is the vector of constants. For the sensor located on the superior part of the mechanic structure of the leg, this equation is considered as:

$$\begin{bmatrix} F1 \\ F2 \\ F3 \\ F4 \end{bmatrix} = \frac{1}{\nu_s} \begin{bmatrix} 87475 & 0 & 0 & 0 \\ 0 & 79598 & 0 & 0 \\ 0 & 0 & 82335 & 0 \\ 0 & 0 & 0 & 151113 \end{bmatrix} \begin{bmatrix} \nu_{def1} - \nu_{ud1} \\ \nu_{def2} - \nu_{ud2} \\ \nu_{def3} - \nu_{ud3} \\ \nu_{def4} - \nu_{ud4} \end{bmatrix} + \begin{bmatrix} 3413 \\ 4344 \\ 4803 \\ 8410 \end{bmatrix} \tag{1}$$

For the sensor placed on the foot bar, the force equation is considered as:

$$\begin{bmatrix} F1 \\ F2 \\ F3 \\ F4 \end{bmatrix} = \frac{1}{\nu_s} \begin{bmatrix} 28651 & 0 & 0 & 0 \\ 0 & 20868 & 0 & 0 \\ 0 & 0 & 23978 & 0 \\ 0 & 0 & 0 & 20463 \end{bmatrix} \begin{bmatrix} \nu_{def1} - \nu_{ud1} \\ \nu_{def2} - \nu_{ud2} \\ \nu_{def3} - \nu_{ud3} \\ \nu_{def4} - \nu_{ud4} \end{bmatrix} + \begin{bmatrix} 16.1 \\ 30 \\ -33.1 \\ 87.7 \end{bmatrix} \quad (2)$$

Where, ν_{def} is the measured voltage when strain gages are strained and ν_{ud} it is the voltage when the strain gages are unstrained, that is to say, there is not force on the robot leg.

It is possible to notice that each strain gages Wheatstone bridge is especially sensitive to some determined forces, as it is a function of their placement in the mechanic structure and material properties.

Some test calibration results of each force sensor are shown in Fig. 5. In this figure the two configurations of force sensor implemented on ROBOCLIMBER leg are compared. Each sensor shows behaviour, approximately, lineal and the calibration results present a small error when the forces are distributed, almost evenly, on the four legs of the robot. When the robot posture is unusual, the forces distribution presents a particular behaviour and the error in the force measurement is increased. Also, it is possible to appreciate that the sensibility of the force sensor placed on the foot bar is greater than the force

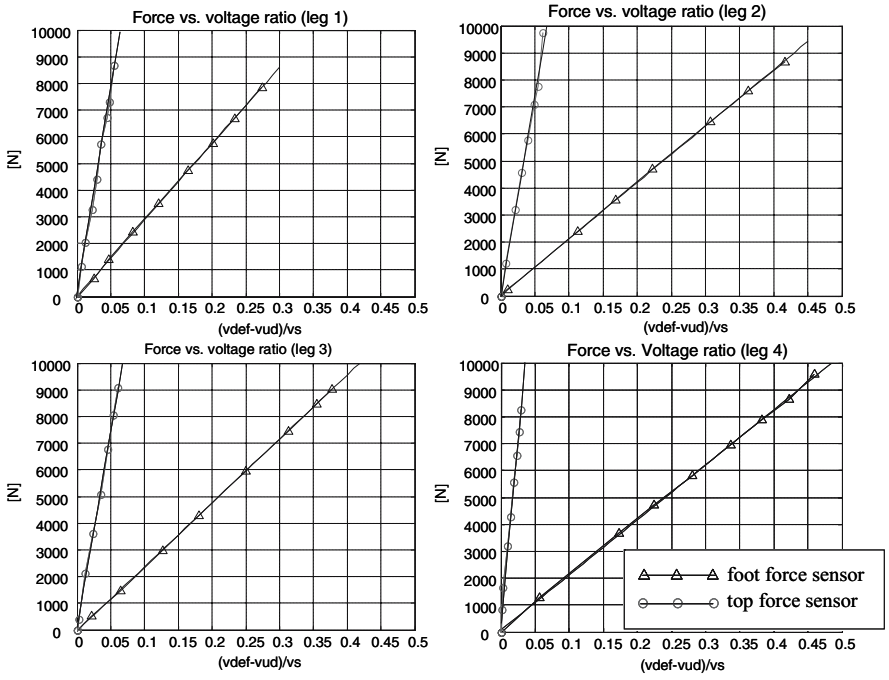


Fig. 5. Calibration results and comparison between two different configurations of force sensor

sensor placed on the top part of the leg structure, where, the voltage ratio, in some leg, is until 10 times bigger. Afterwards of these comparisons several experimentations were carried out to validate the force sensors.

4 Experimental Results

Force sensor was implemented on ROBOCLIMBER. Several experiments carried out in where taken measured of force in each leg of robot over top leg structure and all feet's bars. Some experimental results are shown in the Fig. 6, where it is possible see the force in each leg and the resultant force in the two different configurations.

Before to realise this experiment the robot weight was measured (19355 N) with the reference instrument used for the sensor calibration. First, the robot got up to 0.5 m on the ground and the distribution of the forces was observed. With the first implementation (sensor on the top leg) the robot stayed up for 50 sec, approximately (see Fig. 6(a)). When the force was stabilized the measurement of total force was 17700 N, occurring 8.5% of error in the measurement. It is possible to see transient phenomenon when the robot contact the ground and when the robot delivery the ground.

With the second implementation, also the robot got up to 0.5 m on the ground and it stayed out for 150 sec, approximately (see Fig. 6(b)). In this experiment, in addition, one object of 834 N is moved on top of the robot body, starting in $t \approx 80$ sec until the $t \approx 180$ sec. With this sensor implementation the average measurement of force was 19320 N and 20110 N for both cases. The error in the two average measurement of force is less than $\pm 0.5\%$, meaning the high sensibility of the sensor. The transient phenomenon when the robot starting goes up and when the robot legs leaved the contact ground was a little appreciable.

5 Conclusions

Force sensors are implemented in quadruped walking robot in two different configurations. FEA gave valuable information on the best target points for force measurement, and consequently two strategies were carried out. The force sensors were applied to the upper part of leg and on each foot bar of ROBOCLIMBER. These sensors were calibrated, conveniently, and tested, evaluated, and compared. The sensor pasted on the feet bars shown the best results with less $\pm 0.5\%$ of error in comparison with the sensor placed on top of the leg with $\pm 8.5\%$ of error, approximately. With this force sensor implementation the robot can detect the terrain contact and to measure contact forces in real time with high reliability.

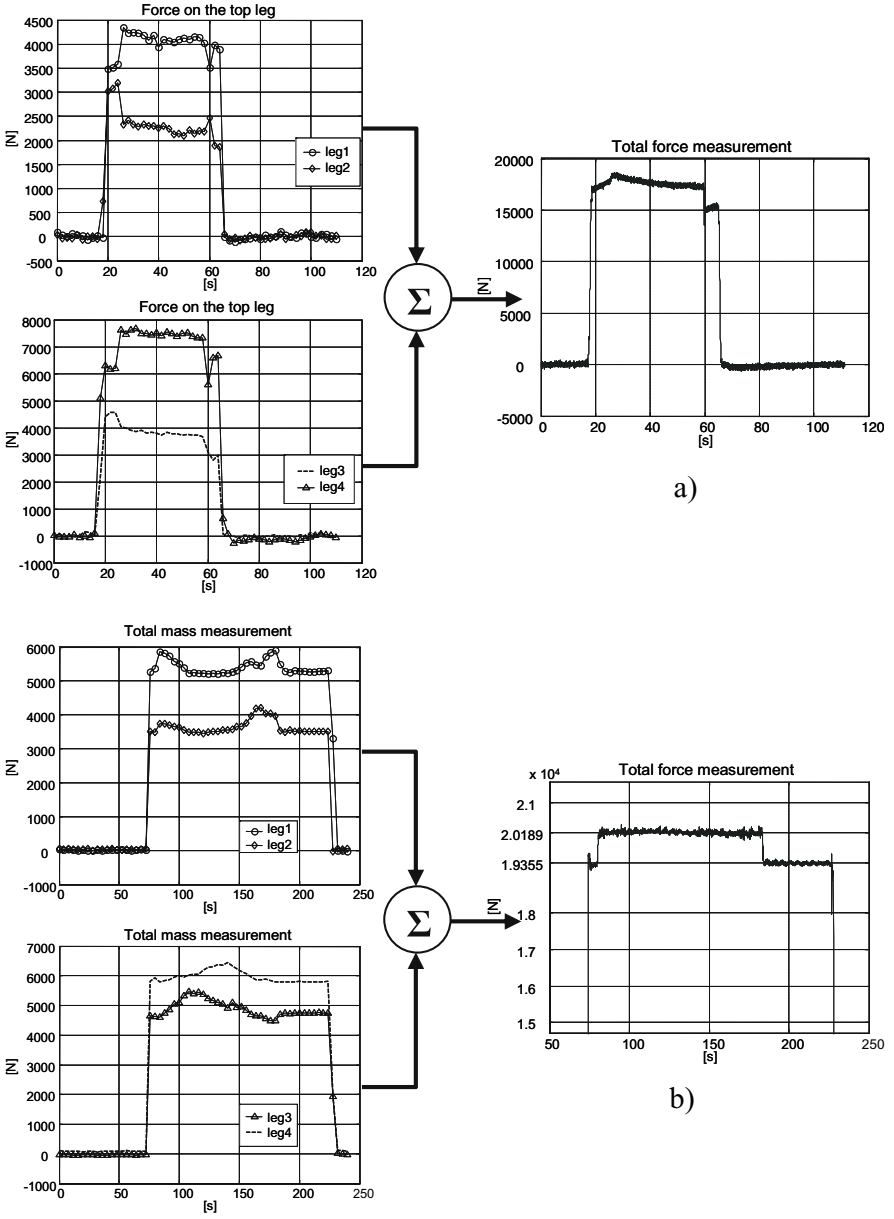


Fig. 6. Measurements of force in each leg and force resultant, (**a**) on the top of robot legs; and (**b**) on the feet bars

Acknowledgments

ROBOCLIMBER project is funded by EC under Contract N°: G1ST-CT-2002-50160. The project partnership is as follows: ICOP S.p.a., Space Applications Services (SAS), Otto Natter Prazisionenmechanik GmbH, Comacchio SRL, Te.Ve. Sas di Zanini Roberto & Co. (TEVE), MACLYSA, D'Appolonia S.p.a., University of Genova-PMAR Laboratory, and CSIC-IAI.

EC funded CLAWAR Thematic Network has been very sipportive of all author's activities in the field of climbing and walking robots.

References

1. Whitney D.E. (1987) Historical perspective and state of the art in robot force control. The International journal of Robotic Research 6:3–14
2. Hogan N. (1985) Impedance control: an approach to manipulation. J Dynamic Systems, Measurement, and Control 107:1–24
3. Salisbury J.K., Roth B. (1983) Kinematic and force analysis of articulated mechanical hands. J of Mechanisms, Transmissions and Automation Design 105:35–41
4. Grieco J.C., Armada M., Fernandez G., Gonzalez de Santos P. (1994) A review on force control of robot manipulators Studies. J Informatics and Control 3:241–252
5. Gorinevsky D.M., Formalsky A.M., Schneider A. Yu (1997) Force Control and Robotics Systems. CRC Press LLC, Boca Raton, FL
6. Siciliano B., Villani L. (1999) Robot Force Control, Kluwer Academy Publishers, Norwell, Massachusetts
7. Spong W., Vidyasagar M. (1989) Robot Dynamics and Control. John Wiley & Sons, Singapore
8. Galvez J.A., Gonzalez de Santos P., Armada M. (1998) A Force Controlled Robot for Agile Walking on Rough Terrain. In: ICV'98, 23–24 March, Sevilla, Spain
9. Kumar V., Waldron K.J. (1990) Force Distribution in Walking Vehicles. J Mechanical Design 112:90–99
10. Nabulsi S., Armada M. (2004) Climbing strategies for remote maneuverability of ROBOCLIMBER. In: 35th International Symposium on Robotics, 23–26 March, Paris-Nord Villepinte, France
11. Armada M., Molfino R.M. (2002) Improving Working Conditions and Safety for Landslide Consolidation and Monitoring. In: Workshop on the role of CLAWAR in education, training, working conditions and safety, on CD-Rom, Madrid, Spain
12. Anthoine P., Armada M., Carosio S., Comacchio P., Cepolina F., González P., Klopf T., Martin F., Michelini R.C, Molfino R.M., Nabulsi S., Razzoli R.P., Rizzi E., Steinicke L., Zannini R., Zoppi M. (2003) ROBOCLIMBER. In: International Workshop on Advances in Service Robotics, 13–15 March, Bardolino, Italy

Detecting Zero-Moment Point in Legged Robot

H. Montes, S. Nabulsi, and M. Armada

Control Department, Industrial Automation Institute, Madrid, Spain
hmontes@iai.csic.es

Summary. The study of Zero-moment point (ZMP) in ROBOCLIMBER (quadruped walking and climbing robot) was realized and calculated in real time during statically postural and dynamically balanced gait. Force sensor with strain gages was implemented on each foot bar of the robot in order to measure the ground reaction forces used for calculate the ZMP and to realize futures force control strategies. With the statically postural the ZMP was localized in order to know the point where the reaction force of the ground acts and so to know the distance between the geometric centre and the statically ZMP. When the robot realized one gait the ZMP was analyzed in order to identify if the ZMP was inside of support polygon. With this analysis it is possible to realize stables gaits with compliance movements.

1 Introduction

The concept of center of pressure is originated in the field of the mechanics of fluids, but is frequently utilized in the study of gait and postural stability of robots. The CoP is defined as the point on the ground where resultant of the ground-reaction force acts [1].

Some authors use the concept of zero-moment point (of the acronyms ZMP) for the study of postural balance in biped robot [2–6] and other authors compare the concept of CoP and ZMP [1–7].

Reference [3] define the ZMP as that point where the vertical reaction force intersects the ground. Other definitions of ZMP can be found in [4–6]. With this four definition of ZMP [1] state that the concept of ZMP is identical to the CoP and that the CoP is better known as the ZMP in the robotics literature. Reference [7] suggested difference between ZMP and CoP. He affirms that the pressure between the foot and the ground can always be replaced by a force acting at the CoP. If this force balances all the forces that acting on the mechanism during the motion its acting point is ZMP.

Consequently, when occur a dynamically balanced gait, CoP and ZMP coincide. When the gait is not dynamically balanced, ZMP does not exist and the robot falls down about the foot edge, and the ground reaction force acting

point is CoP while the point where $M_x = 0$ and $M_y = 0$ is outside of support polygon. This point is called fictitious ZMP (FZMP) [7].

The concept of ZMP is mostly used in gait planning for biped robot and biped robot control as a criterion of postural stability.

The concept of ZMP is mostly used in gait planning and control as a criterion of postural stability for biped robot. In this approach the concept of ZMP in a walking robot of four legs is studied. The ZMP is calculated in real time during standing up of the robot and while the robot remains statically stable. With this, the ZMP is known when the four legs of the robot form a quadrilateral and it will be compared with the geometric centre of the robot. When the robot stands up an object is moved on the robot and the variation of the ZMP is registered, therefore, the ZMP behaviour is observed when a perturbation occurs. Finally, the ZMP is recorded when the robot realizes two-phase discontinuous gait. By means of this study will be carried out subsequently, control in the planning of trajectory, improving the gait of the robot. The locomotion velocity will be increased take into account the robot stability.

2 ROBOCLIMBER General Configuration

ROBOCLIMBER [8–11] is a quadruped walking and climbing robot whose development is funded by the EC under a Growth/Craft RTD project, where the purpose is to develop a tele-operated service robotic system that will carry out consolidation and monitoring tasks on rocky slopes. This robot must work by climbing irregular mountain slopes with inclination ranging from 30° to approximately 90°. Under these conditions, ROBOCLIMBER has to be hold by steel ropes and helped to be pulled up from the top of the mountain by a truck.

This robot is made with a mechanical structure with total mass of 1973 kg. (in the first configuration), mostly made of steel. Cylindrical leg configuration has been preferred [9–11], and it is hydraulically powered and controlled by a PID system based on the IAI-CSIC developed control electronic boards. Each robot leg has 3 DOF, one rotation in the transversal plane; the second is a prismatic joint in the transversal plane, too; and other prismatic joint in the orthogonal axis to the robot body. The system is designed to overcome obstacles greater than 500 mm. and the robot must to support the necessary equipment on board to consolidate and to monitor the slopes mountain. Figure 1 shows ROBOCLIMBER the initial configuration.

The mechanical configuration of ROBOCLIMBER is shown in Fig. 2. In order to know the ZMP equations for this robot, therefore, it is necessary to know its kinematics parameter. Figure 2 shows the transversal plane the robot structure.

In this mechanical configuration of the robot, it is possible to see the four rotational joints with maximum movement of ±45° and four radial prismatic

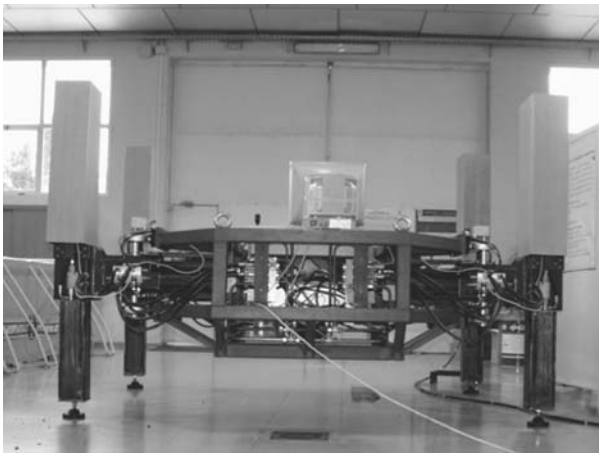


Fig. 1. ROBOCLIMBER at IAI-CSIC lab

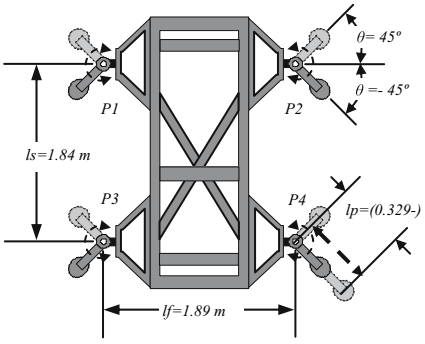


Fig. 2. ROBOCLIMBER kinematics parameter (transversal plane)

joint with 0.3 m of displacement. The fixed distance in the lateral plane is $l_f = 1.89\,\text{m}$ and in the sagittal plane is $l_s = 1.84\,\text{m}$. The prismatic joints orthogonal to the robot body have a lineal displacement of 0.7 m.

With relation to the kinematics parameter of ROBOCLIMBER, it is possible to calculate the ZMP of the robot when this is dynamically balanced, as:

In the sagittal plane,

$$ZMP_s = \frac{d_1\mathbf{a}(\mathbf{Kv}+\mathbf{b}) + \mathbf{cA}(\mathbf{Kv}+\mathbf{b})}{\mathbf{c}(\mathbf{Kv}+\mathbf{b})} \tag{1}$$

In the lateral plane,

$$ZMP_l = \frac{d_2\mathbf{d}(\mathbf{Kv}+\mathbf{b}) + \mathbf{dB}(\mathbf{Kv}+\mathbf{b})}{\mathbf{c}(\mathbf{Kv}+\mathbf{b})} \tag{2}$$

where,

$$d_1 = \frac{l_S}{2}; \quad d_2 = \frac{l_f}{2};$$

$$\mathbf{a} = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}; \quad \mathbf{c} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}; \quad \mathbf{d} = \begin{bmatrix} -1 & 1 & -1 & 1 \end{bmatrix};$$

$$\mathbf{b} = \begin{bmatrix} 16.1 \\ 30 \\ -33.1 \\ 87.7 \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} lp_1 S_1 & 0 & 0 & 0 \\ 0 & lp_2 S_2 & 0 & 0 \\ 0 & 0 & lp_3 S_3 & 0 \\ 0 & 0 & 0 & lp_4 S_4 \end{bmatrix};$$

$$\mathbf{v} = \frac{1}{\nu_s} \begin{bmatrix} \nu_{def1} - \nu_{do1} \\ \nu_{def2} - \nu_{do2} \\ \nu_{def3} - \nu_{do3} \\ \nu_{def4} - \nu_{do4} \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} lp_1 C_1 & 0 & 0 & 0 \\ 0 & lp_2 C_2 & 0 & 0 \\ 0 & 0 & lp_3 C_3 & 0 \\ 0 & 0 & 0 & lp_4 C_4 \end{bmatrix};$$

$$\mathbf{K} = \begin{bmatrix} 28651 & 0 & 0 & 0 \\ 0 & 20868 & 0 & 0 \\ 0 & 0 & 23978 & 0 \\ 0 & 0 & 0 & 20463 \end{bmatrix}$$

It is possible use the (1) and (2) when support polygon is formed by four legs or by three legs and with different position of the rotational and prismatic joints.

3 Sensing ZMP

The ZMP of ROBOCLIMBER was calculated in statically stable postural when the robot legs have quadrilateral shape. The parallelogram dimension is 2.22 m by 1.84 m. By means of these conditions the ZMP was – 0.08 m in sagittal plane and –0.04 m in the lateral plane, considering the geometric centre in the middle of the robot. This one result indicates that the ZMP and the geometric centre are very near and is verified that the mass of the robot is approximately uniformly distributed.

For sensing the variations of ZMP in statically postural but put under perturbations one object with weight of 834 N was moved over the robot body. This object weight represents the 4.3% of the robot weight; therefore, it is a small disturbance that will indicate the sensitivity of the measurement system to detect the displacement of the ZMP. The resultant force measurement was 20110 N (with error of −0.3%, approximately), this is the normal force acting in the ZMP when the object was onto the robot (see Fig. 3).

The object walked contrary to the clockwise movement from the leg one to the leg four, and the ZMP registered realize the same movement. It is possible to observe a movement of the sagittal ZMP of 0.117 m and 0.0768 m in the lateral one.

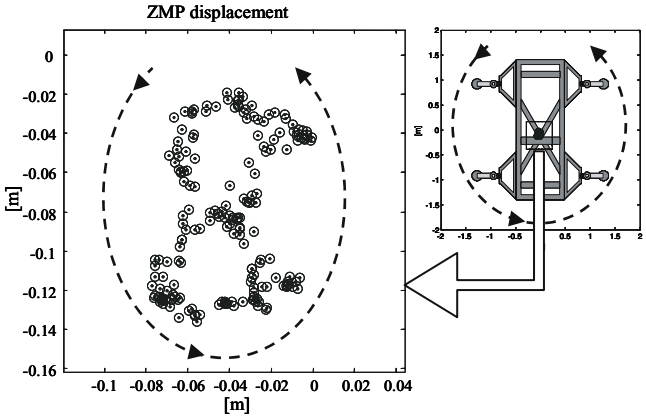


Fig. 3. ZMP displacement when one object walking onto the robot

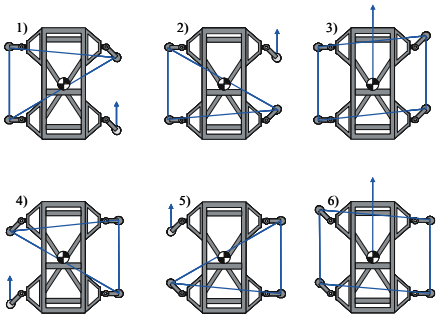


Fig. 4. Two-phase discontinuous gait in ROBOCLIMBER

4 Experimental Result

Several experiments were carried out in order to detect the ZMP when the ROBOCLIMBER realize a dynamically balanced gait. In this experiment, the robot, perform two-phase discontinuous gait [8]; the locomotion on a rigid, flat, and horizontal surface is assumed. It is supposed that the vertical axis is "z" axis and that this, as well, is orthogonal to the robot body. Figure 4 shows a scheme of the two-phase discontinuous gait of ROBOCLIMBER.

In this one periodic gait measurements of the positions of the 12 DOF and vertical leg force were recorded in real time (see Fig. 5). In this experiment the weight of the robot was augmented to 21500 N (with relation to the preliminary experiment described in Sect. 3) because a human operator, machine power supply, and control equipment were added.

In one step of ROBOCLIMBER the ZMP in the sagittal plane and the lateral one was detected by using the (1) and (2), respectively (see Fig. 6). This result demonstrates that the ZMP is inside of the support polygon (if is not inside the support polygon it can not be called ZMP), but is very near

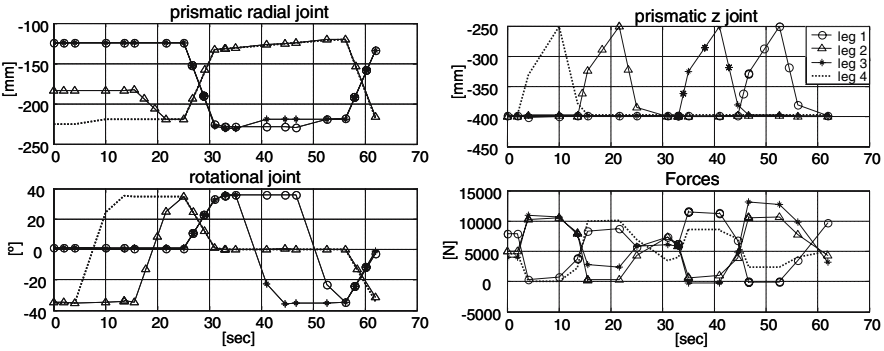


Fig. 5. Measurements in one step of ROBOCLIMBER

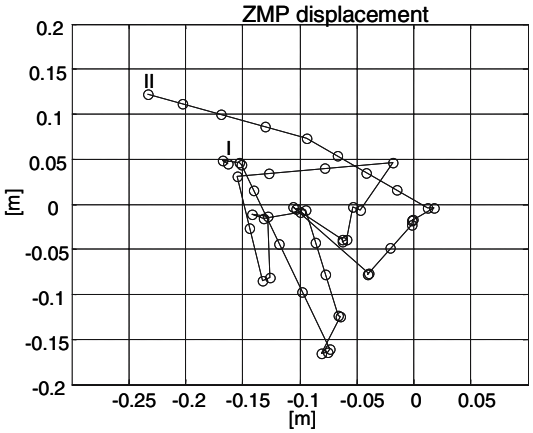


Fig. 6. Detecting ZMP in one step of ROBOCLIMBER

of the polygon boundary. The average resultant force measurement, in this experiment, was 22400 N and it acts in each ZMP during the robot step.

5 Conclusions

The ZMP was detected when the legged robot was onto flat surface. Two different kinds of experiments were realized; the first one, when the robot was stands and statically stables; the second, when the robot realize one step of two-phase discontinuous gait.

In the first case ZMP measurement is near to the geometric centre of the robot, consequently the robot mass is well distributed. In the second case, with dynamically balanced gait the ZMP was detected, but some of these points were near to the support polygon boundary. With analyse this results

it is possible to improve the gait in order to realize more stables gaits with compliance movements in the legs and the body.

Acknowledgment

ROBOCLIMBER project is funded by EC under Contract N°: G1ST-CT-2002-50160. The project partnership is as follows: ICOP S.p.a., Space Applications Services (SAS), Otto Natter Prazisionenmechanik GmbH, Comacchio SRL, Te.Ve. Sas di Zanini Roberto & Co. (TEVE), MACLYSA, D'Appolonia S.p.a., University of Genova-PMAR Laboratory, and CSIC-IAI.

EC funded CLAWAR Thematic Network has been very sipportive of all author's activities in the field of climbing and walking robots.

References

1. Goswami A. (1999) Postural stability of biped robots and the Foot-Rotation Indicator (FRI) point. The International Journal of Robotics Research 18(6):523–533
2. Vukobratovic M., Borovac B., Surla D., Stokic D. (1990) Scientific Fundamentals of Robotics 7. Biped locomotion: Dynamics Stability, control, and Application. Springer-Verlag, New York
3. Hemami H., Golliday C.L. (1977) The inverted pendulum and biped stability. J Math. Biosci. 34(1-2):95–110
4. Takanishi A., Ishida M., Yamazaki Y., Kato I. (1985) The realization of dynamic walking by the biped robot WL-10RD. In: Proc. of the Intl. Conf. on Adv. Robot, 459–466, Tokyo
5. Arakawa T., Fukuda T. (1997) Natural motion generation of a biped locomotion robot using the hierarchical trajectory generation method consisting of GA, EP layers. In: Proc. of the IEEE Conf. on Robot. and Automat. 1:211–216, Albuquerque, NM
6. Hirai K., Hirose M., Haikawa Y., Takenaka T. (1998) The development of the Honda humanoid robot. In: Proc. of the IEEE Conf. on Robot. and Automat. Leuvin, Belgium
7. Vukobratović M., Borovac B. (2004) Zero-Moment point – Thirty five years of its life. International Journal of Humanoid Robotics. 1:157–173
8. Nabulsi S., Armada M. (2004) Climbing strategies for remote maneuverability of ROBOCLIMBER. In: 35th International Symposium on Robotics, 23–26 March, Paris-Nord Villepinte, France
9. Armada M., Molfino R.M. (2002) Improving Working Conditions and Safety for Landslide Consolidation and Monitoring. In: Workshop on the role of CLAWAR in education, training, working conditions and safety, on CD-Rom, Madrid, Spain
10. Anthoine P., Armada M., Carosio S., Comacchio P., Cepolina F., González P., Klopf T., Martin F., Michelini R.C., Molfino R.M., Nabulsi S., Razzoli R.P., Rizzi E., Steinicke L., Zannini R., Zoppi M. (2003) ROBOCLIMBER. In: International Workshop on Advances in Service Robotics, 13–15 March, Bardolino, Italy

11. Acaccia G., Bruzzone L.E., Michelini R.C., Molfino R.M., Razzoli R.P. (2000) A tethered climbing robot for firming up high-steepness rocky walls. In: Proc. of the 6th IAS Intl. Conf. on Intelligent Autonomous Systems, 25–27 July, Venice, Italy

Vision Feedback in Control of a Group of Mobile Robots

Piotr Dutkiewicz and Marcin Kiełczewski

Institute of Control and Systems Engineering, Poznań University of Technology
ul. Piotrowo 3A, 60-965 Poznań, Poland
piotr.dutkiewicz@put.poznan.pl
marcin.kielczewski@put.poznan.pl

Abstract. This article is devoted to the vision system, which is an integral part of control system for mobile robots soccer team. The paper describes an experimental test bed used for real-time image acquisition, processing, and recognition of objects placed on the 2D surface. Description of specific features of objects being recognized, which are controlled and opponent team robots as well as a ball has been also included. Filtering methods used to improve quality of the image received from camera have been presented. The paper also shows a new heuristic algorithm, invented for objects recognition on the scene, playground in our case. The algorithm has been implemented, experimentally verified and quality of measurement has been estimated.

1 Introduction

Research results often lead to situation, where only narrow group of specialists is able to take advantage of the development outcomes. In recent years a concept arose to verify experimentally problems existing in complex systems, and to make them understandable for a group of people as broad as possible. Based on this idea, soccer competition between mobile robot teams were carried out [14]. Such training ground gives great opportunity for presentation complex problems in simple way, such as recognition of unknown environment, action planning and control of a mobile robot system [5, 6].

One of the important components of this is a vision system [3, 11], used for object recognition and measurement of position and orientation of objects placed on the playground. There are the following objects on the playground: three mobile robots of one team, three robots of opponent team and the ball. Measurement data, coming from vision system, are used as input to strategy and control algorithms, which produce set of commands sent to robots of the controlled team. Development of the visual feedback is the main goal of research presented in this paper, which is an extension of achievements

presented in [7]. Features of objects, vital for correct measurement, are examined. Also image filtering methods were selected as necessary for improving the quality of images received from camera. The main task was to develop recognition algorithm for specific objects on the scene. Implementation of extraction algorithm of specific features should satisfy criterion of speed, high reliability of operation and easy characterization of features for objects being recognized. Invented heuristic algorithm for objects recognition is based on the compromise between using widely known methods of filtering and image processing [2, 9, 10], and necessity of real time processing [1, 4].

2 Description of the Measurement Setup

A detailed description of the MiRoSoT (Micro-Robot Soccer Tournament) environment can be found in [11, 14]. According to the MiRoSoT regulations, CCD color camera has to be fixed above the playground. The video stream, coming from the camera, is then collected by a specialized PC card which enables image displaying on the user console and image data processing simultaneously.

The main part of the vision system is the framegrabber card Matrox Corona, achieving 25 frames per second (fps) for 768×576 image size with full soft real-time display and 24 bit color. As an input signal composite video, S-Video or RGB components signals, transmitted in PAL or NTSC standards, may be used with the framegrabber card. Additionally, the card has 24-bit digital input RS-422/LVDS. The library package MIL-Lite, supplied with Matrox Corona card, gives flexible and easy-to-use data processing and displaying framework available under Windows NT/2000 OS. Detailed description of the MIL library with application examples and Matrox Corona card can be found in [12, 13].

Very important part of the vision system, shown in Fig. 1, is a CCD camera. In our application Sony DCR-TRV 900E color camera was used. The

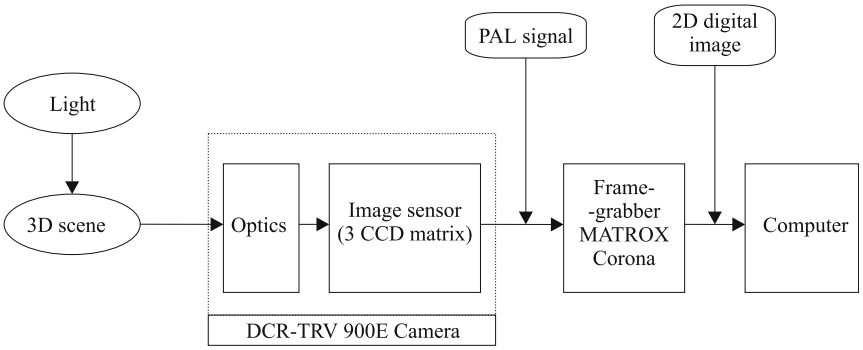


Fig. 1. General diagram of vision system

camera is equipped with three CCD matrices, each for individual RGB components. Analog output formats available in the camera are composite and S-Video signals. Additionally, digital *i.Link* output, transferring compressed image in DV standard, is also accessible. During vision system development Microsoft Visual C^{++} 6.0 programming tool was used.

3 Features of Recognized Objects

In accordance with the MiRoSoT rules [14] two teams of robots participate in the game play. They play an orange golf ball, which is the easiest object to find on the playground. On image it looks like a color circular patch.

During the game a 8×8 cm marker with two color patches is placed on each player. One patch indicates team color (common for one team) and the other patch recognizes individual robot color. Team patch is a uniform spot square with minimum dimensions 3.5×3.5 cm. Yellow and blue colors are allowed. Individual robot patch identifies each robot inside team. The goal of the vision system is patch recognition and classification using color and associate individual and team colors of patches in pairs. After this, orientation and position of robots on playground are calculated using determined image coordinate of recognized objects.

The marker system of our robot is presented in Fig. 2. Recognition of the robot (marked by color patches) depends on correct patch classification (according to defined colors) and finding two sharp corners of rhombus patch composed of individual and team patches. Based on determined points, position and orientation of robot are calculated.

4 Algorithm of Determining Position and Orientation of the Robot

In order to accelerate image processing algorithms, a technique based on a decreased image window [8] is often used. Unfortunately, it gives not always

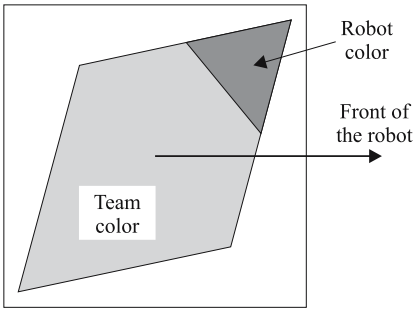


Fig. 2. Patch marker of our robot

expected results, because in extreme case (after losing object) it is necessary to increase processed window to the whole image. Moreover, object features extraction in color is more complicated comparing to black and white images.

It is assumed in this concept of objects recognition, that image is checked with quite large step in x and y directions in order to find a point inside robot patch marker or ball blob. Search step has to be selected to hit all regions of interest on image represented by defined colors and with the smallest amount of pixels to be checked. In this way significant acceleration of image processing is obtained. But with large search step it may occur, that a group of pixels representing a specific object will not be found and object cannot be recognized. To increase probability of hitting all regions representing searched objects, double image searching with the same big step has been introduced. Checked pixels between first and second searching are shifted by half fixed step. Second image checking is activated only if the first pass fails.

4.1 Color Segmentation

During image searching, detection of pixels representing objects is based on colors. Due to a noise on image and to make easier specification and allocation colors to objects, segmentation of available color space is used. This solution facilitates color classification of any object from available color palette and increases noise robustness. In this method, RGB color representation model [2] is used. Color segmentation form 24-bit color palette (R:G:B components have respectively 8:8:8 bits) is done by an indirect addressing algorithm. As a result of that, a three-dimensional version of a look-up-table technique is obtained. Data necessary to realize this are located in three arrays, filled with suitable values during initialization. First, one-dimension array is used for scaling color components. In this array, values of a function transforming discrete RGB components from 0–255 range to discrete values from 0–8 range (distribution of these values can be any function; also nonlinear). Indexing this table by the color component value gives new component value from narrowed range. This operation gives three new values of color components – they are three indices addressing elements in the second color array, which is three-dimensional (one dimension represents one color component). Elements in the second array are indices of colors, which are defined in the third table. Each element of this table is a structure consisting of few fields. These fields (connected with color and objects properties) contain information about color component value, color brightness, object identifier, and logical variable. The identifier indicates number of the object, to which has been assigned this color element, and logical variable defines, if this color has been assigned to any other object. Colors defined in this array are arranged from black to white, according to increasing color component value. In this way, using described arrays, it is easy to check to which object belongs examined point, described by RGB color components. As object a color set from a defined color palette (see Fig. 3(b)) is understood. It is assumed, that there are six objects on the image:

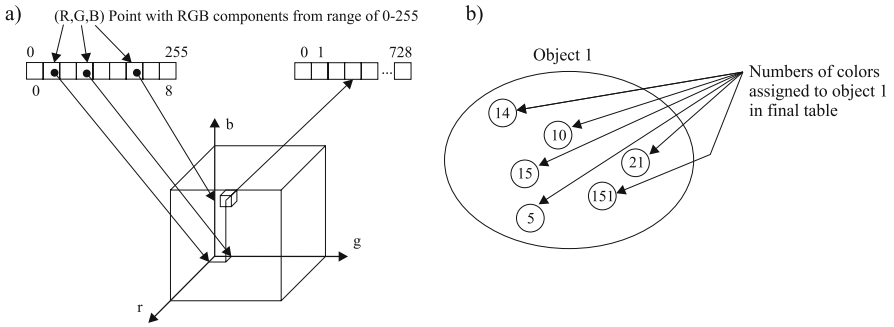


Fig. 3. Method of color space segmentation (**a**) and object defining (**b**)

ball, patches with color of one team, opponent team and three individual color patches of players. Described method of color space segmentation is shown in Fig. 3(a).

4.2 Image Preprocessing Methods

During image searching for pixel of color that is assigned to objects, there are used two image preprocessing methods. First, contextless method (also called single-point filtering), is characterized by operations done only on single image pixels, without considering pixel neighborhood. Second preprocessing method is a logical context filtering, which is used to eliminate noise in the form of isolated pixels with object colors, which have not connection with neighbouring pixels. To check connection between examined pixel and neighbourhood pixels, convolution mask of 3×3 pixels size and center point of mask put on checked pixel is used. Logical filtering is based on checking, if all selected pixels under mask have the same color as the checked point. If so, that pixel is accepted as a point belonging to the object, which has assigned this point color.

In order to obtain independence from color interference, instead of checking team color for edges detection, information about pixels luminance was used. Luminance is transmitted practically lossless in PAL standard, contrary to color information. Pixels luminance of patch on image is similar, while color can change in wide range. Edge detection using luminance is possible, because light color patches are placed on dark background. If luminance of a particular pixel was above the threshold value and for the following checked pixel it was below the threshold, then current pixel was accepted as a border point. Because pixels in the image buffer are described by three RGB components, for checking luminance of each pixel it was necessary to recalculate from RGB space to Y luminance component.

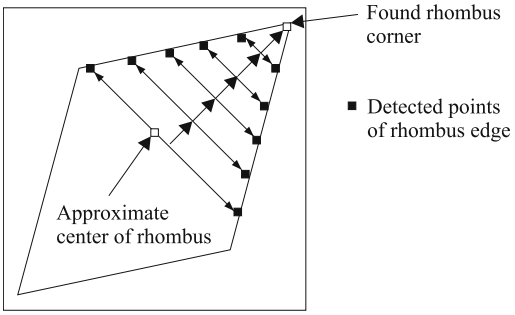


Fig. 4. Scheme of finding sharp rhombus corner

4.3 Recognition of Own Team Robots

Simple comparison of pixel luminance with threshold does not take into consideration direction of edges on the image, so it is necessary to check how the rhombus is oriented before edge following starts. In this way it is judged, if the edge detection should be done along rows or columns of the image. Selection of processing direction is based on check, if longer diagonal of rhombus is oriented more along horizontal or vertical image lines. As it turned out in practice, it is good to distinguish situation, when angle between longer diagonal of the rhombus and image lines or columns is about 45°.

The described method of rhombus edge following and finding of sharp corners is based on the edge detecting along determined directions. The algorithm distinguishes four directions from which one is chosen, depending on approximate orientation of rhombus on the image. In Fig. 5 all four possible directions of cuts of rhombus are shown, which are used in edge detection, starting from approximate center of a rhombus. This method reveals some similarity to edge detection with use of the Sobel operator. The diagram of

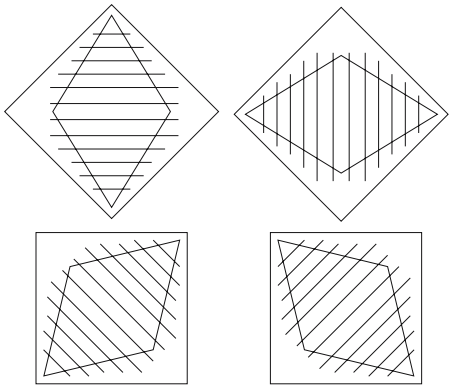


Fig. 5. Distinguished directions of rhombus edges detection

final determining one of sharp corners is presented in Fig. 4. An edge detection is based on test of image function gradient with use of first derivative. The pixels for which absolute value of gradient is big enough are chosen as edge elements. Sobel operator is designed to detect edges independently of their direction on the image.

When sharp corners of rhombus are found, on the basis of their coordinates accurate position and orientation of the robot are calculated. Position of robot is determined as arithmetic average of suitable coordinates of corner points. Orientation is calculated from the equation:

$$\Theta = \mathrm{Atan2}(y_2 - y_1, x_2 - x_1) \,, \tag{1}$$

where pairs $(x_1, y_1), (x_2, y_2)$ mean coordinates of corner points of the rhombus.

After position and orientation are calculated, it is still necessary to determine, which robot has been detected and where is the front of the robot. For this purpose individual robot patch is utilized. This is done by checking in which corner of rhombus individual patch is located and from this front of the robot is determined.

4.4 Determining Position of the Ball and of Opponent Robots Team

Second object, which has to be found by image processing algorithm is the ball. It is the easiest object to recognize on a playground. On image it is represented by circular group of pixels with orange–red color. To determine center of the ball, i.e. ball position on the image, we use a mask of cross shape with width of one pixel and length almost two times greater than diameter of the ball on image. Center of the mask is put on starting point, which was found during image search. After that, point with coordinates equal to arithmetic average of pixels coordinates under the mask, which has color assigned to ball, is determined.

Last group of objects, which should be found and recognized, are opponent team robots. Because in practice different patch markers for different teams are used, positions of robots are estimated by using opponent team color patches. Center of the opponent team patch is determined similar to searching of approximate center of rhombus patch of own team players. Edge detection is done on the base of opponent team color, rather than luminance, because for some configuration of patches results may be incorrect.

5 Measurement Results

During the test 100 measurements were done per several position of a recognized object. For example, the test rhombus patch on black square marker was arranged that longer diagonal of rhombus was at an angle of 45° to image

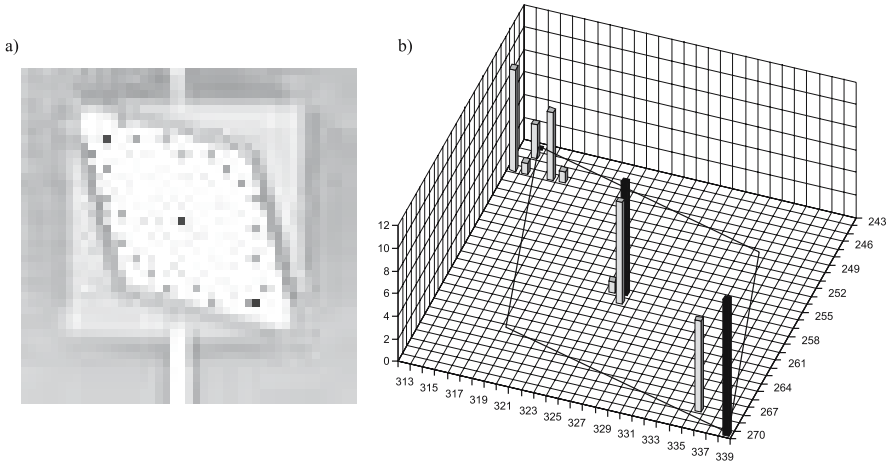


Fig. 6. Photo of patch marker with plotted points for example object position (**a**) and its measurement histogram (**b**)

lines. Based on real and measurement values, maximum measuring error as the highest difference between right values using 100 done measurements were determined, which are:

- first rhombus corner – X coordinate: 3 pixels, Y coordinate: 3 pixels;
- second rhombus corner – X coordinate: 3 pixels, Y coordinate: 2 pixels;
- center of rhombus – X coordinate: 1 pixel, Y coordinate: 1 pixel;
- orientation: 0.08 rad.

In Fig. 6(a) edge points (dark) and found corners and center of rhombus (black) are shown, superimpose on the photo of robot marker for example position and orientation measurement. Figure 6(b) present histogram of pixels found as corners and center of rhombus for 20 measurements. Dark shade represent real points of corners and center of rhombus.

References

1. Corke P.I. (1996) Visual control of robots. Wiley, New York
2. Gonzales R., Woods R. (1993) Digital image processing. Addison-Wesley Publication Company, MA
3. Haralick R.M., Shapiro L. (1992) Computer and robot vision. Addison-Wesley Publication Company, New York
4. Hutchinson S., Hager G.D., Corke P.I. (1996) A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pp. 651–670
5. Dobroczyński D., Dutkiewicz P., Jedwabny T., Kozłowski K., Majchrzak J., Niwczyk G. (2000) Strategy planning for mobile robot soccer. Proceedings of the World Automation Congress, 6 pages (CD-ROM)

6. Dobroczyński D., Dutkiewicz P., Jedwabny T., Kozłowski K., Majchrzak J., Niwczyk G. (2000) Mobile robot soccer. Proceedings of the 6th International Symposium on Methods and Models in Automation and Robotics, pp. 599–604
7. Kiełczewski M. (2000) Hardware and software implementation of position and orientation measurement algorithms of 2D objects. MS Thesis, Chair of Control, Robotics and Computer Science, Poznań University of Technology, Poznań, (in Polish)
8. Kwolek B., Kapuściński T., Wysocki M. (1999) Vision-based impementation of feedback control of unicycle robots. Proceedings of the 1st Workshop on Robot Motion and Control, pp. 101–106
9. Pearson D. (1991) Image processing. McGraw-Hill Book Company, London
10. Sonka M., Hlavac V., Boyle R. (1994) Image processing, analysis and machine vision. Chapman & Hall, London
11. Shim H.S., Jung M.J., Kim H.S., Choi Kim J.H. (1997) Development of vision-basic soccer robots for multi-agent cooperative systems. Proceedings of the Micro-Robot World Cup Soccer Tournament, pp. 29-34
12. MIL-Lite version 6.0. User's guide and command reference. HTML page `www.matrox.com`
13. MIL/MIL-Lite version 6.0. Board-specific notes. HTML page `www.matrox.com`
14. HTML page `www.fira.net`

Part III

Design

Kinematics of a New Staircase Climbing Wheelchair

R. Morales, V. Feliu, A. González, and P. Pintado

School of Industrial Engineering, University of Castilla-La Mancha Campus Universitario s/n 13071, Ciudad Real, Spain
Rafael.Morales@uclm.es

1 Introduction

Powered wheelchairs have been in use for many years now, and it is unquestionable that they greatly improve the mobility of the handicapped. Nevertheless, architectural barriers still exist in many cities and buildings, and it is expensive and time consuming, if not impossible, to eliminate all of them. A wheelchair becomes a useless device when faced with an architectural barrier and, as a result, there have been a number of wheelchair designs that claim to be able to climb stairs. The authors of this paper believe that most of these designs have severe drawbacks that impair their widespread use. Drawbacks are generally related to the lack of safety stemming from mechanically unstable situations during staircase climbing or descending. The mechanisms described in this paper have been designed to enforce mechanical stability while the wheelchair is on the staircase. The weight is transferred at all times to horizontal surfaces (the tread), making it unnecessary to rely on friction to ensure safety. On the other hand, it will be shown in the next section, that the proposed mechanisms can be easily manufactured using mostly off-the-shelf mechanical components.

One may find different designs in the technical literature. Some of these designs are based on several wheels arranged in a rotating link [1]. This system may work satisfactorily for the "design staircase", but proper functioning would be impaired when different treads or step heights are encountered.

The iBOT 3000 [2] is a very compact design that can adjust to different step sizes. Whilst the mechanical design is quite simple, the chair is very sophisticated since it relies on dynamic control to maintain the upright position. There are motion phases during climbing or descend when the chair is standing on just two wheels with a common axis. The major drawback of this design is the tremendous cost necessary to meet reasonable safety standards.

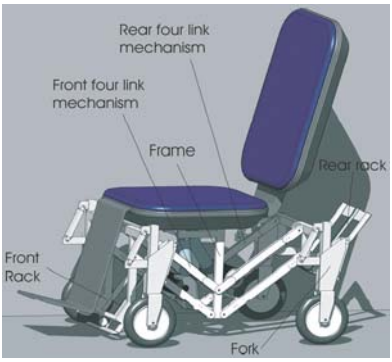


Fig. 1. Schematics of the wheelchair construction. Side view

The tracked chairs [3] have, in our opinion, two serious problems that impair their success. The first problem is that the weight is supported by the vertical component of the contact force between the track and the edge of steps. A high friction coefficient is needed to maintain equilibrium. The second problem is that the entire track is forced to rotate around the edge of the first step when initiating descend. Since there is no support to apply the corresponding torque, the motion must be undertaken using the wheelchair inertia. This process is both difficult and dangerous.

2 Mechanical System Description

The wheelchair described in this paper is based on two different systems: a system to arbitrarily position both axels with respect to the frame to accommodate the overall slope (Fig. 1), and a device that makes the axel climb one step at a time while allowing the wheel to move around the step eluding interfering with it (Fig. 2). In our opinion, the use of two decoupled mechanisms in each axle is the key feature of the mechanical design. The idea may seem quite simple at first sight, but it is responsible of permitting many different climbing strategies, making the design extraordinarily versatile.

In the design of the first system (see Fig. 1), both the front and the rear axles are joined to the frame by means of four link mechanisms. Each four link mechanism is driven by an independent actuator. These mechanisms are parallelograms, which means that the axle doesn't rotate with respect to the frame. In other part, the design of the second system (see Fig. 2), each axle climbs the corresponding step by deploying the rack that is connected to it. This is tantamount to saying that each axle "carries its own ladder". All steps are climbed with the same slope, and the height reached is determined by the step itself, since the wheel moves back to the initial position as soon as the step has been surpassed. This is achieved by connecting the wheel to the axle

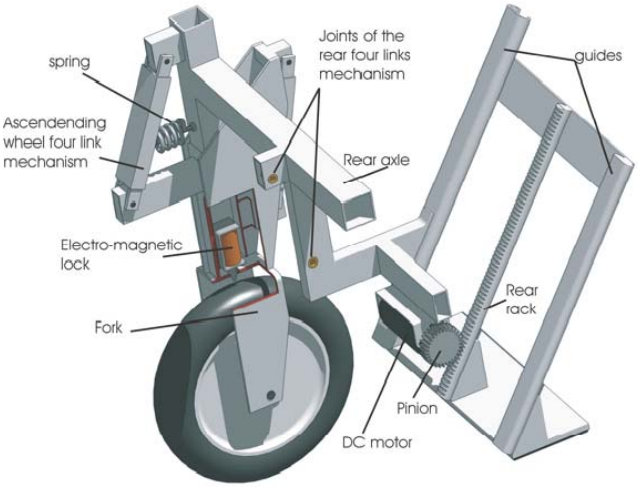


Fig. 2. Cutaway of the mechanism to climb one step

via a four link mechanism. The mechanism may be designed to guide the wheel trajectory with respect to the axle.

This four link mechanism design is quite versatile and could be tuned for different purposes. It could be argued, for example, that the stability condition is rather superfluous since it is safer to lock the wheel fork in its rest position whenever needed.

Finally, the process described has to be synchronized with axle positioning, in order to maintain verticality of the chair frame, and in order to generate the appropriate chair frame trajectory. This synchronization is part of the control strategy that will be outlined in the next section.

3 Stages of the Climbing Motion for the Positioning System and Sensorial System

Once the mechanical system has been described, next we present the basic tasks in which the staircase climbing process is decomposed:

1. Positioning of the wheelchair with respect to the staircase (previous to the climbing process). Alignment between the rear wheels axis and the front of the lower step (Fig. 3).
2. Climbing of the rear wheels while the front wheels remain on the floor (the first few steps of the staircase, Fig. 4).
3. Simultaneous climbing of the rear and front wheels (Fig. 5).
4. Climbing of front wheels with the rear wheels remaining on the upper floor (the last steps of the staircase, Fig. 6).

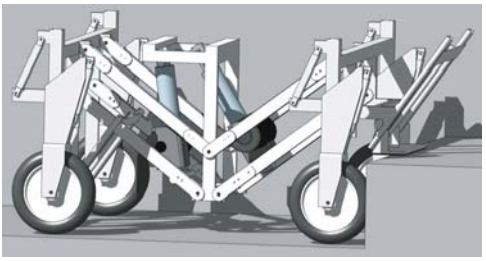


Fig. 3. Schematic view of one side of the mechanism in its position prior to climbing

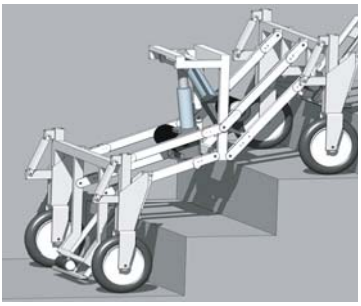


Fig. 4. The rear wheels are on the staircase while the front wheels are on the floor

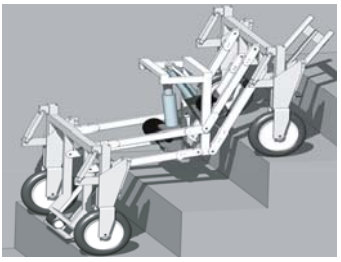


Fig. 5. Both axles on the staircase

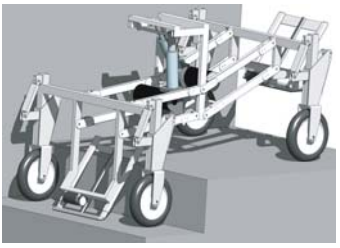


Fig. 6. The rear wheels are on the staircase while the front wheels are on the floor

More information about stages of the climbing motion is supplied in [5].

Moreover the tasks described previously must be carried out in such a way that the centre of mass of the wheelchair describes a specified trajectory, and the inclination is also under control, all this in order to guarantee the comfort of the passenger. Then a relatively complex sensorial system is needed to operate the wheelchair, which is detailed next:

- 8 ultrasound sensors (2 per wheel). They are needed to measure the distances of the wheels to the steps.
- 1 inclinometer which measures the verticality of the chair and detects the instant at which the climbing mechanism contacts the floor when deploying.
- Encoders for the two joints of the chair structure (2), the rear wheels (2), and the racks of the climbing mechanism (2).
- 2 switches that indicate the instant at which the wheel has overpassed the step being climbed (a spring retracts the wheel when it looses contact with the step during its raising process, this sudden motion being detected).

4 Kinematics Model of Control

The most critical part of the whole control system is the subsystem that generates the real time trajectories for the actuators of the wheelchair, in such way that this vehicle should be able to climb and descend staircases keeping the maximum possible comfort for the passenger: smooth motions and very little inclinations from the vertical.

These real time trajectories are the references for the closed-loop systems (servocontrols) that control the angles of the motors (actuators) in charge of moving the several degrees of freedom of our wheelchair. This trajectory generator relies on a kinematic model that should be: (a) precise enough to describe the behaviour of the mechanism, (b) simple enough to be able to be computed in real time, (c) flexible enough to include descriptions of all the tasks mentioned in the previous section, which include different chair configurations and different situations of contact with the environment (floor and staircases).

Figure 7 shows an scheme of the mechanism of the wheelchair, where the most important variables and degrees of freedom are defined. A generic floor profile $\hat{f}(s)$ is assumed, that can be easily particularized to the cases of flat floors or staircases. In what follows we will represent points in the plane by using complex notation: horizontal variable is the real component and vertical variable is the imaginary component. This notation greatly facilitates obtaining kinematic models of our wheelchair because we have found that expression of rotations is simplified leading to more compact equations.

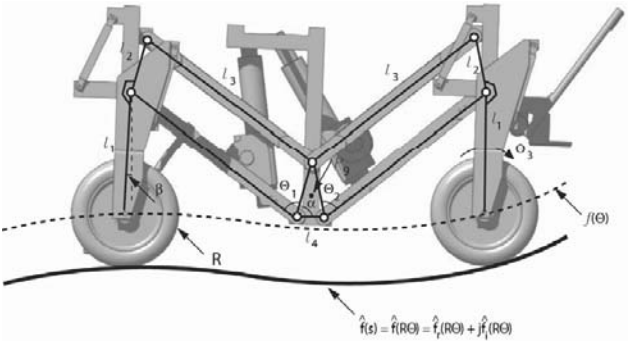


Fig. 7. Kinematic diagram

5 Inverse Kinematic Model

Inverse kinematic model gives de values (angles) of the actuator variables which are needed to achieve a desired centre of mass and inclination of the wheelchair. When the wheelchair climb the stair, it can present in four possible configurations. Next we show all the possible configurations and we show the analytical expressions of the inverse kinematic model.

5.1 Floor Flat with Steps Inverse Kinematic Model

We know the center of mass (P_g) and the inclination of the wheelchair (γ) and the height h between the centre of the wheels and we obtain the two joints θ_1 and θ_2 and the position of the front wheels θ_3. This configuration is show in Fig. 8. The equations are presented next:

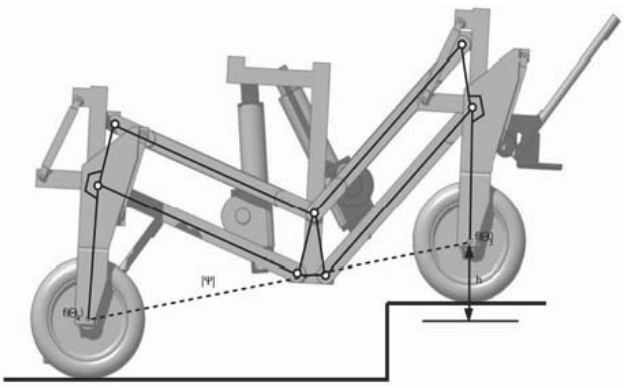


Fig. 8. Floor flat with steps

$$A_{\theta 3} = P_{gx} - X_4 - l_1 \cos\left(\frac{\pi}{2} - \beta + \gamma\right) - \frac{l_4}{2}\cos\gamma - \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2}\cos\left(\frac{\pi}{2} + \gamma\right) \tag{1}$$

$$\theta_3 = \frac{1}{R}\left(A_{\theta 3} - \sqrt{l_3^2 - B_{\theta 3}^2}\right) \tag{2}$$

$$B_{\theta 3} = P_{gy} - Y_4 - l_1 \sin\left(\frac{\pi}{2} - \beta + \gamma\right) - \frac{l_4}{2}\sin\gamma - \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2}\sin\left(\frac{\pi}{2} + \gamma\right) \tag{3}$$

$$\theta_1 = \pi - \alpha - \gamma + ang\left(\frac{B_{\theta 3}}{A_{\theta 3} - R\theta_3}\right) \tag{4}$$

$$A_{\psi} = P_{gx} - X_4 - R\theta_3 - l_1 \cos\left(\frac{\pi}{2} - \beta + \gamma\right) - \frac{l_4}{2}\cos\left(-\pi + \gamma\right) - \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2}\cos\left(\frac{\pi}{2} + \gamma\right) \tag{5}$$

$$\sqrt{|\psi|^2 - h^2} = \left|A_{\psi} - \sqrt{l_3^2 - B_{\psi}^2}\right| \tag{6}$$

$$B_{\psi} = P_{gy} - Y_4 - h - l_1 \sin\left(\frac{\pi}{2} - \beta + \gamma\right) - \frac{l_4}{2}\sin\left(-\pi + \gamma\right) - \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2}\sin\left(\frac{\pi}{2} + \gamma\right) \tag{7}$$

$$\theta_2 = \gamma - \alpha - ang\left(\frac{B_{\psi}}{A_{\psi} - \sqrt{|\psi|^2 - h^2}}\right) \tag{8}$$

5.2 Wheelchair Starts to Raising the Steps with Rear Rack and Front Wheel is on the Flat Inverse Kinematic Model

We know the two joints θ_1 and θ_2, the contact point P_{C} between the rear rack and the flat, and the instantaneous length of the rear stem (z), and we obtain the center of mass (P_{g}), and the inclination of the wheelchair (γ). This configuration is show in Fig. 9. The equations are presented next:

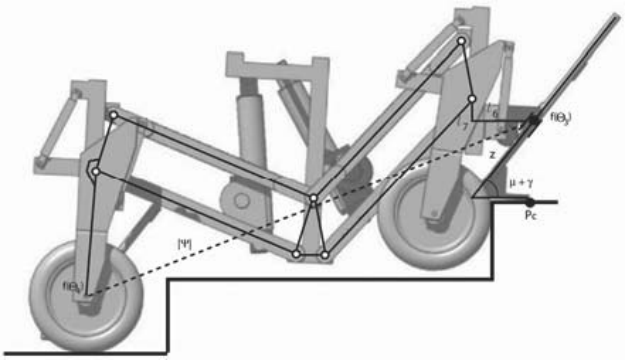


Fig. 9. Wheelchair start to raising the steps with rear rack and front wheel in on the flat

$$\Delta P = \Delta P_r + j\Delta P_i = \left[P_g - P_C + l_7 - jl_6 - \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2}\, e^{j(-\frac{\pi}{2}+\gamma)} \right] \times e^{j(-\mu-\gamma)} \tag{9}$$

$$\theta_2 = \pi + \arcsin\left(\frac{\Delta P_i}{l_3}\right) - \alpha - \mu \tag{10}$$

$$A_{\theta 4} = P_{gx} - X_4 - l_1 \cos\left(\frac{\pi}{2} - \beta + \gamma\right) - \frac{l_4}{2}\cos\gamma - \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2} \times \cos\left(\frac{\pi}{2} + \gamma\right) \tag{11}$$

$$z = \Delta P_r - l_3 \cos(-\theta_2 - \alpha - \mu) \tag{12}$$

$$B_{\theta 4} = P_{gy} - Y_4 - l_1 \sin\left(\frac{\pi}{2} - \beta + \gamma\right) - \frac{l_4}{2}\sin\gamma - \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2} \sin\left(\frac{\pi}{2} + \gamma\right) \tag{13}$$

$$\theta_4 = \frac{1}{R}\left(A_{\theta 4} - \sqrt{l_3^2 - B_{\theta 4}^2}\right) \tag{14}$$

$$\theta_1 = \pi - \alpha - \gamma + ang\left(\frac{B_{\theta 4}}{A_{\theta 4} - R\theta_4}\right) \tag{15}$$

5.3 Wheelchair Starts to Raising the Steps with Front Rack and Rear Wheel is on the Flat Inverse Kinematic Model

We know the center of mass (P_g) and the inclination of the wheelchair (γ), the height h between the centre of the wheels, and the contact point P_C between the front rack and the flat, and we obtain the two joints θ_1 and θ_2, the position

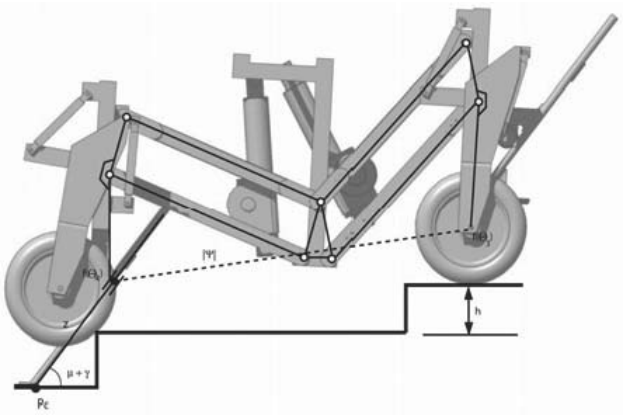


Fig. 10. Wheelchair start to raising the steps with front rack and rear wheel is on the flat

of the front wheels θ_3 and the instantaneous length of the front stem (z). This configuration is show in Fig. 10. The equations are presented next:

$$\Delta P = \Delta P_r + j\Delta P_i = \left[P_g - P_C - l_1 e^{j\left(\frac{\pi}{2}-\beta+\gamma\right)} - \frac{l_4}{2}e^{j\gamma} - \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2}\, e^{j\left(\frac{\pi}{2}+\gamma\right)}\right] e^{j(-\mu-\gamma)} \tag{16}$$

$$z = \Delta P_r - l_3 \cos\left(\theta_1 + \alpha - \pi - \mu\right) \tag{17}$$

$$A_\psi = P_{gx} - P_{Cx} - z\cos\left(\mu+\gamma\right) - l_1 \cos\left(\frac{\pi}{2}+\beta+\gamma\right) - \frac{l_4}{2}\cos\left(-\pi+\gamma\right) - \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2}\cos\left(\frac{\pi}{2}+\gamma\right) \tag{18}$$

$$\theta_1 = \pi + \mu - \alpha + \arcsin\left(\frac{\Delta P_i}{l_3}\right) \tag{19}$$

$$\theta_3 = \frac{1}{R}\left|P_C + z\, e^{j(\mu+\gamma)} + \psi - X_4 - jY_4 - jh\right| \tag{20}$$

$$\theta_2 = \gamma - ang\left(\frac{B_{\theta 3}}{A_{\theta 3} - R\theta_3}\right) - \alpha \tag{21}$$

5.4 Wheelchair Starts to Raising the Steps with Front and Rear Rack Inverse Kinematic Model

We know the center of mass (P_g), the inclination of the wheelchair (γ), and the contact points P_{C1} and P_{C2} between the front and rear racks and the flat,

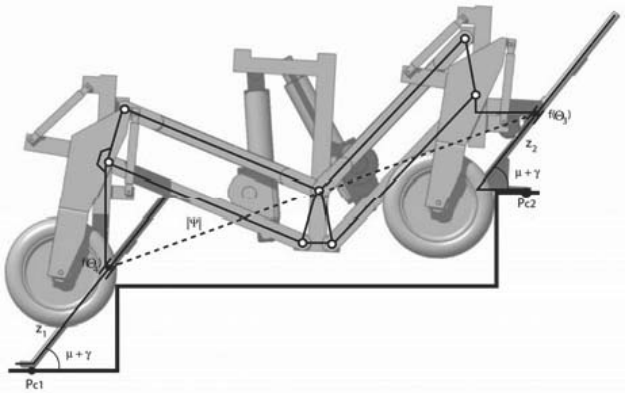


Fig. 11. Wheelchair start to raising the steps with front and rear rack

and we obtain the two joints θ_1 and θ_2 and the instantaneous lengths of the front and rear stems (z_1 and z_2) . This configuration is show in Fig. 11. The equations are presented next:

$$\Delta F = \Delta F_r + j\Delta F_i = e^{j(-\mu-\gamma)} \left[P_g - P_{C1} - l_1 e^{j\left(\frac{\pi}{2}-\beta+\gamma\right)} - \frac{l_4}{2} e^{j\gamma} + \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2}\, e \right] \tag{22}$$

$$z_1 = \Delta F_r - l_3 \cos\left(\theta_1 + \alpha - \pi - \mu\right) \tag{23}$$

$$\Delta G = \Delta G_r + j\Delta G_i = [P_g - P_{C2} + l_7 - jl_6 + l_5 - \frac{1}{2}\sqrt{l_2^2 - \left(\frac{l_4}{2}\right)^2}\, e^{j\left(-\frac{\pi}{2}+\gamma\right)} \Bigg]\, e^{j(-\mu-\gamma)} \tag{24}$$

$$\theta_1 = \pi + \mu - \alpha - \arcsin\left(\frac{\Delta F_i}{l_3}\right) \tag{25}$$

$$\theta_2 = \pi + \arcsin\left(\frac{\Delta G_i}{l_3}\right) - \alpha - \mu \tag{26}$$

$$z_2 = \Delta G_r - l_3 \cos(-\theta_2 - \alpha - \mu) \tag{27}$$

6 Movement Conditions

The trajectory desired for the point P_g is designed by linking two arcs of circumference and one straight trajectory. Also, the movement must be comfortable for the passenger. This constraint implies that we can't exceed the maximum acceleration and the maximum velocity defined, and that verticality of the seat must be maintained. In Fig. 12, we can see the global trajectory

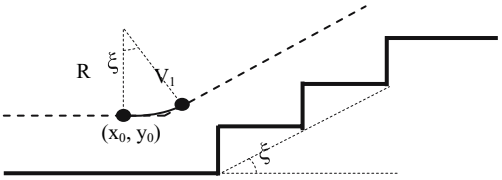


Fig. 12. Geometry of the design trajectory

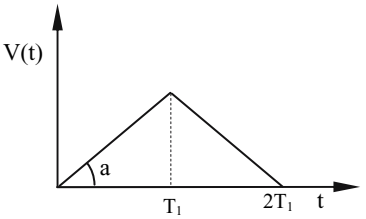


Fig. 13. Profile velocity in the part of the circumference trajectory

desired by the wheelchair. And in Fig. 13, we can see the profile of velocity desired. We assume that the velocity is $V = a \cdot t$ in the interval (0, T_1), where a is the tangential acceleration. The results are:

$$x(t) = x_0 + R \sin\left(\frac{a}{2R}t^2\right) \tag{28}$$

$$y(t) = y_0 + R - R \cos\left(\frac{a}{2R}t^2\right) \tag{29}$$

$$V_x(t) = at \cos\left(\frac{a}{2R}t^2\right) \tag{30}$$

$$V_y(t) = at \sin\left(\frac{a}{2R}t^2\right) \tag{31}$$

$$a_x = a \cos\left(\frac{a}{2R}t^2\right) - \frac{a^2t^2}{R} \sin\left(\frac{a}{2R}t^2\right) \tag{32}$$

$$a_y = a \sin\left(\frac{a}{2R}t^2\right) + \frac{a^2t^2}{R} \cos\left(\frac{a}{2R}t^2\right) \tag{33}$$

If we assume that the velocity of P_g in $t = T_1$ is V_{max}, the conditions of tangencial acceleration a, curvature radius R, and time T_1 to achieve V_{max} are:

$$a = \frac{a_{max}}{\sqrt{1 + (2\xi)^2}} \tag{34}$$

$$R = \frac{V_{max}^2}{2\xi a} \tag{35}$$

$$T_1 = \sqrt{\frac{2R\xi}{a}} \tag{36}$$

In the another parts of the trajectory, the movement is made with a_{max}.

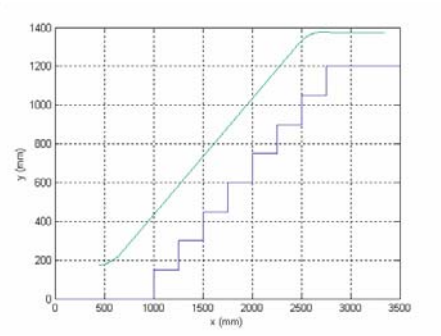


Fig. 14. Trajectory of center of mass

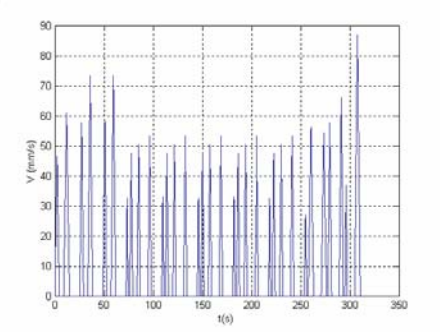


Fig. 15. Velocity of the center of mass

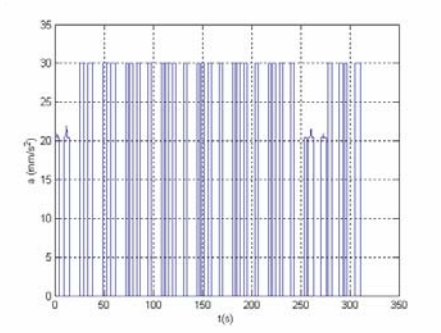


Fig. 16. Acceleration of the center of mass

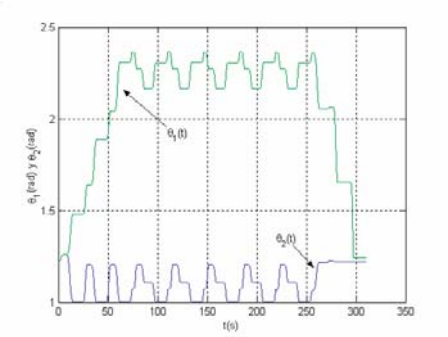


Fig. 17. Angles of joints

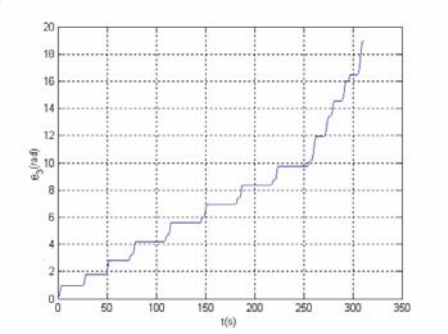


Fig. 18. Position of the rear wheels

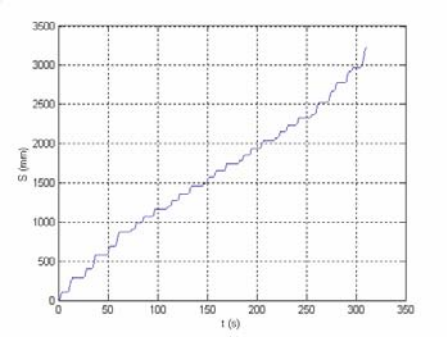


Fig. 19. Trajectory evolution

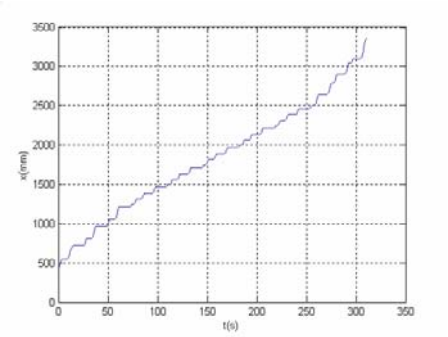


Fig. 20. Horizontal position center of mass

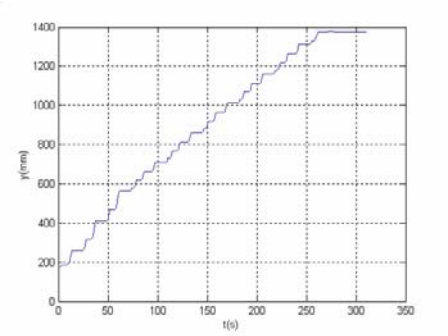


Fig. 21. Vertical position center of mass

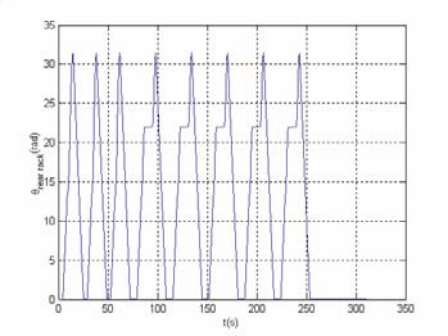


Fig. 22. Angle of rear rack

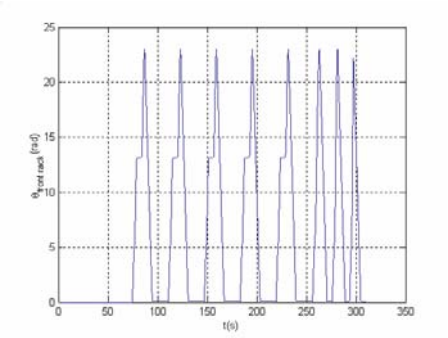


Fig. 23. Angle of front rack

7 Simulation Results

This is one example of the climbing stair movement of the wheelchair. We suppose one stair with eight steps. The dimensions of the steps are 150 mm (height) $\times$ 250 mm (width). The trajectory must be comfortable for the handicapped, this means that accelerations and velocities must be shorter than the maximum acceleration of comfort and maximum velocity of comfort, and the inclination of the wheelchair must be null ($\gamma = 0$). These conditions involve that our movement will be composed of two movements (one of them to accelerate the wheelchair and the other to deccelerate it). The results of the simulation are:

In this simulation, we have obtained one important result: 60% of full the time needed to climb the staircase is a dead time. One of the future work will be the design of new trajectories that minimize the existence of dead times.

8 Conclusions

A new staircase climbing wheelchair has been designed whose main characteristic is being especially stable. Moreover, its additional degrees of freedom allow motions and control strategies that take into account the comfort of the passenger (i.e. maximum acceleration and inclination of the wheelchair). Kinematic models have been developed for the different stages of motion and this models are simple enough to be used for real time control purposes (iterative calculation procedure are not necessary). Finally, as an application of the previous models, control signals for the actuators of the system have been generated in order to drive the wheelchair in an open loop fashion.

At this moment, we are building a first prototype of this mechanism and we hope to run the first climbing experiments (with the sensorial and control systems included) as soon as possible.

Acknowledgments

The authors would like to acknowledge the support provided by the Spanish CICYT (Comisión Interministerial de Ciencia y Tecnología) under Grant DPI2001-1308-CO2

References

1. Prototype of of Tamagawa University School of Engineering. www.tamagawa.jp.
2. Ibot^{TM} 3000 is a product of Independence Technology a Johnson & Johnson Company. www.indetech.com.
3. Stairclimber of The Wheel Chair Lift Company www.thewheelchairlift.co.uk
4. Murray J. Lawn and Takakazu Ishimatsu, "Modeling of a Stair-Climbing Wheelchair Mechanism With High Single-Step Capability". *IEEE Trans. On Neural Systems and rehabilitation engineering*, vol. 11 no. 3, September 2003.
5. R. Morales, A. González, P. Pintado, V. Feliu, "A new staircase climbing wheelchair", ISMCR03, Madrid, December 2003.

Open Modular Design for Robotic Systems

I. Chochlidakis, Y. Gatsoulis, and G.S. Virk

Intelligent Systems Group, Department of Mechanical Engineering, University of Leeds, Woodhouse Lane, Leeds, West Yorkshire, UK
men2ic@leeds.ac.uk
menig@leeds.ac.uk
g.s.virk@leeds.ac.uk

Abstract. Reconfigurable robots offer a new approach in robotic systems so that particular solutions can be easily put together by integrating suitable modular components. The CLAWAR approach to modular design using concepts of basic modules and super-modules to the cooperative systems are extended to develop a generic methodology for the design environment. A case study is also presented for designing climbing robots for biomedical applications based on this methodology.

1 Introduction

A modular reconfigurable robotic system is a collection of individual components that can form different robot configurations and geometries [3] for specific task requirements. The CLAWAR community has proposed the development of basic modules [18] (Input, Output, Processing, and Infrastructure) and super-modules that can integrate to each other via the interaction space (comprising power, databus, environment, digital, analogue and mechanics). These concepts are used here to propose the development of software design tools to assist robotisists to realise effective solutions by selecting components from a library of modules. In this way additional modules may be created to enhance the database.

Compared to a conventional robot system whose design is based on the use of specific parts and processes peculiar to the designer to an open modular designed system offers many advantages; these include the following:

- All the modules do not need to be designed from scratch but chosen from those that already exist. Such reuse will allow much faster and cheaper designs to be realised.
- The design can focus on the new modules needed allowing more time if necessary to realise better performances.
- The design is flexible and can be easily adapted to changing needs.

Of course for this work, a viable framework that is acceptable to the wide community needs to exist. For this reason the CLAWAR community has been working on developing robot component modularity and our desire to support this work.

2 Requirements of Modular Robots

Compatibility among the modules is a critical issue especially when tasks such as fire-fighting, urban search and rescue after earthquakes or natural disasters and battlefield reconnaissance where robots face unexpected situations and many kinds of obstacles [11] while in need to fulfil their complicated mission. The compatibility among connectable modules is a result of the design and implementation of the connectors. These connectors allow a module to connect and disconnect with one another in a simple way. Hence, the mechanics of the connector modules must accomplish the following four requirements.

(a) Power efficiency – as limited on-board power supplies exist at present
(b) Reliable – connectors must endure various operations
(c) Compact (Miniaturization) – the mechanism must fit into a tight place
(d) Designed with open modularity – newly designed modules should be able to connect to older designs

The need for structured modular approach for robot design is a major factor among cooperating supply-chain based companies as it speeds up the whole NPD (New Product Development) process. For medical robots, even for the external non-miniature ones used worldwide a certain number of requirements are common even though most of the time requirements depend on the environment of use [1, 8] and [12]. The four main modules of any modular robot design have been classified by the CLAWAR community as *input* modules that have to do with inputting information to the robot, *output* modules that provide the interaction to the robot's environment and users, *processing* modules which represent the decision making processes with the robot (normally the software algorithms) and finally *infrastructure* components which support the overall processes needed by the robot (e.g. power, materials, communications, etc).

All robotic systems have specific requirements but when considering a generic design environment (Fig. 1) for a specific application there are two levels that the applications is divided; System Level [17] and Module Level [18]. We need to look at how the basic modules (input, output, processing and infrastructure) can be provided by the available software tools. To cover a great range of modules various robotic applications were studied (Wheeled [13], Climbing [2, 6, 10, 14] and [15], Crawling [3, 14] and Walking [7, 15])

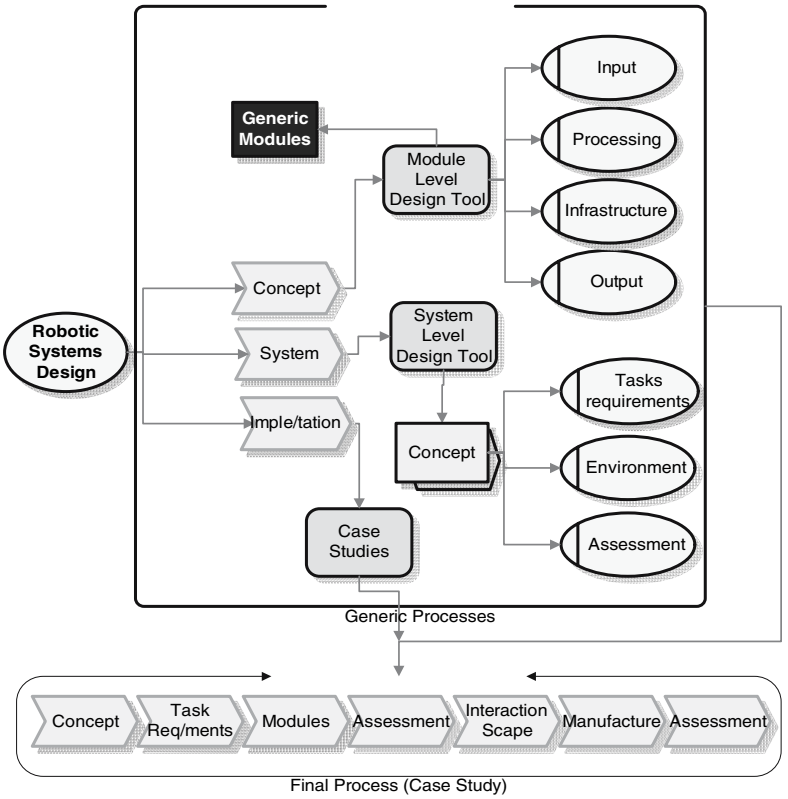


Fig. 1. Overall Generic System Design

2.1 Input Modules

Many input modules are needed to allow data and information and commends to be inputted into the robot. A few examples are sensors, user commands [4] and GPS. Sensors are the modules that provide systems with the required data; they can be of many types and have a number of attributes (data type, range of operation, environment [4, 11], speed of response, etc). A navigation module can give the ability to navigate the system manually while a GPS can perform this automatically by providing the localisation.

2.2 Processing Modules

Processing modules are normally non-physical modules which implement and process the information received to perform decision making. Examples of such modules include data analysis, information manipulation, signal processing, learning and navigation.

2.3 Infrastructure Modules

Infrastructure modules provide some support function. This can be on the mechanical support side, powering aspects or maintenance of the system. Examples of infrastructure modules include materials, power, computing hardware, etc.

2.4 Output Modules

Output modules provide the direct response to the environment and the user. These modules include actuators/manipulators for handling or delivery of objects, user feedback, and collaboration with other robots, etc.

The generic basic modules in a generic robotic system are shown in Fig. 2 in a form of a tree. These can be group to group a range of super-modules [7–10].

3 Software Selection

It is vital to be able to design and simulate the operational environment that the system will work in; hence the need for good design and simulation tools. This should state the selection of possibly several software packages with the required "bridges" or "plug-ins" that can dynamically connect the various applications. For the open modularity concept to work, the software selected

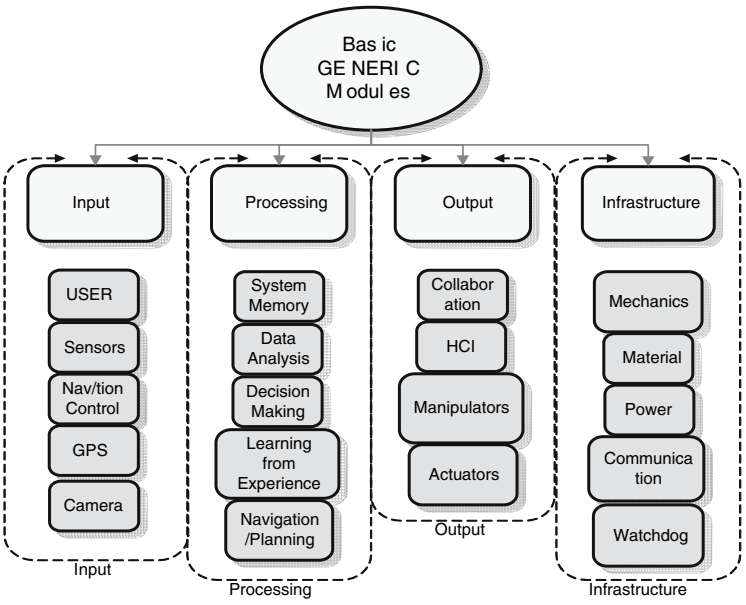


Fig. 2. Generic Modules

needs to be preferably based on freeware or relatively inexpensive packages. The criteria that need to be used in selecting the software is complex but needs to include the following [16, 17]:

- Cost and usability; (training time needed, applications it can handle etc)
- Expandability to new versions
- Help and support available from the supplier and/or user groups
- User friendliness of the HCI
- Complexity of the environment that can be handled; (1D, 2D, 3D, static or
- dynamic situations)
- Simulation aspects provided (robot(s), task(s), environment(s))

The chosen software must be able to provide the ability to design and simulate each of the generic modules needed. In addition it needs to allow three design and simulation functions to be performed, namely allow for the design of robots, the working environments to be set up and it must allow designers to study tasks that need to be performed. Many software applications have been developed and are available. A list of over 100 titles has been identified and the selection criteria for choosing the best for specific cases are presented in [17]. The software selection process identified software for the System and Modular Levels. Of these, the ones presented in the following list were considered to be the most appropriate for the Modular Level Design.

- Visual Nastran for specific limb motion and 3D Design
- Yobotics; Similar to visual Nastran including Biomedical Robot design, with built-in control system
- Darwin2K a dynamic simulation and automated design synthesis package for robotics

4 Colonoscopy Robot Inspection Case Study

The case study discusses the modular design of a mobile robot for colonoscopy [1, 12]. It introduces and proposes design environments for mobile climbing robots on irregular and/or sleek terrains. There are two general categories of climbing scenarios depending on the application; the first includes all kinds of terrain vertical or horizontal, rough or smooth, while the second arises in medical applications where wet and dry, rough and smooth, and rigid and pressure-deformed surfaces need to be addressed.

The ability to handle these types of surfaces and various robot kinematic and dynamic mechanics needed to be simulated in the appropriate internal body environments. In order to visualize the problem(s) and invent new solutions, they an be studied in the simulated set up before risking having to build anything. Applications other than medical ones can be simulated by various software packages that provide 3D design and simulation engines together with

CAD capabilities for kinematic analysis. As a result of the research carried out on the software packages available, packages like Visual Nastran, Darwin2K and Yobotics were discussed to be the most appropriate. Visual Nastran is felt to be most appropriate and will be used in this case study for specific limb motion as it gives a variety of joint types and motions with various properties to match different design and simulation concepts. Player Stage (system level tool) is another powerful simulation studio which includes mapping, localization both in 2D and 3D with on board camera views, path planning and 3D world design and all these make it a useful robot design tool. Darwin2k is a combination of the player-stage system level software but with module level design capabilities. It has not being tested yet but demos have been run and it is also considered and believed to be used.

The CLAWAR design approach to start with formulating the system level requirements and breaking these into appropriate modules level sub-designs is followed here. Regarding any application, the system level design should come before the modular level design. The details of how a system level approach can be realised is discussed in [17] and is used in this case study in order to continue to the modular level design. The design aspects for the modular design are focussed upon here.

For the application of colonoscopy, 3D environments are vital for realising effective robot designs. All mobile robots in any environment need to sense, make some decisions and move themselves or move something. This is also the case for robot colonoscopy systems. By completing the system level design first it is possible to produce a list of the necessary modules from considering the task requirements regarding the specific application. The colonoscopy robot should be biocompatible and able to navigate within the restrictive confines of the large intestine. The main constraint apart from the biocompatibility and strength for locomotion is that it must be compact and able to stabilize itself while navigating. In other words a mechanism must allow the robot to either grip or stick to the walls around it without rupturing them and be able to manoeuvre freely.

Having this in mind the System Level Tasks are listed below:

Working Environment: *The Inner Human Body:*

(a) 3D, Intestine (Colon) – tight confined environment
(b) Pressure-deformed surfaces
(c) Danger of rupturing walls
(d) Sleek/ smooth walls
(e) Liquid flow might be present

Task to be carried out

(a) Semi-autonomous in navigation
(b) Allow medical examination
(c) Provide visual feedback to doctors
(d) Perform medical procedures (remove tumours)

Performance metrics

(a) Safe: biocompatible
(b) Effective: locomotion, inspection, treatment
(c) Compact
(d) Reliable

Operation

(a) Remote
(b) Semi-autonomous

This way of approach highlights the need for different software for the environment design and simulation. Firstly any possible CAD or robot design and simulation software could work well for applications involving rigid pipes in petrochemical plant situations [6, 14], but the introduction of soft tissue tubes in biomedical situation such as in colonoscopy exposes the need for a new type of software where pressure-deformed surfaces and tissue dynamics are included [8]. Pressure-deformed surfaces lead to the need to alter the locomotion methods. End effectors or sensors (grippers/manipulators, pressure sensors, camera, blood flow sensors, temperature sensors, distance; and collision avoidance detection sensors) or mechanics are the system level design requirements. The "pressure-deformed" characteristic of the colonoscopy environment leads to the formulation of the following list of requirements:

- Safety: need for umbilical to retrieve device
- Soft surface dynamics: need for special contact mechanisms
- Mechanics: "wet" bioactive environment.
- Motion planning: Movement without damaging delicate intestine wall
- Umbilical: useful for power, communication to/from device

Simulating soft surface dynamics and in general the whole internal-to-body environments such as those of the colon is not widely available. Legged locomotion on soft ground has been considered [5] but to the authors knowledge there is no software suitable for simulation of internal body environments. In view of this rigid pipes have been used to simulate the colonoscopy application. Hence in the simulations the pressure at the contact points assuming rigid pipes is monitored and controlled so that it does not exceed the threshold for causing rupturing of the colon wall.

CLAWAR's generic methodology uses four basic modules to interact via six variables (power, mechanics, data bus, analogue signals, digital signals and the environment) to integrate with each other to allowing applications specific solutions to be formed. These main basic modules that are needed comprise sensors actuators, power supplies, computing hardware, software, communication devices and the materials, [17–21]. In order to realise and implement the open design methodology we need to determine a good way of integrating the modules by having open protocols for the interfacing allowing the modules to be seen as "black boxes". In order to do this we introduce the

Module Title:		
INPUT	Extra Details	OUTPUT
Power:		Power:
Analogue:		Analogue:
Digital:		Digital:
Databus:		Databus:
Environment:		Environment:
Mechanics:		Mechanics:

Fig. 3. The Module Block representing the inputs and outputs categorized into the six interaction-space variables

Common Module Block shown in Fig. 3 where the input and output variables can be specified.

The "input variables" simply state the input requirements of the module while the "output variables" categorize what is being output for the six interfacing space variable. When a few modules have been designed so that they match and can be integrated, it is hoped that a mature and robust methodology and set of standards can be determined that will be acceptable to the CLAWAR and wider robot communities. Having as reference the subsumption architecture which Brooks established [22], we can extend it to have the emphasis on modular components and how they can integrate together; the concept is shown more clearly in Fig. 4.

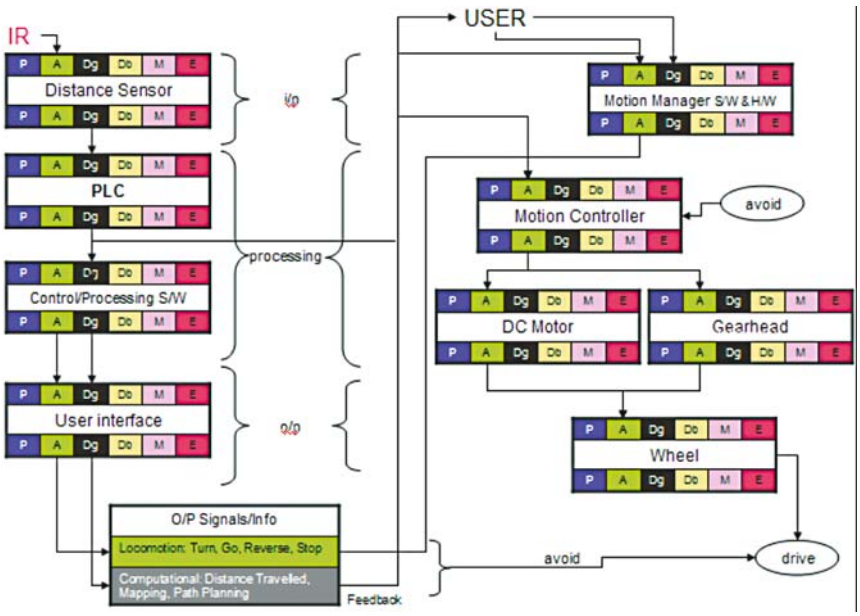


Fig. 4. Module Architecture

The basic concept is that the modular approach can be followed up to build up specific sense-processing-output paths within the robot's actions to build up the design attributes to meet the system level task requirements. In this way the designer can grow the modular level design using an appropriate toolset that allows him to test the inter-connectivity of the modules while assessing if the overall system can perform the required tasks.

5 Conclusions

The paper has presented a generic method to support the CLAWAR open modular approach. Appropriate software tools (using freeware software as much as possible) are needed to allow designs to be put together and tested in a virtual way before actual building is initiated. In this way modular components can selected from a data base to realise the functionalities needed and speed up the process. The software most suitable has been investigated for this purpose and Player-Stage is felt to be the most appropriate. A case study involving colonoscopy has been presented to highlight the details of the generic methodology and this has highlighted the shortcomings in the available tools, namely no software appears to be available for simulating soft tissue applications.

References

1. A. Menciassi, J.H.P., S. Lee, S. Gorini, P. Dario, Jong-Oh Park. *Robotic Solutions and Mechanisms for a Semi-Autonomous Endoscope*. In *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*. 2002. EPFL, Lausanne, Switzerland.
2. B.L. Luk, D.S.C., A.A. Collie, N.D. Hewer, S. Chen. *Intelligent Legged Climbing Service Robot For Remote Inspection And Maintenance In Hazardous Environments*. In *8th IEEE Conference on Mechatrinics and Machine Vision in Practice*. 2001. Konk Kong.
3. Bose, A.K., *Modular Robotics*, Department of Mechanical Engineering, Rashtreeya Vidyalaya College of Engineering: Bangalore. pp. 1–16.
4. David J. Bruemmer, D.D.D., Mark D. McKay, Matthew O. Anderson, *Dynamic-Autonomy for Remote Robotic Sensor Deployment*, The Human-System Simulation Laboratory, Idaho National Engineering and Environmental Laboratory.
5. Iikka Leppanen, Sami Salmi and Aarne Halme, *Workpartner – Hut Automation's new Hybrid Walking Machine*. IMSRI, Clawar 98, 1998.
6. Hisanori Amano, K.O., Tzyh-Jong Tarn. *Development of vertically moving robot with gripping handrails for fire-fighting*. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2001. Maui, Hawaii, USA.
7. Jesse A. Reichler, F.D., *Dynamics Simulation and Controller Interfacing for Legged Robots*. The International Journal of Robotics Research. **19**(01): pp. 41–57.

8. L. France, J.L., A. Angelidis, P. Meseure, M.-P. Cani, F. Faure, C. Chaillou, *A Layered Model of a Virtual Human Intestine for Surgery Simulation.* Elsevier Science, 2004: pp. 1–20.
9. Luis E. Navarro-Serment, R.G., Christiaan J.J. Paredis, Pradeep K. Khosla, *Modularity in small distributed robots*, Institute for Complex Engineered Systems, The Robotics Institute, and Department of Electrical and Computer Engineering, Carnegie Mellon University: Pittsburgh, Pennsylvania 15213. pp. 1–10.
10. Mark A. Minor, R.M., *Under-Actuated Kinematic Structures For Miniature Climbing Robots.* ASME Journal of Mechanical Design, 2002.
11. Michael Beetz, T.B., *Autonomous Environment and Task Adaptation for Robotic Agents*, University of Bonn, Dept. of Computer Science III: Bonn, Germany.
12. P.M.Y. GOH, K.K., *Microrobotics in surgical practice.* British Journal of Surgery, 1997. **84**: pp. 2–4.
13. P.S. Schenker, P.P., B. Balaram, K.S. Ali, A. Trebi-Ollennu, T.L. Huntsberger, H. Aghazarian, B.A. Kennedy and E.T. Baumgartner, K. Iagnemma, A. Rzepniewski, and S. Dubowsky, P.C. Leger and D. Apostolopoulos, G.T. McKee, *Reconfigurable robots for all terrain exploration*, Jet Propulsion Laboratory, Massachusetts Institute of Technology, Carnegie Mellon University, University of Reading (UK). pp. 1–15.
14. R. Aracil, R.S., O. Reinoso, *Parallel robots for autonomous climbing along tubular structures.* Robotics and Autonomous Systems, © 2002 Elsevier Science B.V., 2002/3. **42**: pp. 125–134.
15. S Galt, B.L.L., D.S. Cooke, A.A. Collie, *A tele-operated semi-intelligent climbing robot for nuclear applications*, IEEE, Editor. 1997, Department of Electrical and Electronic Engineering, University of Portsmouth – Portsmouth Technology Consultants Ltd. pp. 118–123.
16. Williams, L.D.a.G., *Evaluating and selecting simulating software using the analytic hierarchy process.* Integrated Manufacturing Systems, 1994. **5**(1): pp. 23–32.
17. Yiannis Gatsoulis, I.C., Gurvinder S. Virk. *Design Toolset for Realising Robotic Systems.* In *CLAWAR 2004.* 2004. Madrid.
18. G.S. Virk, *CLAWAR Modularity for Robotic Systems* The International Journal of Robotics research, Vol. 22, No. 3–4, MArch-April 2003, pp. 265–277, Sage Publications
19. G.S. Virk, *Technical Task 1. Modularity for Clawar machines – Specifications and possible Solutions* Clawar Network, 1999
20. G.S. Virk, *Task 1 Report on "Modularity for Clawar Machines – Specification and Possible Solutions"*, Year 1 Report to the EC for CLAWAR, EC Contract Number BRRT-CT97-5030, 1999
21. P. Maly, G.S. Virk, I. Kagalidis, D. Howard, A. Vitko and L. Jurisica, *Modular Design For Robotic Systems*
22. Rodney, A. Brooks, *A Robust Layered Control for a mobile Robot*, A.I. Memo 864, September 1985, Massachusets Institute of technology

Mechanical Design Optimization of a Walking Robot Leg Using Genetic Algorithm

C. Reyes and F. Gonzalez

Departamento de Ingeniería Industrial y de Sistemas, Universidad Nacional de Colombia, Bogotá D.C., Colombia
{cnreyesp, fagonzalezo}@unal.edu.co

Abstract. This article presents the mechanical design optimization of a walking robot leg using genetic algorithm in order to minimize induced torques on motors that make leg mechanism move, along support phase of step.

1 Introduction

The design of mechanisms and structures is an important part of the robot design process. It involves the analysis of forces and torques induced in different components generated by movement, to generate information that supports decision making and to help to focusing the design to fulfill project constraints.

A walking robot is a dynamic system. Therefore, its analysis requires the definition of basic walking design factors such as duty factor, displacement velocity, step period, and robot mass that allows the calculation of contact forces that are exerted directly on leg mechanism [1]. Contact forces and robot massic properties allow the static and dynamic analysis of leg behavior. This analysis might show possible failures and weaknesses, which could be used to improve the design.

This paper describes the application of a genetic algorithm for design optimization of the leg mechanism of the walking robot shown in Fig. 1 and Fig. 2. The goal is to find the constructive dimension set that minimizes induced torques in leg motors. Smaller torques require motor with less power; this means lighter and cheaper motors and robots with lower energy consumption.

This paper is organized as follows: Sect. 2 presents a review of genetic algorithms applied on mechanical optimization, Sect. 3 introduces the leg design problem and a physical model of the proposed leg mechanism, Sect. 4 presents a genetic algorithm for optimizing the leg mechanism – representation, fitness evaluation and genetic operators. In Sect. 5 some experimental

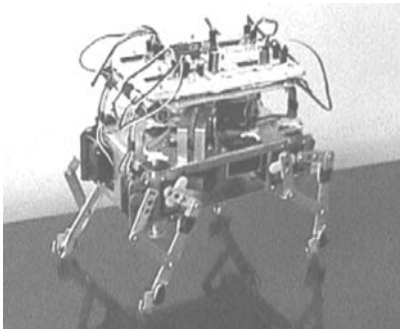


Fig. 1. Perspective view of the robot

results are presented and discussed. Finally, Sect. 6 presents the conclusions of the work. A final appendix shows used equations.

2 Previous Work

Genetic algorithms have been used in mechanism design, and in mechanism features optimization. Some applications are, among others, landing gear optimization [4], wing shape optimization on aeronautics [10], and suspension design on automotive industry [5]. Also, genetic algorithms have been applied on robot design to solve problems such as trajectories optimization [6], robot gripper's design [7], and counterweight balancing [8]. In the area of walking robots, GA application has been focused on central pattern generators (CPG), gait generators, interaction with environment, and walking learning. To our knowledge, there are no precedents for applying GA to the problem of leg design.

3 The Leg Problem Design

In general, the robot leg design problem can be thought as the search of a leg mechanism that exhibits a particular dynamic behavior. Our particular leg design problem is, given a leg mechanism, to find a set of dimensions that minimizes the induced torques in the leg motors.

3.1 Leg Mechanism Description

The leg mechanism is composed by 5 links, as shown in Fig. 2. The motors that give motion to the leg are attached to the robot body and their axles support the leg mechanism. During the support phase, the contact point (point D in Fig. 2) moves parallel to the ground accordingly with the angular position of the motors.

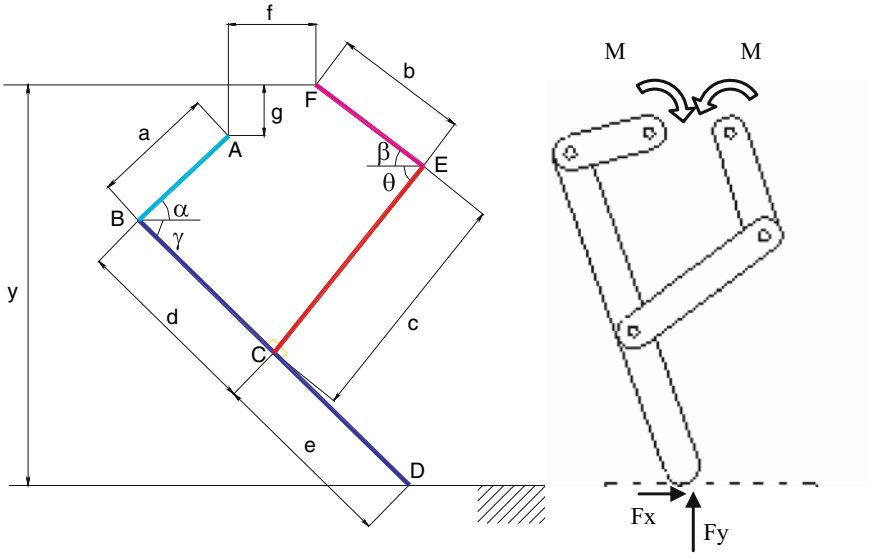


Fig. 2. Mechanism simplification with general nomenclature used for geometrical and mathematical purposes, and original mechanism. Coordinates origin at F point

3.2 Calculating Induced Torques

The first step is to calculate contact forces. These forces are important to determine several internal forces in the leg mechanism. The contact forces are obtained experimentally through Fourier series [9] and its mathematical expression depends on basic walking parameters (3) and (4) – See Appendix for equation listing).

After contact forces are calculated, induced forces on leg links are solved according to geometry and equations found from geometric model shown in Fig. 2.

The correct order to evaluate induced forces on mechanism links is: first, calculate the step extreme points (5)–(14), and their angular configuration; then solve the real angular position of entire mechanism on each position along step (15) and (16) using D point position (see Fig. 2) and inverse kinematics solve α, and γ; finally C point position along step is used to solve β and θ (17) and (18). These angular results are important for later calculating torque equations.

A geometric constraint due to motors used, is that α and β must be con the range $[0, \pi]$.

Once angles along step are calculated, internal forces are calculated using superposition method – frame analysis – [2] because the leg mechanism has links and is determinated.

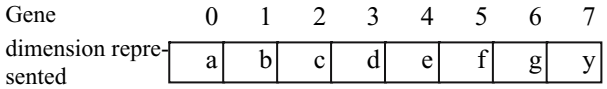


Fig. 3. Chromosome representation of leg dimension set

With internal forces already found, induced torques $M_1'(\alpha, \beta)$ and $M_2'(\alpha, \beta)$ in motors over step length are solved with (19 and 20). It is important to note that induced torques M_1' and M_2' depend on the current position of the leg (determined by (α and β angles).

Let M_1 and M_2 be the maximum values of M_1' and M_2' along step (as defined by 1, 2). M_1 and M_2 are the motor torques necessary for a self sustained walking. Overweight factor is included in initial robot mass estimation. Thus, the objective is to choose motors with a lower torque as possible (because this represent a weight save for robot design, energy consumption save on motor performance, and money save on motor buying) so M_1 and M_2 are the values to be minimized.

4 Proposed Approach

The problem to solve is to find a set of mechanism dimensions (a, b, c, d, e, f, g, and y) that minimizes the maximum induced torques (M_1 and M_2). The size of the search space is potentially infinite and the objective function is very complex; this makes impossible to apply a conventional optimization algorithm. In this work, a GA is proposed as an alternative to overcome these difficulties.

4.1 Representation

A dimensions set is represented by an array of real values. Thus, a chromosome (Fig. 3) is an array of real values, where each gene corresponds to the dimension of each link shown in Fig. 1.

Every individual in a population is a dimension set that defines a particular leg mechanism, and the population is a set of dimensions sets. The values in the chromosome are restricted by maximum and minimum dimensions, angle range along step, and minimum step length.

4.2 Fitness Evaluation

For fitness evaluation, induced torques M_1 and M_2 along step are used (1). This is a multiobjective optimization problem because both M_1 and M_2 should be minimized. The fitness of an individual is an ordered pair $(M_{\max}, M_{\min})$ where $M_{\max} = \max(M_1, M_2)$ and $M_{\min} = \min(M_1, M_2)$. Lexicographic order

is used to compare the fitness value of two individuals, that is, the best one is that with a minor $M_{\max}$. In case of tie of $M_{\max}$ value (two values are equal if differ in an amount less that an *epsilon*), $M_{\min}$ value determine the better option.

When torques M_1 and M_2 are calculated over all population individuals, the best individuals of each generation are saved in a historical list.

4.3 Genetic Operators

Individuals are selected using tournament selection. The size of the tournament is a parameter of the algorithm. Only one operator is applied each time. The operator to be applied is chosen using an uniformly generated random number and the probability assigned to each operator. This process is repeated until the new population is completed. The genetic operators used are:

- *Crossover*: two random point crossover.
- *Mutation*: Gaussian mutation.
- *Simple reproduction*: A copy of a single parent is made to the new population.

Elitism was used – the best individual of each generation is copied into the next generation.

5 Experimentation

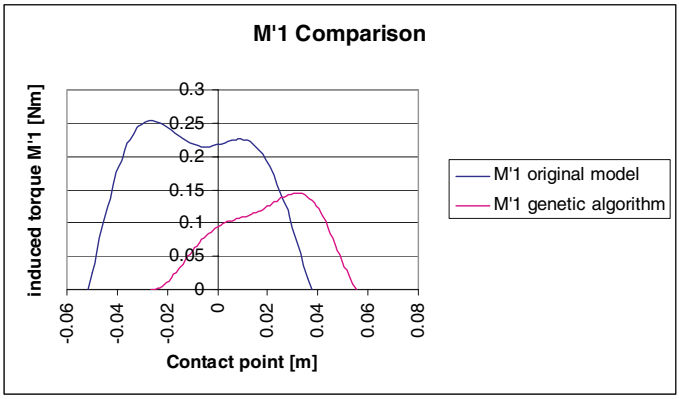
5.1 Experimental Setup

For every algorithm run, the following values were proved in some combinations for a total of 20 experiments:

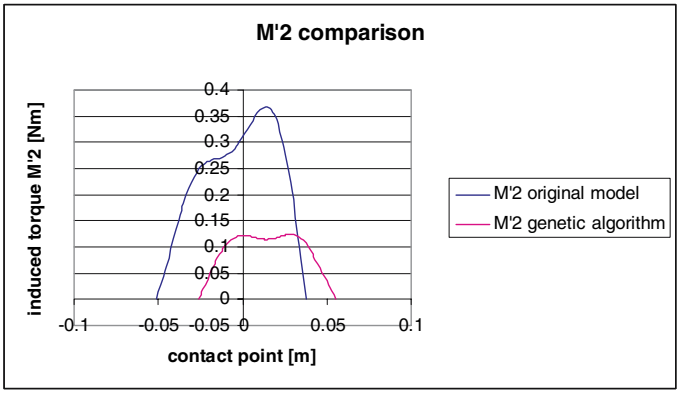
Generations	500, 200, 100, 50
Population size	500, 200, 100, 50, 20
Crossover probability	0.6
Mutation probability	0.05
Tournament size	7
Epsilon	0.01 Nm

5.2 Results

Figure 4 shows an example of a GA run, and it can be seen the best induced torques. Near to zero values are not taken into account because they mean false calculations – one or several parameters like total step length, forward and backward step length or angular range are violated. Fitness penalization takes this failure into account.



(a)



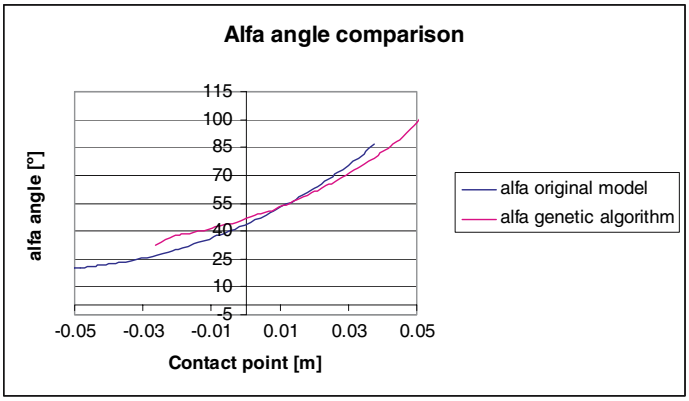
(b)

Fig. 4. Comparison of induced torques along step between results from original model and from genetic algorithm (**a**) M'1 torque, (**b**) M'2 torque

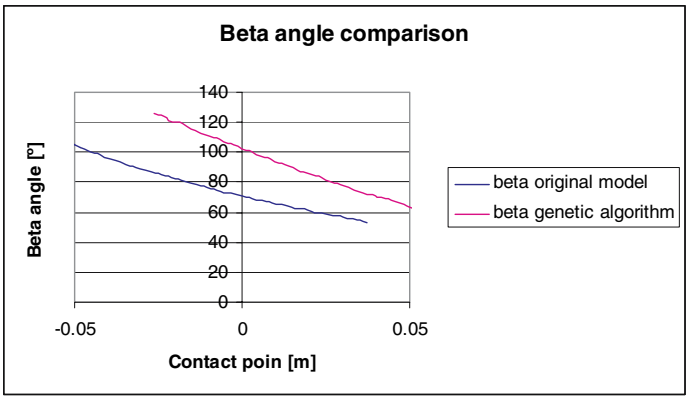
In Fig. 4 can be seen the advantage of results obtained by GA application. Original M_1' data is higher than M_1' values found by GA (Fig. 4a). This reduction is important because a lighter motor can be used. Figure 4b shows the improvement in M_2' data obtained by using the genetic algorithm.

It can also be seen that good solutions are not similar among them. The particular cases in Figs. 4, 5 show a slightly different step long and different extreme step points (begin and end of the step). These trade-off results must be considered by personal evaluation done by the designer. Only the expert can finally decide which solution will be implemented in the physical robot.

Figure 5 shows the comparison of α and β angles. Angles curves with fewer slopes are better because they represent a minor angular variation of the motor axle. This angular variation is also important because the fewer the slope, the slower the motor axle rotates. This fact is relevant given the fact that motors have a top angular speed, so genetic algorithm gives better movement options by giving slower motor axles rotation.



(a)



(b)

Fig. 5. Comparison of motor angles along step between results from original model and from genetic algorithm (**a**) α angle, (**b**) β angle

Even when it seems small improvements in angles curves, actually they are good enough to use the genetic algorithm result. It curves are more straight, so linearity of angular motion is also desirable.

6 Conclusions

The results obtained from experimentation show that evolutionary algorithms can be used to obtain good results for complex mechanical design problems, which surpass by far those obtained using classical mechanical engineering design techniques in this concrete hardware design problem.

It is important to highlight the complexity of the problem; its non-linearity, and the high dimension of the search space make it impossible to solve (and visualize good solutions) using conventional optimization techniques. The main contribution of the present work is to show that genetic algorithms are a good

alternative to solve this problem. However, it is necessary to do further experimentation that allows the assessing of the real strength of the technique and the definition of criteria to fine tune the parameters of the algorithm. Also, it is important to compare it against other heuristic techniques

Future research will be focused on different weighted fitness evaluation, and the use of genetic algorithms for mechanism structure optimization using deformation energy method. Experimentation may be focused on increasing population size, varying tournament size, and enhancement of non-linear equations system solution.

References

1. Reyes C., Padilla R. (2003) Design and construction of a walking robot prototype. Mechanical Engineering Final Project. Universidad Nacional de Colombia (in spanish).
2. Meriam J., Kraige L. (1993) Engineering Mechanics: Statics. 3rd ed. SI version. John Wiley & Sons Inc. pp. 208–209.
3. Norton R. (1998) Machine design: an integrated approach. Prentice-Hall.
4. McNally P. (2004) Comparison of gradient search with genetic algorithms for nose landing gear. ADAMS, Mechanical Dynamics. Last visited: 2004.03.30 [http://www.mscsoftware.com/support/library/conf/adams/euro/1999/Wednesday _papers/Parallel%20Papers/Design%20Optimization%20Technique1.pdf].
5. Motoyama K., Yamanaka T., McNally P. (2004) Design optimization technique: Suspension Design case study. ADAMS, Mechanical Dynamics. Last visited 2004.03.30: [http://www.mscsoftware.com/support/library/conf/adams/euro/1999/Wednesday-papers/Parallel% 20Papers/Design%20Optimization% 20Technique1.pdf].
6. Ortmann M., Weber W. (2001) Multi-Criterion Optimization of Robot Trajectories with Evolutionary Strategies. In: Proceedings of the 2001 Genetic and Evolutionary Computation Conference. Late-Breaking Papers, pp. 310–316, San Francisco, California, July 2001.
7. Osyczka, Krenich S., Karas K. (1999) Optimum Design of Robot Grippers using Genetic Algorithms. In: Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization (WCSMO), Buffalo, New York, May 1999.
8. Coello C., Christiansen A., Hernández A. (1995) Multiobjective Design Optimization of Counterweight Balancing of a Robot Arm using Genetic Algorithms. In: Proceedings of the Seventh International Conference on Tools with Artificial Intelligence, pp. 20–23, Herndon, Virginia, U.S.A., November 1995. IEEE Computer Society Press.
9. SOCIETY FOR EXPERIMENTAL BIOLOGY – SEMINAR SERIES. Aspects of animal movement. Cambridge: Elder, H.Y. Cambridge University Press. 1980. pp. 221–233.
10. Takahashi S., Obayashi S., Nakahashi K. (1998) Inverse Optimization of Transonic Wing Shape for Mid-Size Regional Aircraft. AIAA Paper 98–0601, January 1998.

Appendix: Equations

$$M_1 = \max_{\alpha,\beta \in [0,\pi]} \left(M_1'(\alpha,\beta)\right) \tag{1}$$

$$M_2 = \max_{\alpha,\beta \in [0,\pi]} \left(M_2'(\alpha,\beta)\right) \tag{2}$$

Maximum torque values along step in each motor.

$$F_Y = \frac{3\pi mg}{4\xi(3+q)}\left[\cos\left(\frac{\pi f t}{\xi}\right) - q\cos\left(\frac{3\pi f t}{\xi}\right)\right] \tag{3}$$

$$F_X = \frac{\nu t F_Y}{k} \tag{4}$$

Vertical (F_Y) and horizontal (F_X) contact forces, where ξ: duty factor, f: step frequency, m: robot mass, g: gravity, q: walking factor, v: walking speed, k: constant related with hip height respect to ground. Adapted from [8].

$$(d+e)\sin(\gamma_{P_t}) + a\sin(\alpha_{P_t}) + g = y \tag{5}$$

$$a\cos(\alpha_{P_t}) + f - (d+e)\cos(\gamma_{P_t}) = P_t \tag{6}$$

$$(b+c)\cos(\theta_{P_t}) - e\cos(\gamma_{P_t}) = P_t \tag{7}$$

$$(b+c)\sin(\theta_{P_t}) - e\sin(\gamma_{P_t}) = y \tag{8}$$

Expressions for some hind extreme points of step, where $\mathrm{P_t}$: maximum hind distance of contact point of the mechanism (foot), $\alpha_{\mathrm{Pt}} : \alpha$ angle for $\mathrm{P_t}, \gamma_{\mathrm{Pt}} : \gamma$ angle for $\mathrm{P_t}, \theta_{Pt} : \theta$ angle for $\mathrm{P_t}$

$$P_d = \sqrt{(a+d+e)^2 - (y-g)^2} - f \tag{9}$$

$$\beta_{P_t} = 180 - \theta_{P_t} \tag{10}$$

$$\gamma_{P_d} = a\tan\left(\frac{y-g}{P_d+f}\right) \tag{11}$$

$$\alpha_{eP_d} = 180 - \gamma_{P_d} \tag{12}$$

$$e\sin(\gamma_{P_d}) + c\sin(\theta_{P_d}) + b\sin(\beta_{P_d}) = y \tag{13}$$

$$e\cos(\gamma_{P_d}) - c\cos(\theta_{P_d}) + b\cos(\beta_{P_d}) = P_d \tag{14}$$

Expressions for some fore and hind extreme point of step, where $\mathrm{P_d}$: maximum fore distance of contact point oh the mechanism (foot), $\alpha_{\mathrm{Pd}} : \alpha$ angle for $\mathrm{P_d}, \gamma_{\mathrm{Pd}} : \gamma$ angle for $\mathrm{P_d}, \theta_{\mathrm{Pd}} : \theta$ angle for $\mathrm{P_d}, \beta_{\mathrm{Pd}} : \beta$ angle for $\mathrm{P_d}, \beta_{\mathrm{Pt}} : \beta$ angle for $\mathrm{P_t}$

$$Y_D = g + a\sin(\alpha) + (d+e)\sin(\gamma) \tag{15}$$

$$X_D = -f - a\cos(\alpha) + (d+e)\cos(\gamma) \tag{16}$$

Expressions for α, and γ angles along step.

$$X_C = b\cos(\beta) - c\cos(\theta) \tag{17}$$

$$Y_C = b\sin(\beta) + c\sin(\theta) \tag{18}$$

Expressions for β, and θ angles along step.

$$M_1' = R_{BY'}a\cos(\alpha) + R_{BX'}a\sin(\alpha) - W_1\left(\frac{a}{2}\right)\cos(\alpha) \tag{19}$$

$$M_2' = R_{EY''}b\cos(\beta) + R_{CX'}b\sin(\beta) - W_4\left(\frac{b}{2}\right)\cos(\beta) \tag{20}$$

Expressions for torques M_1', and M_2' induced on leg motors. Expressions are in terms of internal forces and mechanism dimensions.

Design Toolset for Realising Robotic Systems

Yiannis Gatsoulis[1], Ioannis Chochlidakis[2], and Gurvinder S. Virk[3]

Intelligent Systems Group, School of Mechanical Engineering, University of Leeds, Woodhouse Lane, Leeds, West Yorkshire, U.K. LS2 9JT
([1]menig, [2]men2ic, [3]g.s.virk)@leeds.ac.uk

Abstract. This paper describes the important criteria for selecting design software tools for realising specific robotic solutions quickly and efficiently. Possible quantitative and qualitative methods for the proposed selection framework are discussed. A case study on formulating a design toolset for the research and development of search and rescue robotic systems is presented.

1 Introduction

Research and development (R&D) of robotic systems is a complicated process and the produced systems are usually expensive. Robotic systems designers prefer to use a modular approach in order to reduce the R&D and maintenance costs as well as make their systems more flexible. However, these modules are task oriented and restrained within the scope of the particular platform. This kind of modularity is called *in–house modularity*. As a result research teams are wasting considerable amounts of time and resources in developing their own modules.

On the other hand if standard components were available in the market, developers would be able to build their prototypes faster as there will be a reduced need for individual development of all the components. Furthermore, the whole process is more cost effective due to mass production and reuse of the modules and the technologies for a wide range of applications. This kind of modularity is called *open modularity* [15, 16].

Figure 1 shows an approach to open modular R&D where the overall problem is broken down into different stages. Firstly, potential application areas and the operational environments should be identified. A generic list of task requirements is then formed for each. The general capabilities required by the robot are then formed, followed by a design of the specific capabilities. All components are grouped under four generic categories, namely input, processing, output and infrastructure modules. Super modules can be constructed from basic ones. All modules (basic and super) integrate with each other using

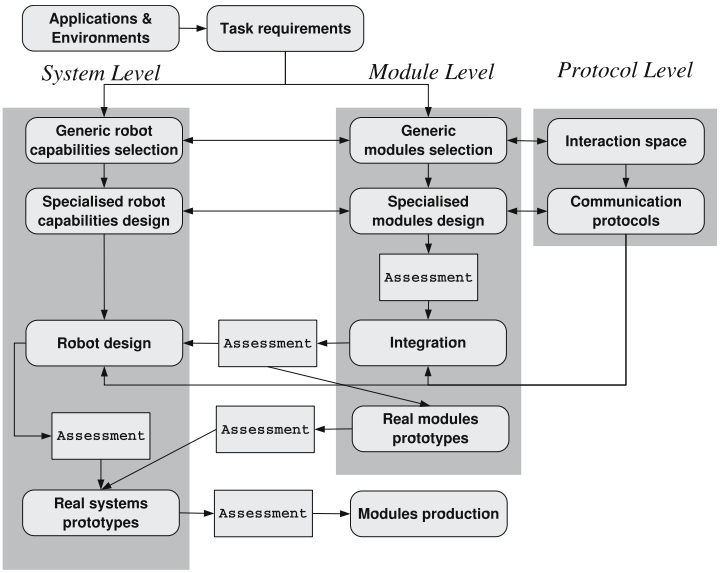


Fig. 1. Open modularity process

the interaction space as defined by the CLAWAR community [14]; this defines six ways in which interactions can take place, namely mechanics, power, digital, analogue, databus and environment. This methodology was inspired by Brooks' subsumption architecture [2] with emphasis given to the modular level [4]. Normally we would expect system level and module level designs to be carried out in a parallel way as shown in Fig. 1. Standard protocols and software tools must be developed for that. Assessments must be conducted at different levels of the R&D before any real components are built and the modules are finalised for mass production. This is achieved by extensive testing and refining of the solutions until all aspects are achieved.

However, the increasing use of design tools to assist the developers has resulted in a vast number of software packages. The selection of an appropriate tool is a difficult and time consuming task, especially if all the available packages have to be investigated to find the best one to use. This is the reason for the development of suitable simulation studies which try to set an evaluation framework for the selection of appropriate software tools according to the specific user requirements [11].

Several evaluation frameworks have been proposed. A hierarchical arrangement with high level characteristics that are decomposed into sub-characteristics and attributes is defined by the ISO/IEC 9126 standard [10]. The high level characteristics include reliability, usability, efficiency, maintainability and portability. An alternative hierarchical framework considers the user, the vendor, the model and the input, the execution, the animation, the testing and the efficiency and the output of the software as the highest levels in the

hierarchy [13]. Another one uses input, processing, output, support and cost as the generic categories [1]. Relative evaluation methods where the candidate software tools are compared in pairs with each other have also been used [5]. Furthermore, case studies have been produced for fields such as aerospace engineering [6], mail transfer issues [7] and structural engineering [12]. Expert systems that implement selection frameworks have also been written [9].

As mentioned previously building a robot prototype is a complex, time consuming and expensive task. Hence, tools are needed to simulate the electromechanical and behavioural aspects to assess the designs before they are actually built. The number of available software tools is vast and each has different capabilities and comes at a different cost, ranging from free to thousands of pounds.

This paper describes an evaluation framework that can aid the selection of appropriate software tools as well as formulate a toolset for robots to perform system level designs. The remainder of the paper is organised as follows. Section 2 describes the evaluation criteria, qualitative and quantitative methods for making the assessments and discusses the various different significances in the overall framework. Section 3 discusses whether the framework is consistent and reliable and a case study example illustrating the principles for a search and rescue robot is presented in Sect. 4.

2 Design Framework

In this section the assessment criteria of the software is presented, how the different elements can be measured and the effect of weighting in the evaluation.

2.1 Criteria

Cost: The cost of the simulator is the first criterion for its selection. It includes user training costs, maintenance costs, expenses for hardware requirements and development time costs.

Usability: This measures how well a design tool meets the users' requirements. It is one of the most important criteria as well as the most difficult one to measure. It depends upon the users' requirements and the task needed to be simulated. The difficulty in measuring it lies on the users' vague idea initially of what needs to be designed and assessed.

Expandability: This measures the likelihood and the time needed for the developers to improve the software, as well as if there are facilities that allow the users to expand it on their own and include their own modules.

Reusability: This measures if a software tool can be used for the design as well as the assessment of a model; if the produced models can be used by other software tools; and if the already written control programs can be reused within real robots.

Development time: This measures how fast new designs can be developed. Existence of predefined blocks and components as well as other tools can greatly speed up the implementation of it.

Efficiency: This measures the performance and other execution facilitation of the software. The performance is determined by the compilation and run time speed. Facilitation include speed control, off-line run, multiple and automatic batch run, reset capability, interaction, start in a non-empty state and debugging tools.

Visualisation: This measures how realistic the implemented designs and environments look like. For example if a chair looks like it is, or if it is represented just as a simple shape.

Portability: This considers if a software can be run in multiple operating systems. Depending on the user requirements this may be omitted.

User friendly: This measures how easy it is for the users to learn to use the software. Manuals, command references, general documentation, illustrative examples, training seminars and a user friendly interface are all features that help in learning.

Technical support: This measures how likely it is to get assistance. It includes technical assistant from the vendor or the developers, from help forums, FAQ lists and user groups.

Analysis facilitation: This measures if the software provides any facilitation for analysing and visualising the data such as business graphs and charts, structured output of the data or exportation into a spreadsheet, analysis functions and video capture or screenshots.

2.2 Qualitative and Quantitative Measurement Methods

Assessment criteria naturally need some kind of qualitative or quantitative measurement. One simple way is to strictly examine if the candidate software tools cover each of the subcriteria or not. The preferred simulator would be the one that covers most of them in as complete manner as possible. However, it is difficult to measure the criteria in all aspects as probably they would be partially covered. Even if we overcome this problem by skipping the high level layer and we consider only the lower one as our list of criteria, we still have to face the same dilemma when a simulator covers a subcriterion to some partial extent.

A more flexible approach is to measure the criteria quantitatively depending on the subcriteria. It can be done either descriptively or numerically. An example of this is a scale of possible values from 0–1, covering from "not at all" to "full", summed to give an overall criterion score, which in turn can be summed up for all the different criteria (with different weights if necessary as discussed in Sect. 2.3) to give the total assessment of the software.

All these are absolute evaluation methods. This means that each simulator is individually evaluated by assigning a value which signifies how well it meets the criteria. An alternative method is relative evaluation [5], where

each simulator is compared with every other simulator under consideration in pairs. Hence, a given simulator will have relative scores that signify how better or worse it is from each of its "competitors". This method has shown to give satisfactory results, although there have been sometimes some surprising rankings in the comparisons [5].

2.3 Weighting of the Criteria

In the methods described in Sect. 2.2 there has been no implication of whether the criteria have different importance. In fact it was assumed that they all have the same weight. Obviously, this is not rational as some criteria are more significant than others depending on the user requirements and priorities. Each *score* can be multiplied by a weight w signifying the contribution of a criterion in the overall assessment and each subcriterion can also have its own weight within the scope of higher level criterion. All scores can be normalised to values in the range 0–1. This is algebraically described by (1) and (2).

$$score_{tool} = \frac{\sum_{i=0}^{i=n}(w_i \times score_{crit_i})}{\max(score_{tool})} \tag{1}$$

$$\text{where, } score_{crit_i} = \frac{\sum_{j=0}^{j=n}(w_j \times score_{subcrit_j})}{\max(score_{crit_i})} \tag{2}$$

3 Discussion

A list of criteria that can be used as an evaluation framework for the formation of an appropriate design tool–based environment for robotic systems was described in Sect. 2. From the proposed software evaluation frameworks presented in Sect. 1 we could say that there is an attempt to categorise the large number of criteria in generic groups. Features can be easily added or removed from a hierarchical framework depending on the simulation domain and the user requirements. Furthermore, the features can be better visualised in a hierarchical organisation.

On the other hand, it can be seen that each study proposes its own hierarchical structure. Even if there are a lot of similarities especially in the low level attributes between the various hierarchical frameworks, this uniformity causes confusion. Moreover, hierarchical models tend to be static within their scope. This means that the features should be independent from each other, otherwise the hierarchical structure is compromised. One of the strengths of hierarchical frameworks is that they are flexible to minor changes. However, "major changes, although their occurrences are rare, may cause a re-organisation of the hierarchy" [13].

Our list of criteria is concentrated on the selection of robot system simulators. The main criteria include various subcriteria and this gives the impression that this framework has a strict hierarchy as well. However, in this

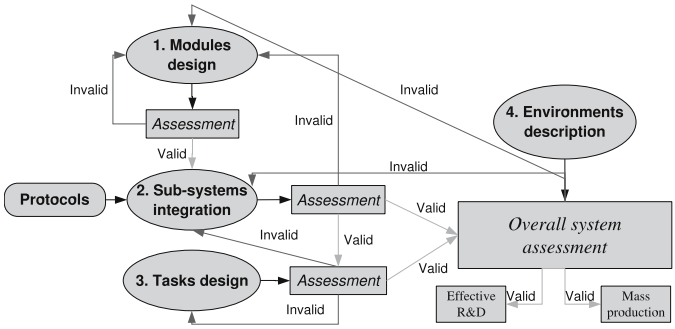


Fig. 2. Simulation cycle

structure generic criteria can be added or deleted according to the user requirements and sub-features can have multiple instances when required. Furthermore, the criteria can be affected by other ones outside their generic groups. The pay-off of a dynamic structure is its computational complexity when interactions between the criteria occur. It must be noted that this is rare as the generic criteria are distinct from each other.

4 Case Study: Urban Search and Rescue Applications

A design and assessment cycle is shown in Fig. 2. At the module level the basic and the super modules are designed and assessed both individually and together. At the system level the complete robotic systems are considered in their operational environments. Suitable design tools for the module level design are described in [4]. To highlight the system level design toolset, a specific case study on search and rescue robots is presented upon.

It is obvious that a system level simulator should provide the capability to implement a virtual environment realistic to the real one. The modularity concept adds to this list of requirements that the simulator should not be restricted to a particular robotic system or to a specific set of predefined modules, rather than it should allow the user to expand the library of modules and allow any desirable integration of them.

The operational environment in urban search and rescue (USAR) applications can be characterised as dynamic, hostile and rough. Entry points are often narrow and difficult to reach. The terrain the robots have to navigate can be extremely uneven, making even the simplest movements difficult without getting stuck. There is always the danger of a further collapse and the light conditions are normally very poor. Even worse, due to the complete disorder of the environment the readings of individual sensors can be noisy and unreliable. Robots have been used for the first time in the World Trade Centre disaster and very useful information has been collected [3]. However, USAR robots are

far from achieving their missions mainly due to their incapability of traversing through large piles of debris that are often found in urban disaster sites, teleoperation difficulties and communication losses in these environments and lack of reliable robot autonomy. Therefore researchers frequently simplify the operational environment to partially collapsed and unstable structures for investigation to be allowed.

A semi-autonomous USAR robot in its minimal form must carry a video camera, be able to be teleoperated through partially collapsed buildings and transmit the video either wirelessly or through a communication cable back to the operator. The following additional modules are also useful to have:

- Input modules: thermal camera, directional microphones, laser range finder, etc.
- Processing modules: hardware such as better microcontrollers, more memory, etc. and software modules like internal monitoring, auto navigation to a given position, recovery from communication dropouts, etc.
- Output modules: grippers, drills, suction pipe, etc.
- Infrastructure modules: more power, more powerful motors, better locomotion system, waterproof structure, etc.

More details about these as well as a more detailed presentation of a modular design of a USAR robot is found in [8].

About 150 software tools that can be used in the R&D design of robotic systems have been identified. They are classified in various categories such as graphics-environment modelling, image processing, programming libraries, physics libraries, planners, crossover tools, robot control libraries, robot dynamics and statics and system simulators as defined in our context. About 40 of them can be considered as system simulators where a robot can be simulated in a virtual environment. In order to make an evaluation of these, the assessment framework described in Sect. 2 was used.

The most important criteria from the requirements of the software tools are usability, reusability and expandability. Cost and technical support are also considered important. These five criteria contribute nearly 80% of the total evaluation. Table 1 shows the most suitable ones that cover to some extent the requirements described previously and Table 2 shows their scores based on our evaluations.

From all these the open source package of Player/Stage/Gazebo (PSG) seems to be the most appropriate. Stage is a 2D simulator capable of simulating large numbers of robots. Gazebo is a full physics 3D simulator which is quite processing demanding allowing a small number of robots to be simulated simultaneously. Player is a robot interface providing a network interface to a variety of robot and sensor hardware. The server/client architecture allows the control programs to be written in any language. These can be used for simulating virtual robots in Stage, Gazebo as well as real ones with no or little modification. PSG can be extended as new interfaces can be written for the hardware modules. As far as technical support is concerned there is helpful

Table 1. Suitable system level simulators

Name	License	URL (http://)
Player/Stage/Gazebo	Open source	playerstage.sourceforge.net
Simulation Studio	Commercial	eyewyre.com/studio
Webots	Commercial	www.cyberbotics.com/products/webots
Easybot	Commercial	iwaps1.informatik.htw-dresden.de :8080/Robotics/Easybot
Dynamechs	Open source	sourceforge.net/projects/dynamechs
Missionlab	Open source	www.cc.gatech.edu/aimosaic/ robot-lab/research/MissionLab
Rossum's Playhouse	Open source	sourceforge.net/projects/rossum

Table 2. Scores of system level simulators in a range 1–5. The weight of each criterion is shown in the parenthesises.

Name	Overall Score	Usability (20%)	Reusability (15%)	Expand-ability (15%)	Cost Rest (20%)	Technical Support (10%)	(20%)
Player/Stage Gazebo	3.9	4.7	4.7	4.7	5.0	3.9	0.9
Simulation Studio	3.2	4.1	3.8	3.1	4.0	4.1	1.0
Webots	3.2	4.5	4.6	4.7	1.3	4.1	1.1
Easybot	3.0	3.8	3.4	3.8	4.0	2.0	0.7
Dynamechs	2.9	3.3	2.5	3.3	5.0	2.6	0.7
Missionlab	2.8	1.6	3.8	2.9	5.0	3.0	0.6
Rossum's Playhouse	2.0	1.2	1.3	1.3	5.0	3.1	0.5

documentation as well as an active email list. One of the disadvantages of PSG is that the learning time is much longer than for other tools.

From the rest of the freeware simulators the following are worth a mention:

- Dynamechs and its graphical front end RobotBuilder is a good 3D simulator in which both the environment and a robot system can be modelled. It also has a long learning time like PSG. It is used for simulation and not for controlling real robots.
- MissionLab is a set of tool that allows the simulation in 2D environments and control of a single or a group of robots.
- Rossum's Playhouse is a simple 2D simulator and although it has a score below the average (2.5), it is mentioned here as it is the first software release of the ROSSUM project which "will grow into a resource where developers can download reliable, highly capable modules to incorporate into their robots".

The following simulators are the commercial ones:

- Simulation Studio™ is a 3D interactive simulator which allows the control of simulated and real robots with a BASIC Stamp microcontroller. It can be extended with new modules provided by the developers.
- Webots™ is also a powerful simulator with similar characteristics to PSG with the only difference that it is more expensive than most of the others. Both simulated and real robots can be controlled in full physics 3D environments. It can also be extended with new user's modules.
- Easybot is also a commercial tool which depends on the cost of the used 3D modeller LightVision3D™. Unlike PSG and Webots, Easybot does not allow the control of real robots. Another drawback is the lack of a physics engine. The simulator is discontinued for the favour of JRoboSim which is the successor of Easybot.

For the requirements of a USAR robot simulation and those of modularity it seems that the most suitable tools are PSG and Webots. Both have capabilities to implement virtual USAR environments, modelling of various modules and systems and teleoperation of the semi-autonomous systems. The development time of Webots should be faster as it is seems to provide more graphical facilities for the implementation. Webots is a commercial package and in fact it is expensive compared to the free PSG. This is the main reason why PSG is preferred. Furthermore, both have interconnections with other third party software; Webots with Matlab™ and LabView™ which are also commercial, while PSG with the free SourceForge[a] projects MARIE and RobotFlow/FlowDesigner.

5 Conclusions

The design of modular robotic systems can be decomposed into system and module level designs which are expected to be carried out in parallel. Software tools are needed for these. However, the vast number of available packages makes the selection of appropriate ones difficult and time consuming.

This paper described a guidance evaluation framework for the selection of appropriate design software tools. It consists of high level criteria which can be decomposed into particular features/subcriteria. A software toolset focusing on the system level design is formalised with the aid of the evaluation framework. This was illustrated through a case study on search and rescue robot applications.

References

1. J. Banks. Selecting simulation software. In *Simulation Conference Proceedings*, pp. 15–20. IEEE, 8–11 Dec. 1991.

[a] http://sourceforge.net

2. R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, Mar. 1986.
3. J. Casper. Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. Master's thesis, Dept. of Computer Science and Engineering, University of South Florida, May 2002.
4. I. Chochlidakis, Y. Gatsoulis, and G.S. Virk. Open modular design for robotic systems. In *Proc. of CLAWAR 2004, 7th Int. Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. Springer, 22–24 Sep. 2004.
5. L. Davis and G. Williams. Evaluating and selecting simulation software using the analytic hierarchy process. *Integrated Manufacturing Systems*, 5(1):23–32, 1994.
6. D.S. Eccles. Building simulators for aerospace applications: processes, techniques, choices and pitfalls. In *Aerospace Conference Proceedings*, volume 1, pp. 517–527. IEEE, 18–25 March 2000.
7. X. Franch and J.P. Carvallo. A quality-model-based approach for describing and evaluating software packages. In *Joint Int. Conference on Requirements Engineering Proceedings*, pp. 104–111. IEEE, 2002.
8. Y. Gatsoulis, I. Chochlidakis, and G.S. Virk. A software framework for the design and support of mass market clawar machines. In *Proc. of IEEE Mechatronics and Robotics Int. Conference*, 13–15 Sep. 2004.
9. V. Hlupic and A.S. Mann. Simselect: a system for simulation software selection. In *Simulation Software Proceedings*, pp. 720–727. IEEE, 1995.
10. ISO/IEC. Standards 9126 (information technology – software product evaluation – quality characteristics and guidelines for their use), 1991.
11. A.M. Law. How to conduct a successful simulation study. In *Winter Simulation Conference*, volume 1, pp. 66–70. IEEE, 7–10 Dec. 2003.
12. N.A. Maiden and C. Ncube. Acquiring COTS software selection requirements. *IEEE Software*, 15(2):46–56, March–April 1998.
13. J. Nikoukaran, J. Hlupic, and R.J. Paul. Criteria for simulation software evaluation. In *Simulation Conference Proceedings*, pp. 399–406, 1998.
14. G.S. Virk. Technical Task 1: Modularity for CLAWAR machines – specifications and possible solutions. In G.S. Virk, M. Randall, and D. Howard, editors, *2nd Int. Conference on Climbing and Walking Robots*, 1999.
15. G.S. Virk. CLAWAR: Modular robots for the future. In *Proc. of the 3rd Int. Workshop on Robot Motion and Control, RoMoCo02*, pp. 73–76. IEEE, 9–11 Nov. 2002.
16. G.S. Virk. CLAWAR modularity for robotic systems. *The International Journal of Robotics Research*, 22(3–4):265–277, March–April 2003.

Design, Dynamic Simulation and Experimental Tests of Leg Mechanism and Driving System for a Hexapod Walking Robot

J. Roca, M. Nogués, and S. Cardona

[a] Dept. d'Informàtica i Enginyeria Industrial, Universitat de Lleida, Spain
[b] Dept. d'Enginyeria Mecànica, Universitat Politècnica de Catalunya, Spain

Abstract. In this paper a design and a simulation of a leg mechanism for a hexapod is presented. The driving system for each joint consists of a DC motor, a timing belt and an Harmonic Drive gear.

In order to evaluate the energy consumption of the mechanism, the driving system has been experimentally tested and a friction model has been developed. Finally this model is used to simulate the actuator when driving the leg motion in order to get an accurate estimation of the motor torque and power requirements.

1 Introduction

High power consumption and reduced autonomy of operation are main drawbacks of walking machines, even in the most developed and modern walking robots. The reasons are that walking locomotion needs a lot of independent and highly controllable actuators, leading to a high mass of leg mechanisms and driving system, and poor energy efficiency of the latter due to continuously variable working conditions.

Interesting contributions focused on leg configuration and gait planning to get the minimum mechanical energy consumption at the powered joints [1, 2]. A second and necessary approach is an accurate study of the driving system itself, and the evaluation of energy consumption as a result of the driving output requirements, the friction in the transmission and the motor efficiency [3, 4].

This paper presents a leg mechanism for a hexapod and is focused in the joint and driving system design, which consists of a permanent magnet DC motor and a high performance mechanical transmission to fit joint output requirements. Cinematic and dynamic simulation of the mechanism is then computed to determine joint velocities and torques during a walking cycle.

Table 1. Specifications of robot and legs

Body length	700 mm
Robot width	520 mm
Femur length	180 mm
Tibia length	340 mm
Longitudinal foot stroke	240 mm
Vertical foot stroke	75 mm
Walking speed	9,6 m/min
Total robot mass	40 kg

A non linear model is proposed to describe the internal friction in the transmission assembly. Experimental tests have been carried out to determine the friction parameters in the transmission model. Non linear optimisation is then used to fit the model parameters to the experimental results. Finally, leg mechanism and the three joint actuators are simulated during a walking cycle and using the transmission model. One of the results is the mechanical power at the motor shafts, in comparison with the required mechanical power at the actuated joint in the leg mechanism.

Robot configuration considered for this study is a hexapod, with insect-like legs. The size is decided to be able to operate in a human environment (see Table 1) and the total mass is estimated including control hardware and batteries.

2 Leg Mechanism Design

Leg mechanism has been designed and modeled using the Pro/Engineer 3D-CAD software (see Fig. 1). Every joint and its corresponding actuator is implemented by an *Harmonic Drive* (*HD*) gear [5] and a pair of angular contact ball bearings (see Fig. 2). Low weight and high efficiency are the most important advantages of the *HD* gear for joint actuation in a walking robot. Zero backlash and low stick-slip effect are also convenient for leg movement control.

Required high transmission ratio is complemented with a timing belt between the motor and the gear, allowing a better location of the main components, the motor and the gear, one parallel to the other. A permanent magnet DC motor from the MAXON RE series has been chosen for this application, because of its high performance and easy to implement control.

3 Dynamic Analysis by Mechanism Simulation

The parametric CAD software has also been used to dynamically simulated the leg mechanism, allowing a fast iteration process of simulation and design optimisation. Operating conditions in this study are walking on a horizontal

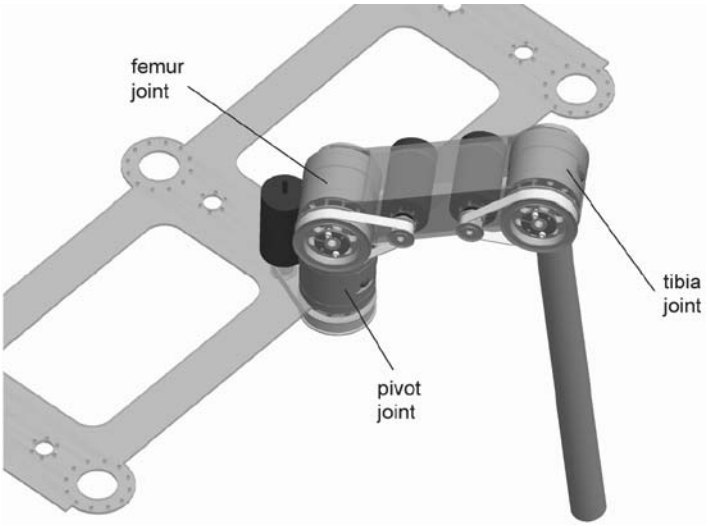


Fig. 1. Middle-right leg 3D model

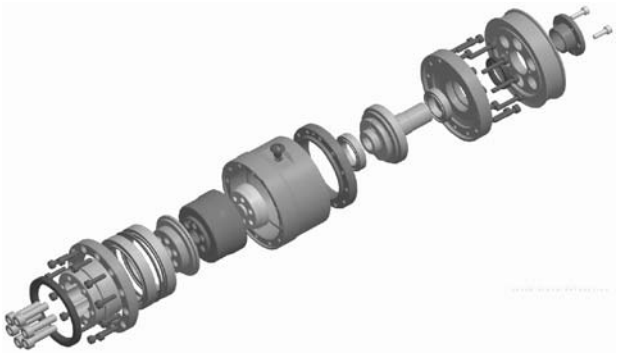


Fig. 2. Exploded view of joint and *HD*-gear assembly

ground and following an alternating tripod gait. Walking speed and foot stroke are shown in Table 1.

Ground reaction on the foot is supposed to increase and decrease progressively at the beginning and at the end of the support phase, and it is supposed that there are no transversal redundancies at the contact between feet and ground. No friction is considered at the joints in this simulation, since the bearings to implement the different joints are part of the actuators and will be considered later, when the driving system is analysed.

The objective of the simulation is to determine the required torque at every actuated joint due to the external actions on the mechanism and also due to the inertia of its components. In the simulation, motor shafts and transmission

components rotate at the corresponding speed, so the gyroscope effect due to their movement is also taken in account. Results are output torque at each joint (see in Fig. 3 the case of the middle-right leg) and instantaneous mechanical power to run them (see Fig. 4).

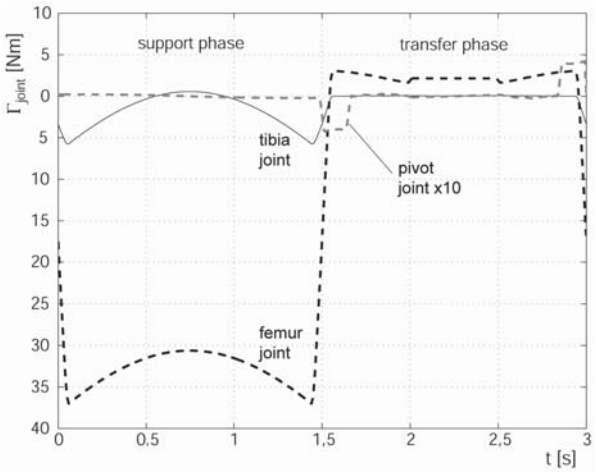


Fig. 3. Torque at each joint as a function of time during a walking cycle

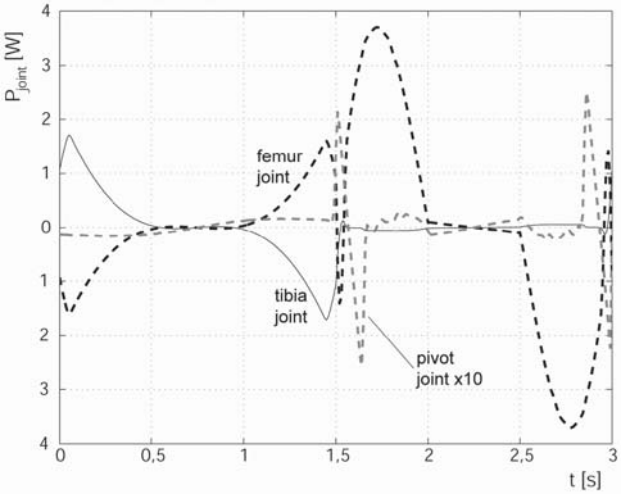


Fig. 4. Mechanical power to run each joint during a walking cycle

Table 2. Actuator components in the driving system

Component	Model	Ratio (i)
HD gear	HFUC–17	120 : 1
Timing belt	AT5	40 : 12
DC motor	MAXON RE-40	–

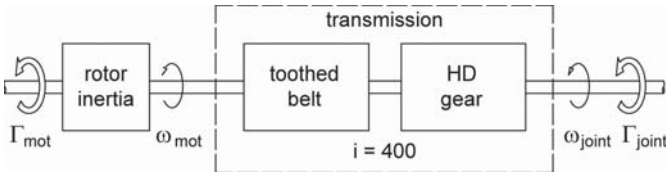


Fig. 5. Driving system layout for every joint

4 Actuator Model Including Friction

Driving system for every joint consists of a DC motor, a timing belt and an *HD* gear (see Table 2 and Fig. 5). A high transmission ratio is essential to fit output requirements, low speed and high torque needed for leg motion, to DC motor performance. So, from the dynamic and energetic points of view, it is necessary to study the motion resistance of the different parts in the transmission.

Resistance to motion in the *HD* assembly is a result of friction between moving components, at the gear teeth contact, the bearings and the shaft sealing rings, and the internal friction in the flexible parts, that is the hysteresis in the flexspline component. For the timing belt it's also a result of friction between moving components, at the bearings and at the belt-pulley contact areas, and the internal friction of the flexible component, the belt.

From an energy balance of the actuator, it is derived that motor torque (Γ_{mot}) has to overcome the output torque required to run the joint (Γ_{joint}), the transmission friction torque (Γ_{fr}, representing the equivalent friction torque at the output shaft), and the inertia of rotating parts in the motor and transmission (J_{in}) (see (1)). In order to model the motion resistance, it is considered that resulting friction torque can be decomposed in no-load friction, torque dependent friction and speed dependent friction. It has been experimentally probed that speed dependent friction increases with the square root of joint velocity. The way in which friction affects required input torque depends on joint velocity direction, as reflected in (2).

$$\Gamma_{\text{mot}} = \frac{\Gamma_{\text{out}} + \Gamma_{\text{fr}}}{i} + J_{\text{in}} \cdot \alpha_{\text{mot}} \tag{1}$$

$$\Gamma_{\text{fr}} = \left(\Gamma_{\text{o}} + \mu \cdot |\Gamma_{\text{out}}| + c \cdot |\omega_{\text{out}}|^{0,5}\right) \cdot \text{sign}(\omega_{\text{out}}) \tag{2}$$

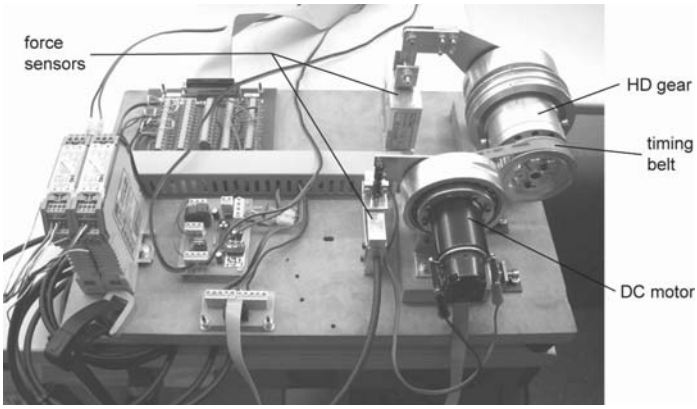


Fig. 6. Experimental setup to test the actuator

According to the Stribeck effect (see (3), friction coefficient μ in the torque dependent part is higher when joint motion is just started (static friction coef. μ_s) and rapidly decreases to a lower value (dynamic friction coef. μ_k) when joint velocity is slightly increased.

$$\mu = \mu_k + (\mu_s - \mu_k) \cdot e^{-|\omega_{out}/\omega_k|} \tag{3}$$

5 Determination of Parameters in the Model

Friction dependence on working conditions is then essential to understand transmission performance and to evaluate energy expenditure in this application. Most parameters in 2 are unknown and have to be experimentally evaluated. Figure 6 shows the experimental setup built up for this purpose. DC motor and gear housing are mounted on free rotating supports, using levers and force sensors to measure input and output torques.

A series of experimental tests have been carried out using the DC motor and the mechanical transmission to drive different output loads. Every result correspond to working conditions in steady-state operation, so actuator inertia doesn't affect the torque values. In Fig. 7 experimental results for input torque are represented corresponding to different output torques and varying output speed.

Parameters in the model presented in last section are identified by a non-linear least-square fit implemented by Matlab. Harmonic drive gear, timing belt and bearings assembly in every joint are then characterised by the parameters in Table 3. In Fig. 7 required input torque according to the described model is represented in comparison to the experimental data.

Table 3. Identified parameters in the transmission model

$[Nm]\ \Gamma_o$	μ_k	μ_s	ω_k [rad/s]	c
3,715	0,177	0,268	0,02	5,38

6 Actuator Simulation in Walking Operation

Once the friction model has been identified, it is feasible to simulate the performance of the whole joint actuator assembly when running the leg mechanism during a walking cycle. This simulation will then also consider friction and actuator inertia. Results are an accurate estimation of the motor torque and the instantaneous motor power required at each joint.

In Fig. 8 motor torque to run the femur joint is represented. In this graph it is shown the motor torque dependence on the actuator inertia and friction when they are considered or not.

Mechanical power to run the femur joint is represented in Fig. 9. Power at the motor shaft is much higher than the one at the joint when it is accelerated, and it is negative, no matter the output requirements, when the joint is sharply decelerated. It is clear then the influence of the actuator inertia.

Corresponding results for the tibia and the pivot joints have also been calculated. Finally, the integration of the power curves permits to evaluate the energy consumption per walking cycle of the leg mechanism. The results of such integrations are detailed in Table 4. The involved energy, at the joint

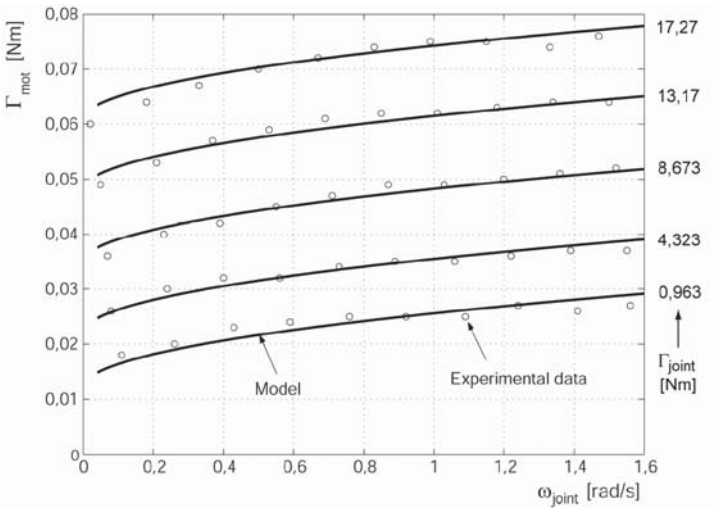


Fig. 7. Experimental data for different joint loads compared to the required motor torque according to the model

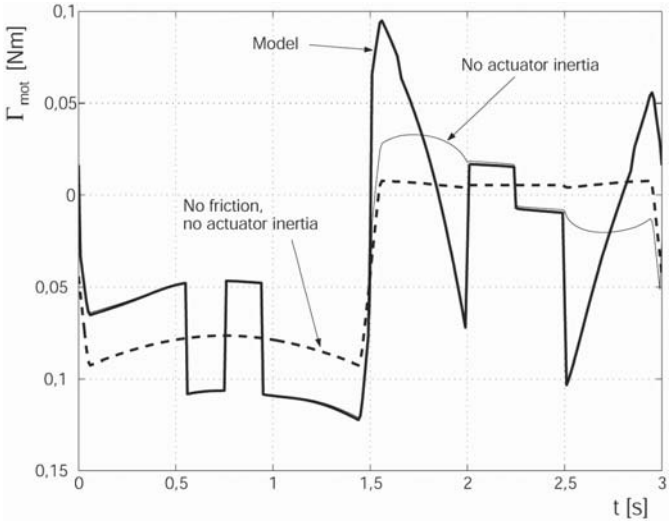


Fig. 8. Required motor torque at the femur joint during a walking cycle

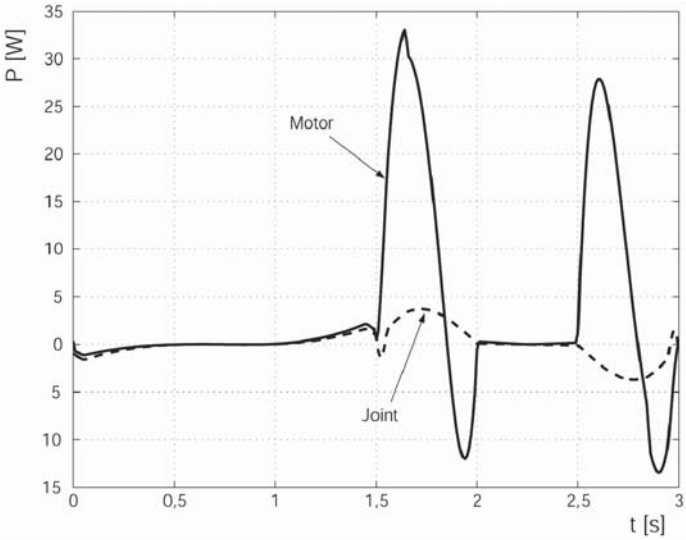


Fig. 9. Joint and motor mechanical power at the femur actuator

and at the motor, are split in two terms: the energy consumption to drive the output and the energy to be dissipated when braking or back-driving the output. Electrical energy to be supplied to the actuator (E_{el}) has been also computed from the model described in [4] and is included in Table 4.

Table 4. Total energy per cycle at the different joints

	Femur		Tibia		Pivot	
	Driving	Back-dr	Driving	Back-dr	Driving	Back-dr
E_{joint} [J]	1,56	–1,56	0,40	–0,40	0,036	–0,036
E_{mot} [J]	12,7	–3,05	13,1	–2,83	23,5	–1,11
E_{el} [J]	18,6	–2,24	15,9	–2,23	28,6	–0,88

7 Conclusions

A leg mechanism with three identical actuators – consisting of a DC motor, a timing belt and an *HD* gear – has been designed and simulated using a CAD software. A non-linear friction model of the joint and actuator assembly has been defined, which fits the data from experimental tests for different working conditions. This model is then included in the simulation of the mechanism during a walking cycle.

From this study it is clear that transmission friction and actuator inertia have a significant influence on the motor torque. Then, in some situations, required motor power is much higher that the power needed to run the joint. On the other hand, it is important to notice that electrical energy to be supplied to the DC motor is much more than the one expected from the joint requirements.

References

1. Marhefka D.W., Orin D.E. (1997) Gait planning for energy efficiency in walking machines. Proc. Int. Conf. on Robotics and Automation. IEEE, pp. 474–480 vol 1
2. Zielinska T., et al. (1999) Design of autonomous hexapod. Proc. of the 1st Int. Conf. on Robot Motion an Control. IEEE, pp. 65–69
3. Guardrabrazo Pedroche T.A., Jiménez Ruiz M.A., Gonzalez de Santos P. (2003) A detailed power consumption model for walking robots. Proc. of the 6th Int. Conf. on Climbing and Walking Robots, Catania, Italy, pp. 235–242
4. Roca J., Palacin J., Cardona S. (2003) Energy efficiency of a DC motor based driving system for leg movement of a hexapod walking robot. Proc. of the 6th Int. Conf. on Climbing and Walking Robots, Catania, Italy, pp. 845–852
5. Harmonic Drive A.G. (2003) Engineering Data for Harmonic Drive Gears. Precision in Motion, pp. 333–363

Limb-Mechanism Robot with Winch Mechanism

N. Fujiki[1], Y. Mae[2], T. Umetani[1], T. Arai[1], T. Takubo[1], and K. Inoue[1]

[1] Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, 1–3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan
[fujiki, umetani, arai, takubo, inoue]@arai-lab.sys.es.osaka-u.ac.jp

[2] Department of Human and Artificial Intelligence Systems, Faculty of Engineering, University of Fukui, 3-9-1 Bunkyou, Fukui, Fukui, 901-8507, Japan
mae@rc.his.fukui-u.ac.jp

The present paper describes a legged robot equipped with an omni-directional winch mechanism to climb up and down steep inclines. A small and omni-directional winch mechanism is developed for robots moving in rubble environments. The wire of the winch mechanism can pull a robot on steep inclines. The developed winch mechanism has a guide which can reel up and release a wire in all directions. A legged robot with the developed winch mechanism can move in all directions on steep inclines without changing the direction of the robot. In addition, we propose an idea of using one limb of the robot as a guide for the wire. The method improves flexibility of the robot moving on steep inclines. We show feasibility of our proposed idea through the experiments.

1 Introduction

Various kinds of working robots have been developed to do work instead of humans, especially in dangerous environments, such as disaster environments, deep sea, space, or inner nuclear plant. In these environments, the terrain that robots move on is not plane or regular. Most of the conventional working robots have wheel or crawler as their locomotion mechanisms, and it is difficult for them to move on irregular terrains or steep inclines [1]. On the other hand, a legged robot suits to work on an irregular terrain because it can select suitable landing point and keep a stability of robot's body during work. A walking robot which has six limbs that are used as both legs and arms has been developed [10]– [16]. We call this type robot "Limb-mechanism robot."

Pulling up and releasing a robot using a wire improves a mobility of the robot since the tension of the wire supports the weight of the robot. Since the

reactive force from the ground decreases when a robot is on steep inclines, it is difficult to support the weight of the robot in the rubble environment. Some researches suggest that a winch mechanism is fixed to the environment. This has a problem since the wire released from the winch may hook into the rubble environment and the robot cannot move any more. The winch mechanism should be mounted on the robot body.

The present paper describes a limb-mechanism robot mounted on a small and omni-directional winch mechanism on the body to climb up and down steep inclines. The robot can move omni-directionally without changing the direction of the robot. The proposed winch mechanism has a guide that rotates freely and can reel and release a wire omni-directionally. It supports the weight of the robot and help omni-directional locomotion of the robot on steep inclines. Furthermore, we propose the idea of using one limb as a guide of the wire. This method improves flexibility of the robot mobility on the inclines. We show feasibility of our proposed idea through the basic experiments.

2 Limb-Mechanism Robot

The outdoor robots require high manipulability and mobility in the application of construction, agriculture, and space or ocean developments [10–16] thus they should have both handling and mobile mechanism. Most of the conventional working robots have been designed in such a manner that a manipulator is simply mounted on a mobile platform [17, 18]. There are some animals or insects that can use their legs for their hands to manipulate objects dexterously while walking on a rough terrain. Based on this notion, the "limb mechanism" has been proposed that has an integrated locomotion and manipulation function into one limb.

In actual tasks it is more essential for a legged working robot to walk in any direction on rough terrain quickly and smoothly with keeping its stability as constant as possible. Thus the limb mechanism robot should be designed taking this omni-directional mobility into account. As one of feasible structures of the limb mechanism a 6-limb mechanism was analyzed and evaluated in the aspects of omni-directional mobility. This paper introduces the small winch mechanism for a limb-mechanism robot.

3 Steep Incline Locomotion of Limb-Mechanism Robot Using Winch

The limb-mechanism robot has omni-directional mobility. The robot, however, will not be able to move on a steep incline in a rubble environment, since the frictional force between the end of the limb and the ground cannot support the weight of the robot. The paper proposes a winch mechanism mounted on the robot and its application for walking on steep inclines.

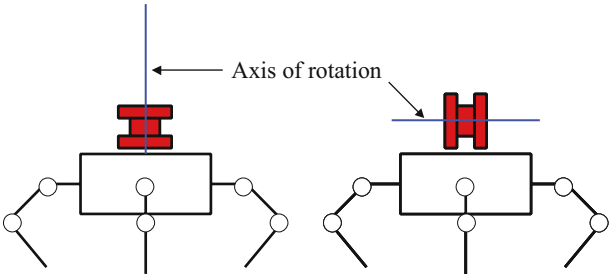


Fig. 1. Rotate axis of the winch

3.1 Omni-Directional Winch Mechanism

Most winch mechanisms are fixed to a suitable surrounding environment. Furthermore, almost all winch mechanisms can reel up and release a wire only in a fixed direction. Thus usual winch mechanisms disturb the omni-directional mobility of the robot. We discuss the conditions that the winch mechanism can reel up a wire omni-directionally and improve the robot mobility and manipulability with no deterioration in its omni-directionality. First we discuss the direction of winch mechanism as shown in Fig. 1. If the rotation axis of the winch mechanism is fixed horizontally, it is difficult that the winch mechanism reels up a wire omni-directionally. It is necessary to rotate the whole drum part of the winch including an actuator. Then the winch mechanism becomes complicated. On the other hand, if the rotation axis of the winch mechanism is fixed vertically, the winch mechanism can reel up a wire omni-directionally in the simplest way. We propose to equip the winch mechanism with a guide which can rotate around the vertical center axis. The winch mechanism can reel up a wire omni-directionally by setting a guide to the winch mechanism and rotating it. Thus the rotation axis of the winch mechanism is set vertically. Then the winch can reel up a wire omni-directionally.

Second, we discuss the number of connecting points in an object. When we manipulate an object with wire in general, we need more wires connected to the object in different points in order to fix stably. This is true when we support our robot with wire, however, we will need more winches mounted on the robot in this case. Furthermore, wires will interfere with the rubble environment and they will not work effectively. We propose that the limb-mechanism robot has a winch mechanism in the center of body. The winch mechanism reels up at one point in the robot body. Figure 2 shows the details of the winch mechanism. The robot is mounted on a small compact winch mechanism on the robot body. Using the proposed winch mechanism, the robot can move every direction even on an incline as well as on a flat. On a steep incline, however, a winch mechanism may not work efficiently. This is because that the guide rotates freely around the center of the winch and then the robot will rotate itself due to small friction in each supporting leg.

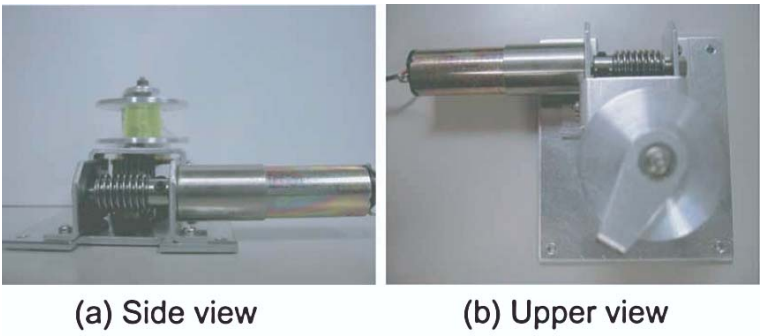


Fig. 2. The winch mechanism

3.2 Guiding Wire with Limb

The robot has six limbs. Each limb has both leg and arm function, and can realize manipulation and locomotion. Furthermore the robot can move even when some of its limbs work as arms. If the robot is on a steep, it cannot maintain sufficient reaction force from the ground and unstable. In the worst case, the robot falls down. In this case the limb will be used a guide for winch mechanism. It is possible to use a limb as a fixed guide by supporting a wire with the gripper. When one of limbs is used as a guide for wire, we can reel up the robot stably. If an incline is steep and the robot uses one of limbs as a guide for wire, it can walk with the wave gait. When an incline is not so steep, the robot can use whole limbs as legs. In this case, we can reel up the robot using the guide equipped with the winch. This winch can reel up a wire omni-directionally and reel up the robot stably even at the steep incline.

4 Experiment on Steep Incline Locomotion

The developed limb-mechanism robot is shown Fig. 3. The weighs is about 2.8 [kgf] and the full-length of the robot expanding the limbs is 720 [mm]. First we experimented with the new winch mechanism mounted on the limb-mechanism robot to reel up the robot at an incline. Without the reel up assistance by the winch mechanism, the robot could not walk any more when the slope angle is over 10 degrees. When the robot used the winch mechanism for assistance of its locomotion, it could walk on the slope of 30 degrees by the same gait on the plane.

Second, we tried the omni-directional motion of the winch mechanism mounted on the robot. The robot moved on the slope of 30 degrees. It shows the omni-direction of the winch mechanism.

Finally, the robot tried to go down a slope of 75 degrees by using one limb as a guide for wire Fig. 5. The robot could go down a steep incline by using

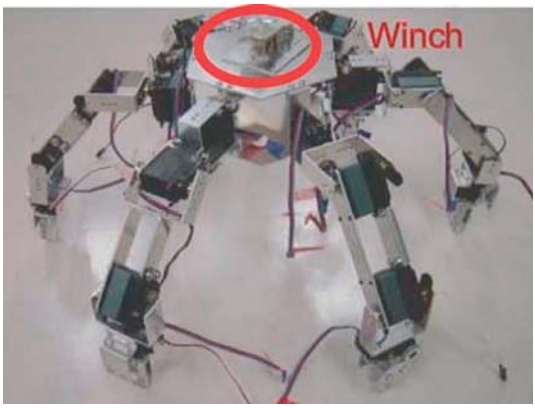


Fig. 3. Limb-mechanism robot with the winch

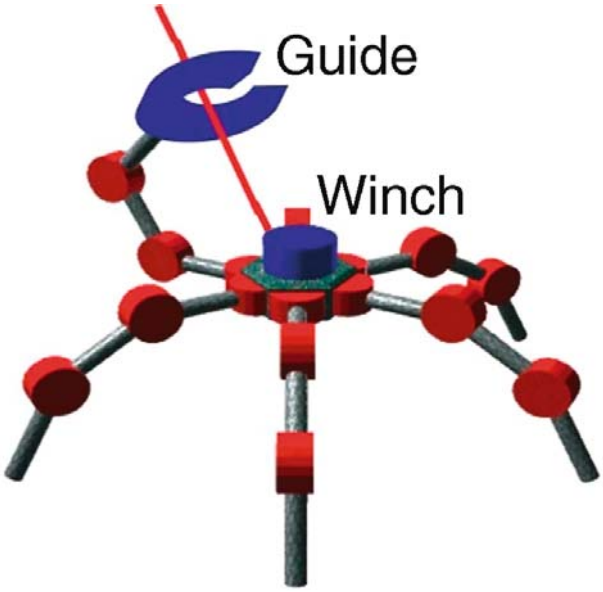


Fig. 4. The limb as a guide

its limb as a guide as shown in Fig. 6. This shows that the winch mechanism can improve the robot mobility on the slopes remarkably.

5 Conclusions

The paper describes a limb-mechanism robot mounted on a small winch mechanism. This winch mechanism has a guide which rotates freely and can reel

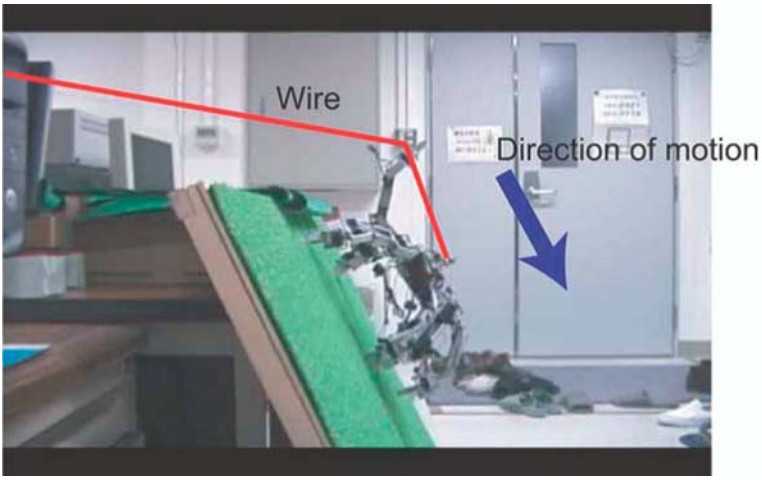


Fig. 5. Motion of the robot

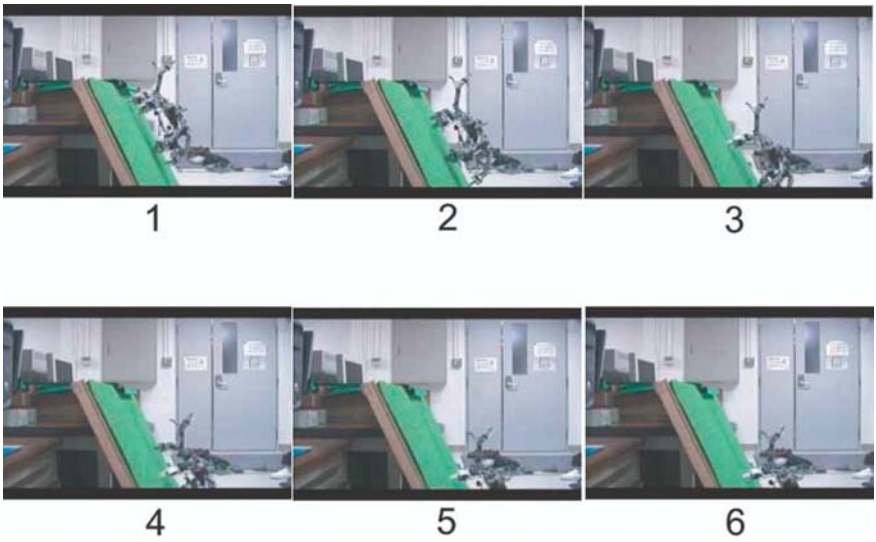


Fig. 6. Experiment of locomotion

and release a wire omni-directionally. Furthermore we propose an idea of using one limb as a guide for the wire. This method improves the robot mobility on the inclines. We have shown feasibility of our proposed idea through the basic experiments. Our final goal is its application to rescue tasks in rubble environments.

References

1. Atsushi Yamashita, Hajime Asama, Tamio Arai, Jun Ota, Toru Kaneko, "A Survey on Trends of Mobile Robot Mechanism", Journal of the Robotics Society of Japan, Vol. 21, No. 3, pp. 282–292, 2003
2. N Koyachi, T Arai, H Adachi, Y Itoh, "Integrated Limb Mechanism of Manipulation and Locomotion For Dismantling Robot – Basic concept for control and mechanism – ", Proceedings of the 1993 IEEE/Tsukuba International Workshop on Advanced Robotics, pp. 81–84, Tsukuba Japan, November 1993
3. N Koyachi, T Arai, H Adachi, "Hexapod with Integrated Mechanism of Leg and Arm", Proceedings of IEEE Conference on Robotics and Automation, pp. 1952–1957, May 1995
4. T Arai, N Koyachi, H Adachi, K Homma, "Integrated Arm and Leg Mechanism and its Kinematics Analysis", Proceedings of IEEE Conference on Robotics and Automation, pp. 994–999, May 1995
5. N Koyachi, T Arai, H Adachi, A Murakami, "Design and Control of Hexapod with Integrated Limb Mechanism: Melmantis", Proceedings of 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 877–882, Nov. 1996
6. J Racz, N Koyachi, T Arai, B Siemiaatkowska, "Melmantis – the Walking Manipulator", Proceeding of the 5th International Symposium on Intelligent Robotic Systems, pp. 23–29, July, 1997
7. Yuuya Takahashi, Tatsuo Arai, Yasushi Mae, Kenji Inoue, and Noriho Koyachi, "Development of Multi-Limb Robot with Omnidirectional Manipulability and Mobility", Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 877–882, Nov. 2000
8. Tatsuo Arai, Yuuya Takahashi, Yasushi Mae, Kenji Inoue, and Noriho Koyachi, "Omni-directional mobility of limb mechanism robot", Proceedings of 4th International Conference on Climbing and Walking Robots, pp. 635–642, 2001
9. Tatsuhi Mure, Kenji Inoue, Yasushi Mae, Tatsuo Arai, Noriho Koyachi, "Sensor-based walking of limb mechanism on rough terrain", Proceedings of 5th International Conference on Climbing and Walking Robots, pp. 479–486, 2002
10. G Pritschow, et al.,"Configurable Control System of a Mobile Robot for On-site Construction Masonry", Proc. 10th Intn. Sym. on Robotics and Automation in Construction, 1993, pp. 85–92.
11. E Papadopoulos and S. Dubowsky, "On the Nature of Control Algorithms for Free-Floating Space Manipulators", IEEE Trans. on Robotics and Automation, 7(6), 1991, pp. 759–770
12. E Nakano, T Arai, et al.,"First Approach to the Development of the Patient Care Robot", Proc. 11th Intn. Sym. on Industrial Robot, (Tokyo), 1981, pp. 87–94.
13. S Skaar, I Yalda-Mooshabad, et al., "Nonholonomic Camera-Space Manipulation", IEEE Trans. On Robotics and Automation, 8(4), 1992, pp. 464–479
14. Yuan F. Zheng, and Qichao Yin, "Coordinating Multi-limbed Robot for Generating Large Cartesian Force", Proceedings of IEEE Conference on Robotics and Automation, pp. 1653–1658, 1990
15. Chau Su and Yuan F. Zheng, "Task Decomposition for a Multi-limbed Robot to Work in Reachable But Unorientable Space", IEEE Transaction on Robotics and Automation, Vol. 7, No. 6, pp. 759–70, 1991

16. Anderw C. Zeik, "Motion Coordination for a Mobile Manipulator System", OSU master thesis, 1989
17. Sugiyama, et al., "Quadrupedal Locomotion Subsystem of Prototype Advanced Robot for Nuclear Power plant Facilities", Proceedings of Fifth International Conference on Advanced Robotics, pp. 326–333, June 1991
18. Hartikainen, Halme, Lehtinen, Koskinen, "Control and Software Structures of a Hydraulic Six-Legged Machine Designed for Locomotion in Natural Environments", Proceedings of IEEE/RSJ International Workshop on Inteligent Robots and Systems, pp. 590–596, 1996

Embodiment in Two Dimensions

Christian R. Linder

Department of Biological Cybernetics and Theoretical Biology, University of Bielefeld, Germany.
christian.linder@uni-bielefeld.de

Abstract. I propose an approach for the evolution of six-legged walking in two dimensions. This approach makes the evolution of higher behaviours for six legged walkers feasible. Preceding work in the evolution of control systems for moving artifacts usually emphasizes one of two aspects: It is either focussed on the control of a complex body, pursuing a simple task (e.g. the evolution of six-legged walking in full featured dynamic simulations) or on the control of a simple body, pursuing a more complex task (e.g. the evolution of path planning and orientation for wheeled robots). In this work, I introduce a simulation residing on an intermediate abstraction level, i.e. embodiment in two dimensions. Due to the computational efficiency of the applied kinematics, it allows the evolution of complex behaviours (like turning, orientation and obstacle avoidance) in simplified, but still adequately embodied artifacts. First results regarding the evolution of six-legged walking demonstrate the advantages of modular versus central controllers (for evolutionary approaches) and the optimal number of inter-leg connections in modular control systems.

1 Introduction

To identify the ultimate reason of behaviour displayed in animals and to distinguish indispensable constraints from phylogenetic dead-ends, it would be very helpful to study the outcome of alternative evolutionary runs (e.g. on different planets), compared to the one we can see all around us. Since this is impossible to accomplish, the discipline of embodied artificial life tries to breed "artificial" life forms in silica. The ultimate goal is thus the evolution of artificial, embodied agents under real physical constraints. The obtained principles can then help us to build physical robots and their controllers.

A long pursued goal is the production of walking machines with capabilities comparable to biological walkers. This paper is confined to entities with six legs, a body-architecture that gained considerable attention both from the field of robotics (for its static stability during walking) and the field of biology (for its ubiquity and surplus degrees of freedom). Biologists extensively studied

the behaviour and neurophysiology of e.g. stick insects [3] and cockroaches [5] to understand the mechanisms evolved to accomplish six legged walking.

Using the data of the stick insect *Carausius morosus*, a neural and distributed controller for six legged walking, the WALKNET, was developed [4]. So far, this controller is capable of producing straight walk in the presence of various distortions, and can overcome shallow obstacles. For this behaviour (and also the production of slightly curved paths), it is sufficient to place each tarsus close to its anterior neighbour in swing mode, a task accomplished by a module called TARGETNET.

The goal pursued with the approach presented here is the control of more complex stepping sequences with small, sideward and backward steps (as for example required for narrow curves [6], crossing a gap [1] or avoiding obstacles (Stuebner in prep.), which require extended control of the anterior and posterior extreme positions and extended mechanisms for action selection.

As a preliminary experiment, I compare two esentially different approaches to the control of walking: The central and the modular approach. While most robotic implementations use the central approach so far, biological evolution in insects seems to have chosen the modular approach: There are small brain-like subsystems (ganglions) at each leg's base, responsible for the control of this leg. I use 2D models of an insect, controlled by one-layered neural networks with feedback connections, to compare both approaches with respect to their evolvability.

A second experiment deals with the optimal number of inter-leg connections in a modular controller. While the WALKNET, due to observations in the stick insect, comprises 7 distinct coordination mechanisms, other approaches suggest that one or few intersegmental connections might be technically sufficient (Roggendorf in prep., see also [9]).

2 Methods

To evolve controllers for embodied artifacts, one ideal is a dynamic simulation exactly modelling real world physics. Another ideal is an open ended evolution, achieveable through complex interaction and coevolution of a very high number of artifacts. So far, every experiment in artificial life holds a heavy trade-off between these two extremes. The work of Sims, Lipson & Pollack, Komosinski & Ulatowski and Bongard & Pfeifer [2, 7, 8, 10] focusses on the former ideal, with impressive results. But although all these groups tackle the problem with massively parallel computing, complexity and higher functions evolve very slowly. On the other hand, many advances have been made in the control of higher functions in punctiform entities (i.e. without simulation of a body), that could be transferred to wheeled robots.

Here, I try to close the gap between both approaches. Desired is a model of an articulated body that preserves essential peculiarities of six-legged locomotion, but is computationally cheap enough to simulate high numbers of

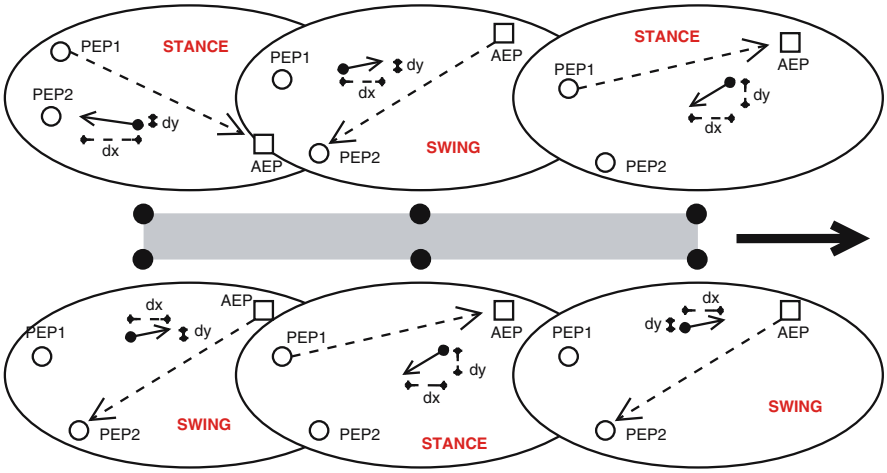


Fig. 1. The two-dimensional projection of a six-legged walker. Leg endpoints swing from a posterior extreme position 1 (PEP1) to an anterior extreme position (AEP). During stance, they push the body forward while moving from AEP to PEP2. The neural controller can either control the movement of the tibia endpoint (dx and dy) and toggle between stance and swing (3 outputs per leg), or place the AEPs and PEPs and control the speed at which the tibia endpoint is moved back and forth (5 outputs per leg)

individuals with complex neural controllers on standard hardware. Figure 1 shows such a model: The body of a stick insect is modelled as a rectangle on a two-dimensional plain. To represent a single leg, only the two-dimensional projection of the tibia endpoint and the swing/stance information are provided. The neural controller can either control the movement of the tibia endpoint and toggle between stance and swing (3 outputs per leg), or place the AEPs and PEPs and control the speed at which the tibia endpoint is moved back and forth (5 outputs per leg).

2.1 Kinematics of a Six-Legged Walker in 2 Dimensions

The movement of the endpoint of a leg touching the ground is interpreted as an inverse movement of its respective mounting point (Fig. 2, left side). The movements of several legs therefore generate possibly contradicting movements of different points on a rigid body. This conflict is resolved by splitting up n contradicting movements of leg mounting points into n! consistent hypthetical movements (Fig. 2, right side) and by then calculating the mean over all these hypothetical movements. First, for every pair of mounting points Ci and Cj, the rigid body is shifted along mi. Then, the body is rotated around Ci' in such a way, that the distance between Cj' and the desired point Cj + mj is minimized. For n legs touching the ground, this results in n! values for

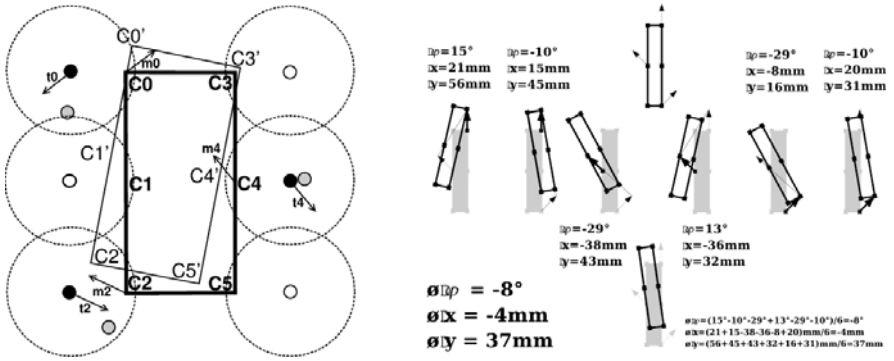


Fig. 2. The movement of the endpoint of a leg touching the ground is interpreted as an inverse movement of its respective mounting point (*left*). The movements of several legs therefore generate possibly contradicting movements of different points on a rigid body. *Right side*: The n contradicting movements of leg mounting points are split up into n! consistent hypthetical movements (second row). The actual movement is then computed as the mean of the hypothetical movements (*bottom right*)

desired translation and n! values for desired rotation, which are then averaged (Fig. 2).

2.2 First Experiments

In the following, neural networks are implemented as one-layered perceptrons with feedback-connections. Initial tarsus positions are randomized over 50% of their action range, and a traditional genetic algorithm with roulette wheel selection, p(mutation) = 0.01, p(crossover) = 0.9 and p(replication) = 1 is used. Since the number of involved computers and therefore the effective population size varies between experiments, speed of evolution can only be compared within experiments, not between experiments.

Experiment 1: Central Versus Modular Approach

Three controllers are compared: Controller M is composed of 6 identical modules, each having 12 inputs and 10 outputs, controllers C1 and C2 consist of one central neural network with 26 inputs (C2: 41 inputs) and 21 outputs. C1 and C2 differ in their number of feedback connections: 3 (as in one module of controller M) for C1, and 18 (as in all modules of controller M together) for C2. The fitness function is defined as

$$xt + 0.01 \times \sum s(t), \tag{1}$$

with xt being the distance travelled at time t and $s(t)$ describing the stability of the walker at time t.

Experiment 2: The Optimal Number of Inter-Leg Connections

Three slightly different modular controllers similar to the ones described in the previous section are employed to find the optimal number of inter-leg connections. The modular controllers M0-M4 incorporate zero, one, two, three, and four mutual connections between neighboring legs.

Experiment 3: The Optimal Number of Feedback Connections

The evolvability of four modular controllers is tested for its dependency on the amount of feedback connections within each module. To avoid the evolution of oscillatory pathways in the inter-leg connectivity, only two one-way connections were retained for each leg, connecting two of its output neurons with two input neurons of its posterior neighbor. The feedback connections within a module are realized as n additional output neurons that are connected 1:1 to n additional input neurons. The modular controllers F0-F6 incorporate zero, two, four, and six internal feedback connections.

3 First Results

In Fig. 3, the mean fitness of 3 populations with modular controllers is compared to the mean fitness of 10 populations with central controllers after 500,

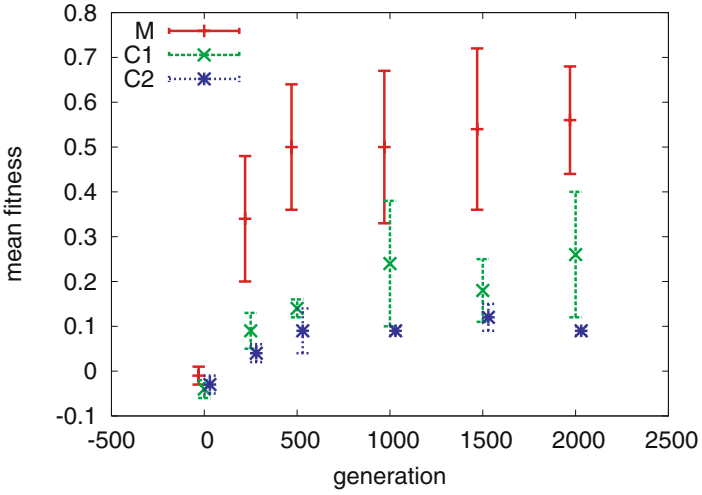


Fig. 3. The evolution of six-legged walker controlled by a modular controller (M) compared to that controlled by different variants of a central controller (C1, 3 feedback connections and C2, 18 feedback connections

1000, 1500 and 2000 generations. The modular controlled walkers reach a stable tripod gait from randomized starting positions. The fitness of the centrally controlled walkers increases much slower, and no tripod is established.

Figure 4 depicts the results of experiment 2: in 7 independent evolutionary runs, the optimal number of inter-leg connections is determined. From zero to two inter-leg connections, an increase in evolutionary progress and final fitness can be seen. For more than two inter-leg connections, no further advantage of additional connections is recognizable: Solutions take longer to evolve and do not reach a tripod gait in the observed 2000 generations.

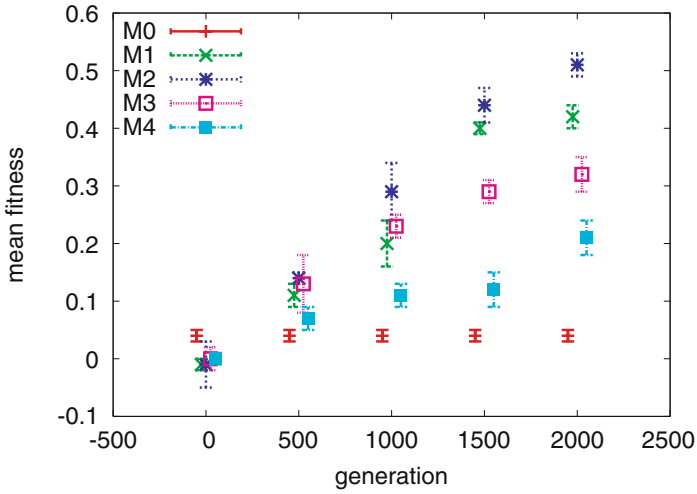


Fig. 4. The evolution of six-legged walkers controlled by different variants of a modular controller. M0-M4 feature 0, 1, 2, 3 and 4 inter-leg connections, respectively

For the dependency of fitness gain on the number of within-module feedback connections, no clear pattern is recognizable in 3 independent evolutionary runs (Fig. 5). The speed of evolution is highly variable between single runs, but finally all employed controller architectures reach a plateau with fitness values around 1.

4 Discussion

The two-dimensional approach proves to be a promising paradigme for the evolution of higher level behaviours in embodied agents. It can be used to rapidly evaluate the impact of morphological and neuroidal changes not on the level of dynamics (to accomplish that, see [2, 7, 8, 10]), but on the level of higher tasks relevant to moving artifacts (turning, obstacle avoidance, path planning). The results obtained so far suggest a clear advantage of modular

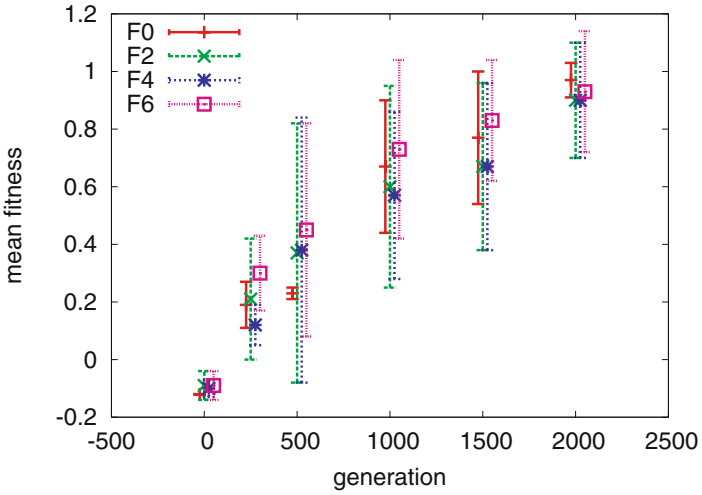


Fig. 5. Fitness gain in dependence on the number of feedback connections within each module in modular controllers. F0, F2, F4, and F6 incorporate 0, 2, 4, and 6 feedback connections in the controller of each leg, respectively

controllers as opposed to central controllers in terms of their evolability (Fig. 3). Given a modular controller, the question remains how much coupling is necessary between adjacent modules. The results in Fig. 4 propose 2 inter-leg connections to the anterior and posterior neighbour, respectively. It might however turn out, that for more compex tasks in more complex environments, a higher number of inter-leg connections is beneficial or even necessary. Interestingly, as can be seen in Fig. 5, the speed of evolution in 3 independent evolutionary runs shows no trend regarding the optimal number of feedback connections within each module. Purely reactive controllers, that do not have the possibility to evolve intrinsic pattern generators (neither within a module nor between modules), reach similar fitness values in similar timespans as modules endowed with internal feedback connections. If this finding can be confirmed for more complex tasks, it might allow judgements regarding the significance of pattern generators from an evolutionary point of view.

5 Outlook

As of submission date, it is investigated how the center of mass in robots influences optimal leg placement and if artificial six-legged walkers under special fitness restrictions evolve a positive feedback control comparable to the one found in the stick insect. In the future, different morphologies and neural architectures (multi layered perceptron, neural gas, ...) will be tested for their evolvability, hopefully in higher order tasks.

References

1. Bettina Blaesing and Holk Cruse. Stick insect locomotion in a complex environment: climbing over large gaps. *J Exp Biol,* 207(8):1273–1286, 2004.
2. Josh C. Bongard and Rolf Pfeifer. A method for isolating morphological effects on evolved behaviour. In Bridget Hallam, Jean-Arcady Meyer, Gillian Hayes, John Hallam, and Dario Floreano, editors, *From animals to animals 7,* pp. 305–311, Cambridge, Massachusetts, 2002. Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior, MIT Press.
3. H. Cruse. The control of the body position in the stick insect (carausius morosus), when walking over uneven surfaces. *Biol. Cybern.,* 24:25–33, 1976.
4. H. Cruse, T. Kindermann, M. Schumm, J. Dean, and J. Schmitz. Walknet – a biologically inspired network conctrol six-legged walking. *Neural Networks,* 11:1435–1447, 1998.
5. F. Delcomyn. The locomotion of the cockroach. *J. Exp. Biol.,* 54:443–452, 1971.
6. Volker Duerr and Tobias Authmann. Insect curve walking revisited: Transitions versus steady states. *Zoology 105 Supplement V (DZG 95.1),* 65, 2002.
7. M. Komosinski and Sz. Ulatowski. Framsticks: towards a simulation of a nature-like world, creatures and evolution. In *Proceedings of the 5th European Conference on Artifical Life,* Lausanne, Switzerland, 1999. Springer Verlag.
8. Hod Lipson and Jordan B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature,* 496:974–978, 2000.
9. J. M. Porta and E. Celaya. Efficient gait generation using reinforcement learning. In Karsten Berns and Ruediger Dillmann, editors, *Clawar 2001,* pp. 411–418, London, 2001. Proceedings of the Fourth International Conference on Climbing and Walking Robots, Professional Engineering Publishing Limited.
10. Karl Sims. Evolving virtual creatures. In *Computer Graphics (SIGGRAPH '94 Proceedings),* pp. 15–22. Annual Conference Series, 1994.

Legged Robot with Articulated Body in Locomotion Over Complex Terrain

Palis[1], Rusin[1], Schumucker[2], Schneider[2], and Zavgorodniy[1]

[1] Otto-von-Guericke University of Magdeburg, Germany
frank.palis@et.uni-magdeburg.de, rusin@iff.fhg.de, zavgo@web.de
[2] Fraunhofer Institute for Factory Operation and Automation, Germany
schmucker@iff.fhg.de,
schneider@iff.fhg.de

1 Introduction

Today, many questions about multilegged walking robots (the choice of construction, the design of control system, as well as locomotion organization) are known and well investigated. These investigations consider usually multilegged robot locomotion on smooth or easy rough terrain, overcoming simple obstacles, movement on soft ground, body maneuvering etc. From the algorithmically point of view this class of operations can be realized on robot kinematical level just in automatically mode by means of periodic gaits and contact information in discrete form (yes/no). From the mechatronical point of view this class of operations can realize robots with rigid bodies and simple contact sensors.

Recently investigations of improved robot tasks are concerned with he multilegged robot locomotion over an impassable road or a strongly complex terrain such as earthquake affected area, mountain regions, high ledges, ditches, trenches. The key performance for such tasks is the *additional body maneuvering*, the *measurement and control of support reactions* as well as the *control and forecasting of the robot motion stability* [1–3]. From the mechatronical point of view it is connected with enhancements in the robot construction and using the interaction force sensors. From the algorithmically point of view it is connected with the software development for the climbing tasks (with a possibility of using both legs and body for support during the separate stages of movement).

A number of the problems devoted to climbing tasks have been shown in publications [4, 5] considering robot movement inside or outside a pipeline. For this task the robot has to press itself against the inner or outer surface of the pipe and to move, so that support reactions were inside cones of friction. In [1]

are considered in details the problems of climbing up/down a large obstacle with the walking robot with rigid body. However, the functional capabilities of the walking robot on overcoming the obstacles can be essentially expanded by change of the body design. The problems of overcoming of obstacles equal to the geometrical size of the robot are not covered in the literature except the works [6, 7], which simulate the climbing of six-legged robot with an articulated body on a high ledge by means of a change in the body configuration during the separate phases of climbing. In [7] the design of such walking robot is considered and its mathematical modeling is presented. It is shown, that size of the obstacles, which the robot can overcome, increases in case of the body having controlled segments.

The technical purpose of this work is the development of a robot with additional controlled DOF in the body as well as with the possibility of the measurement and control of support reactions. The algorithmically purpose of this work is the development of control algorithms for overcoming the large obstacles.

2 Improvements in the Robot Construction

In accordance with requirements discussed above a multilegged robot with articulated body "SLAIR" (Fig. 1) has been developed at the Fraunhofer Institute for Factory Operation, Magdeburg, Germany and Otto-von-Guericke University of Magdeburg, Germany. The robot mechanics, sensor system and control system makes possible to maintain the additional flexibility in body, to measure and control the support reactions as well as to control and forecast the robot motion stability.

Fig. 1. Modular multilegged robot "SLAIR" with articulated body

2.1 Mechanics

The robot construction consists of $n = 3$ modular segments (shoulders) linked to each other through one DOF joints and 6 legs. Each shoulder includes one articulated body segment linked with two 3-DOF-insectomorphic legs. It is possible to extend the construction to the case of $n > 3$ shoulders. The main mechanical parameters are shown in the Table 1. The robot drives was realized as servomotors, which consist of DC motor with maximal power $P_{DC} = 4.5\,W$, potentiometer with angular range $\varphi_{POTI} = \pm 90^\circ$ and gear with ratio $i_{GEAR} = 251$ and efficiency $\eta_{GEAR} = 85\%$.

Table 1. The main mechanical parameters of the robot

Leg Lengths	Total	$l_{LEG} = 314$ mm
	Shoulder	$l_0 = 64$ mm
	Thigh	$l_1 = 100$ mm
	Shank	$l_2 = 150$ mm
body dimensions	body total length	$l_B = 760$ mm
	body segment length	$l_{BS} = 260$ mm
total robot mass		$m_{ROBOT} = 3.2$ kg
maximal speed		$v_{ROBOT_MAX} \approx 1$ km/h
effective load		$m_{LOAD_MAX} \approx 2$ kg
maximal power consumption		$P_{MAX} \approx 90$ W

2.2 Sensor System

The sensor system of the robot consists of components that are standard for mobile robot and that makes possible to achieve autonomous robot functions in an environment. It includes:

- 20 potentiometers and 20 current sensors (installed in each robot joint),
- three-component force sensors (mounted in each leg's shank),
- 3-axis gyroscopic sensor (located in body) and
- further it will be equipped with stereoscopic camera system.

In accordance with requirement on measurement and control of support reactions the developed force sensor (Fig. 2) consists of two parts of the core measuring lateral components of support reaction, and the elastic parallelogram module for measurement of the longitudinal component.

In the compact sensor unit the force-measuring elastic plates are located in plane parallel to the action of the longitudinal force. The module is mechanically selective to the action of twisting moment; the strains caused by lateral components of a supporting reaction in the plates have identical sings.

Fig. 2. Three-component force sensor

The sensing units are made of metal (duralumin), are simple to manufacture, and joined together with a threaded coupling. Three-component sensor with the amplifier is mounted in the shank of a leg, and the bottom end of the sensor is connected to the foot. The sensor is designed for loadings up to $F_{CONTACT} = 50\,\mathrm{N}$; interference between channels does not exceed 1%.

Additionally the foot design should provide good friction with a support along all directions of action of support reaction. Therefore for increase of friction it's expedient to make a foot from rubber. The spherical form of foot is more preferable at absence the gimbal-lock between a leg and foot.

2.3 Control System

The hierarchically organized, modular, fast, and flexible DSP control system (Fig. 3) is located on-board and controls and monitors all actuators and sensors of the robot. The control system can be also extended for the additional shoulders in exactly the same manner as the mechanical structure. The realized hardware abstraction layer (HAL) of the robot provides flexibility and simplifies the development of control algorithms. The connection with standard PC can be made via RS-232, USB, FireWire®, or via Wireless LAN, or Bluetooth®.

3 Extending Locomotion Tasks

All above-mentioned properties of mechanical structure, sensor system and control system make possible from mechatronical point of view the standard locomotion tasks of multilegged robots such as:

- locomotion over rough terrain,
- foot force distribution,
- estimation or identification of the ground mechanical properties,

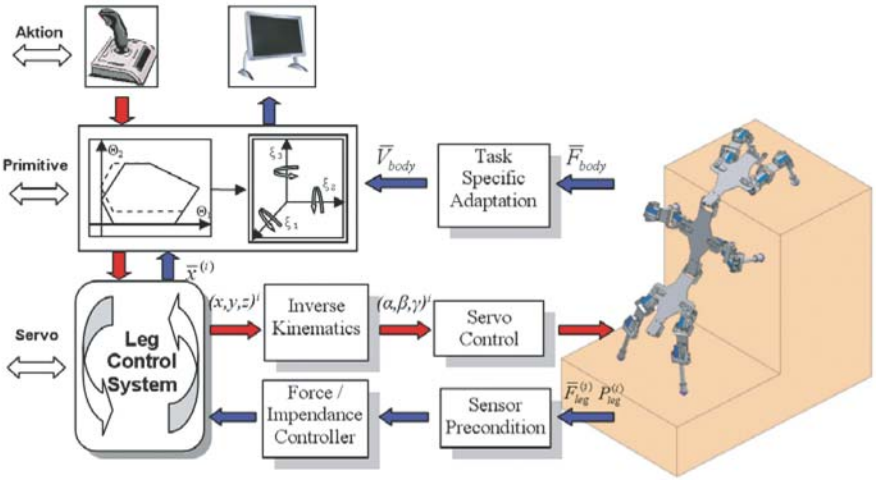


Fig. 3. The hierarchically organized robot control system

- maintenance of statically stability in a complex terrain,
- control of linear and angular movements of the body during operation,

as well as advanced locomotion tasks such as:

- overcoming obstacles equal in height to robots body, and
- motion over rather complex terrain.

In contract to standard locomotion tasks witch can be realized on "primitive level" advanced locomotion tasks are more complex and must be described on "action level". For example, the possibility of using the robot for the climbing on the obstacles equal to robots body (Fig. 4) will be examined. The key words for construction of the "climbing tasks" are: the "symmetrical gallop" gait mode, the definition of the foot support points on the "action level", the check and the prediction of motion stability and the movement in cones of friction.

The following algorithm is used for completing the climbing task in our simulation:

(1) the robot motion to the ledge as well as determination and processing of information about the obstacle dimensions (*as a result the calculation of the foot support points*);
(2) switch to "symmetrical gallop" gait mode (*always in following sequence "rear→ middle→front" shoulder*) and transfer the legs possibly near to the ledge ($d_{COM} \rightarrow 0$) concerning the stability margin (*Centre of Mass COM within support pattern*);
(3) lift up the front shoulder legs with the forward movement of the body at the same time and setting them on the horizontal obstacle part (*the first gallop wave after that is complete*);

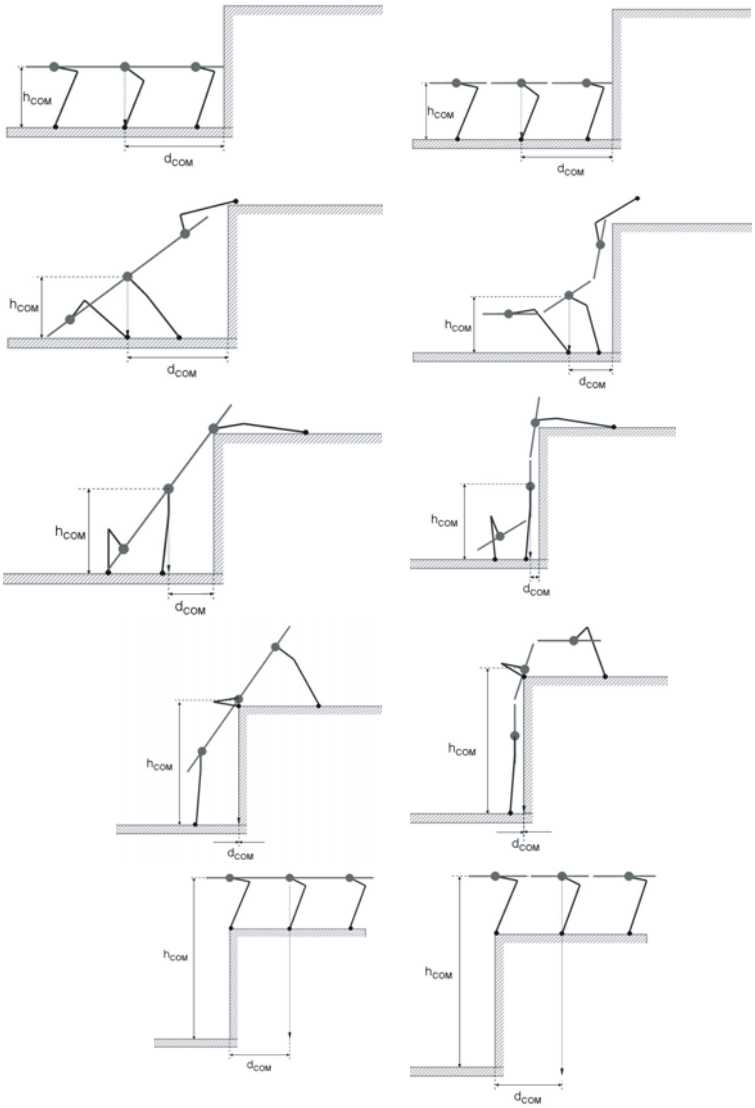


Fig. 4. Comparison of climbing between robots with rigid (*left*) and flexi (*right*) body

(4) transfer the body to the obstacle (*COM must be near as middle foots*);
(5) transfer the rear shoulder legs to the obstacle, than lift up the middle shoulder legs on the obstacle surface (*on the edge*), and than transfer forward the front shoulder legs as far as possible (*the second gallop wave after that is complete*);

(6) transfer the body after the obstacle (*COM between middle and front shoulder*);
(7) support on the middle and front shoulders as well as body and lift up the back shoulder legs, what is connected with the strict control of the support reaction and of the center of mass location inside the supporting area at each moment of time;
(8) transfer than middle and front shoulder legs (after that the third gallop wave and that climbing task is complete).

4 Discussion and Conclusions

As can be seen in algorithm considered above and in the pictures (Fig. 4) the usage of the articulated body allows closely approaching the COM to the ledge for the robot with flexible body rather than for robot with rigid body. Articulated body makes also possible to adjust steeper an angle α between the body segment and horizon and therefore to overcome bigger obstacles.

The maximal overcoming obstacle high can be estimated as following $h_{MAX} = l_{LEG} + l_{BS} \cdot \sin(\alpha)$. If the body segment length is smaller than the leg length, the usage of articulated body don't yields significant advantages concerning to obstacle height. In our case

$$h_{MAX_RIGID} \approx 314 + 260 \cdot \sin(50^\circ) = 497\,\text{mm}\ .$$
$$h_{MAX_FLEXY} \approx 314 + 260 \cdot \sin(80^\circ) = 560\,\text{mm}\ .$$

The usage of articulated body brings significant advantages concerning to calculation of the required maximal motor torque. Because in the most critical for a drive case (step 7 in the algorithm, Fig. 4 picture 4) the maximal motor torque $\tau_{MAX} \sim l_{BS} \cdot \cos(\alpha)$ is inversely proportional to cosine of angle α.

Therefore we can to conclude the following:

- The multilegged robot with articulated body has been developed. The modular design of the robot and of the control system allows easily to extend and to upgrade the robot with additional capabilities.
- The sensor system makes possible the completion of the standard and advanced autonomous tasks in complex environment.
- The robot is specially intended for development of control algorithms, for the investigation of the robot motion over extremely complex terrain, as well as for climbing obstacles that are equal to robot's body dimensions.

References

1. Song S.M., Waldron K.J. (1989) Machines that walk. MIT Press Cambridge, Massachusetts, 315

2. Okhotsimsky D.E., Golubev Yu F. (1984) Mechanics and motion control of an automated walking vehicle. Published House "Nauka", Moscow, 1984, 310 p. (in Russian)
3. Gorinevsky D.M., Schneider A. Yu (1990) Force control in locomotion of legged vehicle over rigid and soft surfaces. Int. J. Robot. Res., 9(2), 1990. 4–23
4. Pfeiffer F. (2000) Refined control for the tube crawling robot-Moritz. CLAWAR'02, Madrid, Spain, 2000, 339–346
5. Golubev Y.F., Korianov V.V. (2003) Motion design for six-legged robot overcoming the vertical column by means of friction forces. CLAWAR'03, Catania, Italy, 2003, 609–616
6. Alexandre P., Doroftei I., Preumont A. (1998) An autonomous micro walking machine with articulated body. CLAWAR'98, Brussels, 26–28 November, 1998, 61–66
7. Brazevic P., Iles A., Okhotsimsk D., Platonov A.K., Pavlovsky V.E., Lensky V. (1999) Development of multi-legged walking robot with articulated body. CLAWAR'99, Portsmouth, UK, 1999, 205–212

WALKIE 6.4: A New Improved Version of a Rigid Frames Hexapod Rover

N. Amati[1], B. Bona[2], S. Carabelli[2], M. Chiaberge[3], G. Genta[1], M. Padovani[4], and R. Volpe[4]

[1] Dipartimento di Meccanica and Mechatronics Laboratory,
[2] Dipartimento di Automatica e Informatica and Mechatronics Laboratory,
[3] Dipartimento di Elettronica and Mechatronics Laboratory,
[4] Mechatronics Laboratory, Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italy.
nicola.amati@polito.it

Abstract. Rigid frames walking rovers have proved to be effective for motion on rough terrain if low power consumption, simplicity of the mechanical structure and control system are basic specifications and low motion speed is not a limiting factor. A research line on rigid frames walkers is active at the Mechatronics Laboratory of the Politecnico di Torino since 1997. Several models of the prototype named Walkie 6 have been studied realized and tested. The present paper describes the improvements of the electromechanical subsystem and the characteristics the new control system of version 6.4. The mechanical system is realized without any modification of the architecture while a new control unit based on DSP and re-configurable FPGA device has been designed.

1 Introduction

Rigid frames walking machines were proposed in the past for planetary exploration (Beam Walker [5], Walkie 6 [0]) and for underwater civil works (ReCus [0]). They proved to be effective for carrying heavy payloads and for operation on extremely rough terrain [3–7, 9]. Rigid frames walkers are characterized by low power consumption and simplicity of the mechanics and of the control system. The acceleration and deceleration of the frames at each step limits the maximum speed and represent the main drawback of this type of machines.

The research work in the field of biologically inspired walking robots has not produced the expected results. The complexity of the control system and the lack of materials able to withstand high fatigue stresses in the joints are the main limiting factors.

The scenario previously described and the margins of improvements of non-zoomorphic machines suggested the Authors to persist in the research line of rigid frames walking robots. Three different prototypes of Walkie 6

were developed at the Mechatronics Laboratory of the Politecnico di Torino since 1997. Many efforts were devoted to the development of mathematical models [2, 8] with higher degree of complexity. The aim is to have a design tool for rigid frames machines and a validated virtual prototype to implement and test motion strategies.

The present paper is devoted to the description of the last prototype of Walkie 6, initially named 6.3 and then 6.4. The initial control system based on 6051 microprocessor was substituted by a new control architecture based on the so called AKU (Actuator Kontrol Unit) electronic board [10]. The improved performances are analyzed in the first part of the paper while the second part is mainly devoted to the description of the control architecture and of the electronic board.

2 Mechanical System

The design of the electromechanical system of the machine has been supported by the mathematical simulator [0, 0] validated experimentally using the previous version of Walkie 6.

The mechanics of the new prototype has been designed following three main guidelines.

1. Improvement of the performances of the rover on level and rough terrain without modifying the basic architecture of the previous version (Walkie 6.2). The main modifications are focused on the improvements of the actuators and of the general layout of the machine.
2. Increasing of the redundancy and sensitivity of the touch sensors with the objective of making the rover more reliable in autonomous operating conditions.
3. Optimization of the manufacturing process by simplifying the mechanical parts for assembly and reliability reasons.

The above mentioned considerations are here below described in detail.

Each actuator is powered by a brushless electric motor (Faulhaber/Minimotor 1525) with the driver and the position sensor embedded in the motor case. Information about the position is therefore available without any additional optical sensor and the interface with the control system is that of a typical DC motor.

The legs are telescopic as in the previous version [0] but the maximum stroke has been increased from 160 mm to 250 mm to improve the obstacle overcoming. The motion of the foot is realized by a lead screw (KERK, LPX 6-M15 pitch=1.5 mm) powered by a geared electric motor as in the previous version. The global reduction ratio is of 0.09375 mm/rev instead of 0.0789 mm/rev. The screw has a teflon coating and while the nut is in polyacetal with lubricating additive to reach an efficiency slightly lower than 0.5 ($\eta = 0.46$). The aim is to maximize the efficiency using no reversible system

as irreversible leg actuator allows no power consumption during the stance phase without any braking system. The main modification of the leg actuator regards the foot, which has been sensorized using four touch sensors placed in such a way to detect the ground reacting force also with a not centered resulting force. The sensitivity of the sensors has been chosen to ensure the switching in each load condition without having false signals. The aim is to increase the sensitivity and the reliability of the touch sensor system.

The longitudinal motion of the frames has been obtained by using two sliding bushings (STAR M 8) and a ball screw (SKF SH 6X2, pitch = 2 mm) powered by a geared electric motor. A reduction ratio of 0.5 mm/rev instead of 0.1579 rev/mm has been used to optimize the energy consumption and increase the speed. An additional increasing of the speed has been obtained modifying the longitudinal stroke of the frames from 180 mm to 280 mm. Data reported in Table 2 show that an increasing of the longitudinal stroke of 55% has been obtained by increasing the longitudinal dimension of the 20%. This is a result of the architecture optimization.

The machine layout has been resized to increase the steering angle from 24 to 40 degrees per step which means that 2.25 steps are needed to rotate of 90 degrees. The steering system is realized by a vertical shaft connecting the two frames. The rotation of the shaft is realized by worm gear connected to the geared electric motor (reduction ratio = 1080). Such a configuration has shown to be effective and reliable. Figure 1 shows the 3D view of the machine realized by SolidWorks® CAD and a picture of the mechanical subsystem of the robot while in Table 1 are reported the main characteristics of the rover.

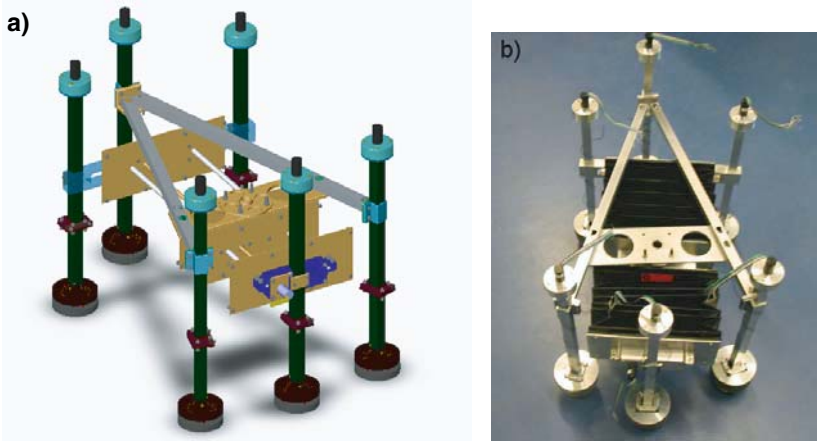


Fig. 1. Walkie 6.4: **(a)** SolidWorks® CAD model, **(b)** picture of the mechanical assembly

Table 1. Main characteristics of Walkie 6.4 compared with the previous version. The mass of the payload contains the power supply system (batteries)

Description	Units	Walkie 6.2	Walkie 6.4
Size (L × W × H)	[mm]	430 × 300 × 260	520 × 400 × 400
Maximum payload	[kg]	4	13
Total mass of the rover	[kg]	3.6	5.7
Mass of the upper frame	[kg]	1.4	1.8
Mass of the lower frame	[kg]	2.2	3.9
Mass of the leg moving part	[kg]	0.035	0.45
Motor type	–	Standard DC	Brushless
Legs stroke	[mm]	160	250
Longitudinal stroke	[mm]	180	280
Maximum steering angle	[deg]	24	40

3 Experimental Analysis

The aim of the present analysis is to validate experimentally the expected improvements predicted at the design stage by using the mathematical simulator. The analysis is mainly addressed to test the vertical and longitudinal motion.

The time history of the total current absorbed by the electric motor during each actuation (coils + internal driver) has been obtained by measuring the drop voltage over a sense resistor. A Tektronix TDS 3012 oscilloscope has been used for the signal acquisition.

The sense resistor has been placed between the ground reference of the generator and the ground reference of the motor, common to the coil circuit and to the driver, in such a way that the total current flowing to the motor runs through it. The value of the ohmic resistance (checked by a multimeter) is of 1.15 Ω. The recorded drop voltage signal has been exported for each test and filtered using a numerical algorithm. The voltage/current conversion is obtained considering the ratio between the voltage drop and the resistance of the sense resistor.

The vertical motion has been performed rising and lowering the rover of 100 mm with a payload of 13 kg, which means that considering the mass of the upper frame the axial load acting on each leg is of 50 N.

Results reported in Table 2 show that the average rising and the lowering maximum vertical speed increased of about 50%. This is the increasing of the leg elongation and of the total height of the rover. Otherwise the increasing of the power is of 10% while the average reduction of the energy is of 25%. This is mainly due to higher efficiency of the leg screw actuator, which has effects also on the maximum payload that the rover can carry (from 4 to 13 kg).

Table 2. Performance of the vehicle during vertical motion (max payload of version 6.2 = 4 kg, max payload of version 6.4 = 13 kg, max elongation = 100 mm)

Description	Unit	Walkie 6.2	Walkie 6.4	Perc %
Max. vertical speed (rising of 100 mm)	[m/h]	24.6	34.2	+39
Max. vertical speed (lowering of 100 mm)	[m/h]	40	65.5	+63.7
Energy consumption during rising	[J/m]	470	482	+2.5
Energy consumption during lowering	[J/m]	202	97	−52
Total power consumption during rising	[W]	9.65	13.8	+43
Total power consumption during lowering	[W]	6.71	5.32	−20.7

The longitudinal motion has been performed implementing the tripod gate strategy by a virtual panel able to coordinate the required motion strategy. The tripod gate provides a cycling step divided into six phases (1: rising of the lower frame legs, 2: forward motion of the lower frame, 3: lowering of the lower frame legs, 4: rising of the upper frame legs, 5: forward motion of the upper frame, 6: lowering of the upper frame legs). Two limiting working conditions have been considered:

- forward motion on level ground (leg elongation of 10 mm),
- forward motion on a uneven ground (maximum leg elongation of 250 mm).

A feet elongation theoretically tending to zero is required for motion on flat surfaces. However, the leg actuators have been driven to span a stroke of 10 mm, which is a reasonable value without running into practical problems. Such a condition corresponds to the maximum horizontal speed of the rover. Figure 2 a shows the time history of the total current flowing in the electric motors during a single forward step. A simultaneous actuation of three legs occurs during phase 1, 3, 4 and 6. The corresponding current time histories are the sum of the current adsorption of each actuator.

The worst condition in terms of average speed and energy consumption occurs when each leg actuator needs to cover the maximum elongation at each step to avoid obstacles. Such a condition occurs if a certain number of obstacles having a height equal to the maximum vertical clearance of the rover, are disseminated on a level ground. The maximum elongation of the legs is the most powerful strategy to avoid the impact with obstacles in this case. The speed characterizing such a motion may be considered the lowest for the rover performing a straight motion without vertical rising of the payload or steering operations due to contacts with the obstacles. The total current

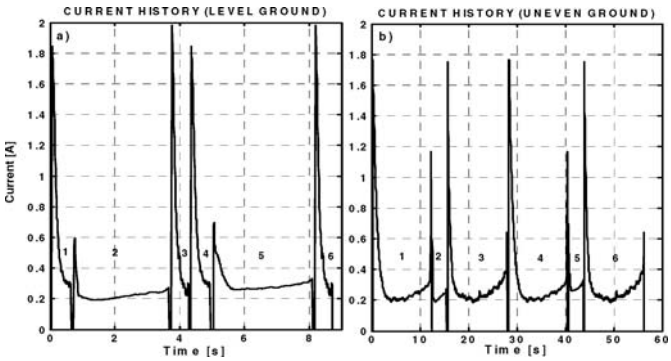


Fig. 2. Total current time history of the forward motion step on a level (**a**) and rough (**b**) surface. Each phase is numbered, the motions of the longitudinal stroke is of 280 mm while the stroke of the legs is of 10 mm in case (a) and of 250 mm in case (b)

Table 3. Longitudinal horizontal motion. Performances of Walkie 6.4 ($h_{\max}$ = 250 mm, frame stroke = 280 mm) compared with version 6.2 ($h_{\max}$ = 160 mm, frame stroke = 180 mm)

Description	Unit	Walkie 6.2	Walkie 6.4	Perc.%
Max. average speed ($h_{\max}$ = 10 mm)	[m/h]	53	115	+117
Mean power ($h_{\max}$ = 10 mm)	[W]	1.3	4.2	+223
Total energy consump tion ($h_{\max}$ = 10 mm)	[J/m]	86.5	130	+50
Energy consumption of legs ($h_{\max}$ = 10 mm)	[J/m]	48.1	61.7	+48
Min. average speed (max leg stroke)	[m/h]	14	32	+128
Mean power (max leg stroke)	[W]	1.36	3.7	+172
Total energy consumption (max leg stroke)	[J/m]	337	743	+120
Leg energy consumption (max leg stroke)	[J/m]	299	674	+125

time history, related to this operating condition (Fig. 1b) shows that the step time and the energy consumption due to leg actuation is a relevant fraction of the total time and energy consumption.

Results reported in Table 3 summarize the main performances of the vehicle moving in longitudinal direction. The relevant increase of the speed is paid by an increasing of the mean power. The higher payload has no effect on the energy consumption which is increased for the higher mass the feet (see leg

elongation of 10 mm) while the difference of the maximum leg elongation is the main reason of the different energy consumption during the walking with maximum leg elongation.

4 Control Unit

The control system of Walkie 6.4 is designed to control individual leg movements, coordination, gait, obstacle avoidance, and recovery from anomalies.

The functionalities of the rover can be divided into three levels:
Low:

- driving and position control of the actuators,
- ground sensorization (current monitoring and touch sensors),
- parameters storage during standard or accidental shut down.

Middle:

- local navigation (check the forward/backward step and the rotation of the rover)
- equilibrium control
- communication channel (radio) management

High:

- global navigation
- payloads management

The control of each actuator is base on state automata. Each operation of the low functional level (leg positioning, body motion) is implemented as a library function on the FPGA. It is possible to use them at upper levels. Moreover, the use of programmable logic devices allows low power consumption, time response in the order of microseconds (very important during real-time management of alarms), effective reduction of electronic components (optimization of space and mass). The state automata implemented on the FPGA use the previous described library functions to do elementary operations. The remote user can change some parameters through the radio link (for example the current threshold for the touch sensors, the maximum leg extension, etc. . .).

The control strategy can be modified during the motion of the vehicle on the base of accelerometers signals, commands and data coming from all the other navigation sub-systems (global navigation, local navigation, user commands).

The rover is provided with two radio links and a black and white micro camera. The first unidirectional, high bandwidth radio link is used to transfer video images at full rate from the rover to the control station; the second one, bidirectional, with lower bandwidth, to exchange telemetry and commands between the rover and the human operator. The human operator may use

the visual information from the camera to supply navigation and payload commands to the rover. The low level management of the radio system is performed by the FPGA.

The global navigation task is algorithmic. The controller can program the DSP to interact with the rover navigation system and send global obstacle avoidance commands as "go forward", "turn right", "turn left", etc. in order to plan the main direction of movement according to the rover status, terrain conditions and data coming from sensors and cameras. The "system supervisor" functions are therefore performed by the DSP.

The control platform, named AKU, supports dynamic re-configuration of the FPGA device: the basic idea is to create a flexible architecture based on DSP and FPGA. This allows to "virtualize" both HW and SW sources allowing the control algorithms to store different configurations on non-volatile memories and use them only when really needed by the system.

The control strategy is implemented on the AKU following the flow chart reported in Fig. 3.

Another result of the proposed architecture is an increased fault tolerance of the entire control system: in fact it is quite easy to repair a data error (caused for example by radiation or heavy ions) with the re-configuration of the "damaged" component.

Low-power strategies and components are also considered. All the system is designed to optimize power consumption using different solutions (clock optimization, stand-by of unused components, stand-by of the whole system under critical conditions, etc.).

5 Electronic Platform

AKU (Actuator Kontrol Unit) platform is made up of HOST and TARGET architecture. AKU HOST framework consists of a SERVER and CLIENTS. The SERVER concentrates the CLIENT applications requests and forwards them to the TARGET.

AKU TARGET framework is composed of 3 elements: software (real-time software and user tasks on DSP), firmware (FPGA) and hardware (FPAA and Field Modules). Figure 4 shows the block diagram of the AKU Platform.

Fig. 3. Control Strategy Implementation

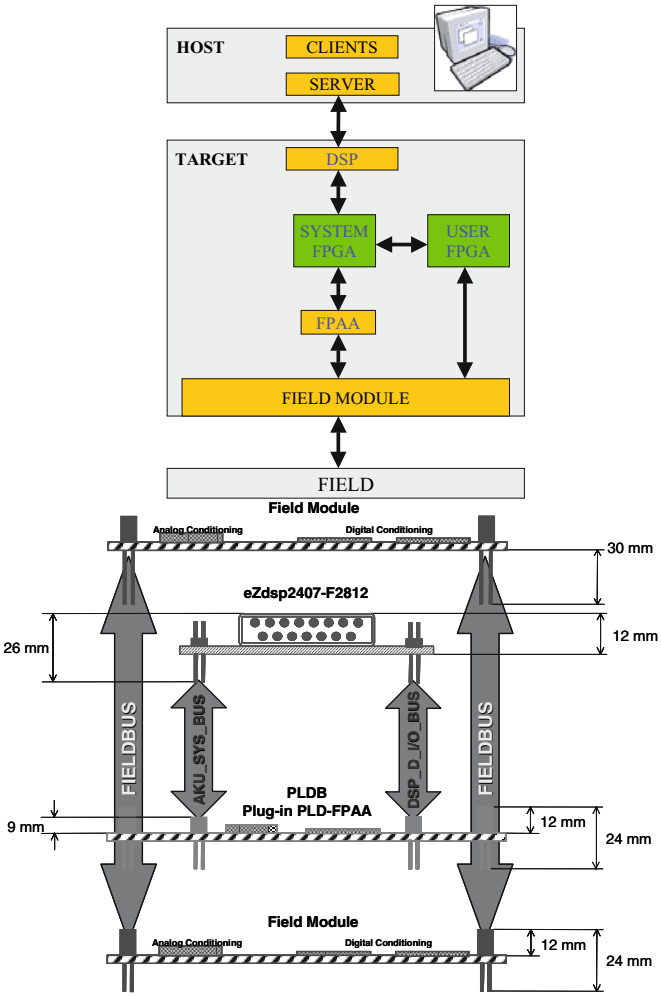


Fig. 4. AKU Platform diagram

6 AKU Platform

The AKU consists of two electronic boards: DSP board and FPGA/FPAA board.

The DSP board is the Spectrum Digital EzDSP2407 or EzDSPF28 (respectively based on TI DSP TMS320LF2407 or TMS320LF2812). The FPGA/FPAA plug-in is based on Altera FPGA and Anadigm FPAA and is produced by the CSPP-LIM– Politecnico di Torino.

The DSP is the master of the entire system; A RTOS manages the communication with the host and configure the platform. The users can implement in the users part the real-time application that can: monitor signals from

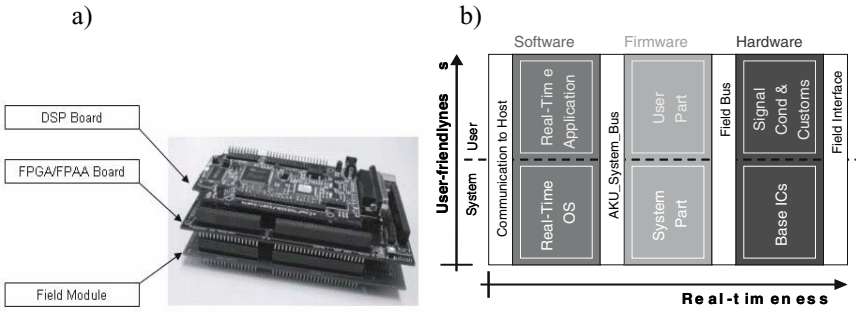


Fig. 5. (**a**): AKU Control Platform, (**b**) AKU System/User architecture

field, generate commands properly and implement the control algorithm. The FPGA operates as peripheral I/O extension of DSP functionalities, as well as coprocessor.

The user establishes its behavior in the User part Intellectual Properties (IPs), which interacts with the field (addressing digital I/Os) and/or improves the DSP computing capability. The System part (peculiar locked section of the firmware architecture) manages the communication between DSP and IPs in memory-mapped mode while the FPAA satisfies the request of configurable analog components for real time programmable analog conditioning using switched capacitance technology. The well-defined distinction between the System part and the User part is the key to obtain the openness of the system: the user can do everything with the security that the system will continue to operate (Figure 5b).

7 Field Module (FM)

The Field Module is a special purpose interface board; it serves as hardware interface between the AKU and the field, performing the required actions and conditioning signals to send back to the AKU. The features of a Field Module can be optical isolation, uncoupling functions, signal conditioning, voltage and current control. The AKU communicates with the Field Modules through digital and analog signals, grouped in the AKU Field Bus, which can be customized by the user through a simple and flexible assignment. The open structure and the available I/Os of AKU Field Bus allow plugging on the different Field Modules so that it is satisfied the demand of quite different applications. Changing the application field, the user can modify the DSP software application, the user firmware and the Field Module custom hardware, whereas HOST tools, RTOS and the AKU remains the same for all kind of applications. Figure 7 shows a Field Module device plugged into the AKU.

8 Conclusions

The preliminary experimental tests performed on Walkie 6.4 confirm that the improvements of the mechanical system expected at the design stage are effective. The vehicle architecture and the actuator efficiency optimization achieved the expected results in terms of

- longitudinal and vertical motion speed,
- energy consumption,
- reliability of the steering device and of the touch sensor system,
- increasing of the steering angle.

The performances and the improvements (compared with the previous version) of Walkie 6.4 underline that the field of rigid frames walking rovers is promising and need further investigation. This also because this typology of machines leads to be scaled up and downs maintaining the machine architecture and scaling the actuators and the structure.

The new control platform has been designed to achieve the requirements of autonomy and the flexibility of the rover. The control platform supports dynamic re-configuration of the FPGA device with the aim of realizing a flexible architecture based on DSP and FPGA. This allows to "virtualize" both HW and SW sources allowing the control algorithms to store different configurations on non-volatile memories and use them only when really needed by the system.

References

1. Genta G, Amati N, Chiaberge M, Miranda E, Reyneri LM (2000) WALKIE 6-A Walking Rover Demonstrator for Planetary Exploration. Space Forum, Vol. 5, No. 4, pp. 259–277
2. Genta G, Amati N (1998) Performance Evaluation of Twin Rigid-Frames Hexapod Planetary Rovers. In: Proceedings of the Fourth Int. Conference on Motion and Vibration Control (MoViC), Zurich, August 1998, Vol. 3, pp. 895–900.
3. Genta G, Amati N (2002) Non-Zoomorphic versus Zoomorphic Walking Machines and Robots: a Discussion. European Journ. Mech. & Env. Eng., Vol. 47, n. 4, pp. 223–237
4. Todd DJ (1985) Walking Machines: an Introduction to Legged Robots, Kogan Page Ltd., London
5. Roseheim ME (1994) Robot Evolution: the Development of Anthrorobotic, Wiley, New York
6. Song SM, Waldron KJ (1989) Machines that Walk: the Adaptive Suspension Vehicle, Cambridge-MIT
7. Peabody J, Gurocak HB (1998) Design of a Robot that Walks in Any Direction, Journal of Robotic System, pp. 75–83

8. Genta G, Amati N, Padovani M (2002) Performance of twin rigid frames walking rover on uneven ground-simulation and experimental tests", Proceedings of the 5th Int. Conference on Climbing and Walking Robots, Paris, September 2002, pp. 515–522
9. Genta G, Amati N (2003) Mobility on planetary surfaces: may waking machines be a viable alternative. Planetary and Space Science, Vol. 52, pp. 31–40
10. Carabelli S, Chiaberge M, Miranda E, Argondizza A, Del Mastro A (2000) Rapid prototyping of control hardware and software for electromechanical systems. IFAC Mechatronics 2000

WallWalker: Proposal of Locomotion Mechanism Cleaning Even at the Corner

T. Miyake[1 2] and H. Ishihara[1]

[1] Kagawa Univ. Japan
[2] MIRAIKIKAI Inc., Japan

Abstract. The purpose of this research is to develop the window cleaning robot for cleaning a single large windowpane such as a show window. It requires the following demands to apply the window cleaning robot for the practical use:

1. Clean the corner of window because fouling is left there often.
2. Sweep the windowpane continuously to prevent making striped patterns on a windowpane.

The keys of mechanisms are the rotatability of the mobile part around the other parts and the continuous locomotion in order to achieve the above points. The former enables the robot to change the direction with keeping its position and attitude at the corner of window. The latter is necessary for preventing leaving the striped pattern on a windowpane. We designed the continuous motion using two-wheel locomotion with adhering on the windowpane using a suction cup.

The size of prototype is about 300 mm × 300 mm × 100 mm and its weight is about 2 kg without batteries. As the results of basic experiments of the prototype on a vertical smooth window glass, traveling velocity of going up direction was 0.08 m/s, one of going down direction was 0.14 m/s and horizontal direction was 0.11 m/s.

In this paper the 1st chapter mentions background and objectives of this research, and also introduces the concept of WallWalker. The 2nd chapter discusses the adhering and moving mechanism. The 3rd chapter illustrates its basic properties based on the experiments. Finally, problems and future works are discussed in the 4th chapter.

1 Introduction

Recently, we have had many requests for the automatic cleaning of outside surface of buildings. Some customized window cleaning machines have already been installed into the practical use in the field of building maintenance. However, almost of them are mounted on the building from the beginning and they needs very expensive costs. Therefore, requirements for small, lightweight

and portable window cleaning robot are also growing in the field of building maintenance.

As the results of surveying the requirements for the window cleaning robot by the field research with the cleaning companies, the following points are necessary for providing the window cleaning robot for practical use:

1. It should be small size and lightweight for carried by one person to everywhere.
2. Clean the corner of window because fouling is left there often.
3. Sweep the windowpane continuously to prevent making striped pattern on a windowpane.

The locomotion mechanism must be chosen to satisfy these demands, especially later two subjects. Here locomotion mechanism means the combination of adhering mechanism, traveling mechanism and a mechanism for changing a traveling direction.

Various researches of locomotion mechanisms on wall climbing robots have been reported [1–5]. However they do not adapt to above three points completely. For example, climbing robot by legged-wall walking can not realize the continuous movement, and also its turn-ability is low [6].

We focused on the application of the window cleaning robot on a single windowpane. It is apparently necessary to cross over the window frame or joint line to use it at any window, but the single windowpanes like as a show window also exist as an important application.

According to such considerations, we adopted the two-wheel locomotion mechanism with adhering by a suction cup. This paper mainly deals with this mechanism and functions specialized in cleaning the corner of window.

First requirement brought the following specifications for designing the window cleaning robot.

– Weight: 5 kg, including the weight of battery and washing water,
– Size: 300 mm × 300 mm × 100 mm.

These are also defined by the results of surveying the demands from the cleaning companies.

This paper proposes the small, light and portable window cleaning robot named WallWalker, which are designed to satisfy the market demands as mentioned above. Figure 1 is the rendering at a scene of practical use of WallWalker. The WallWalker is adhering on a windowpane and cleaning as moving on large windows.

This paper discusses the effectiveness of proposed locomotion mechanism. The 2nd chapter discusses the locomotion mechanisms and illustrates the prototype for testing the proposed locomotion mechanism. The 3rd chapter illustrates its basic properties based on the experiments.

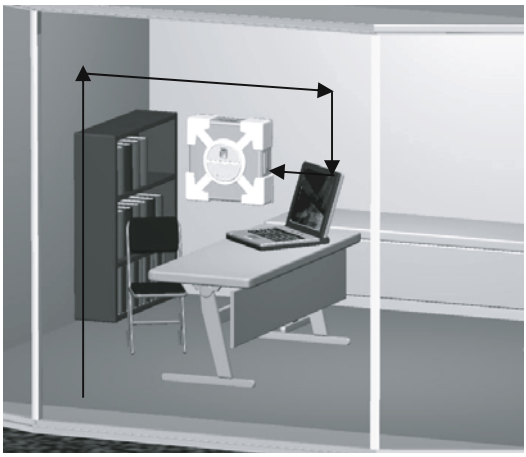


Fig. 1. Small-size window cleaning robot on a window

2 Locomotion Mechanism

Various researches of locomotion mechanisms on the window cleaning robots have been reported. However they do not meet our specifications defined based on the market demands above-mentioned. For example, climbing robot by legged-walk cannot realize the continuous movement, and also its turn-ability is low [6]. Climbing robot using crawler mechanism allows continuous movement, but the rotatability is as low as or lower than the legged-walk [7]. Window cleaning robot by crawler mechanism had been developed (Size: $440 \times 400 \times 180$ mm Weight: 6.5 kg maximum speed 2 cm/sec) by Shraft et al. [8]. It must bring its own crawler up from the adhering surface and rotate it in order to change its traveling direction. This mechanism needs strong adhering force to hold the whole system on the vertical plane with lifting the mobile mechanism, and also it takes a long time to finish the process of changing its front.

Both of Legged-Walk and Crawler mechanism need the complicated structures, and therefore it is difficult to lighten and downsize it.

According to such considerations, we adopted the two-wheel locomotion mechanism with adhering by suction cup. Figure 2 shows conceptual structure of WallWalker, which includes two driving wheels, a suction cup put in the center of robot, an air regulator, a small vacuum pump, some electronic circuits and some cleaning units. This chapter deals with the details of structures and the prototype designed for testing the proposed mechanism.

2.1 Traveling Mechanism

WallWalker moves on windowpane by two wheels with holing the body on the surface using a suction cup. The most important point in the mechanism is

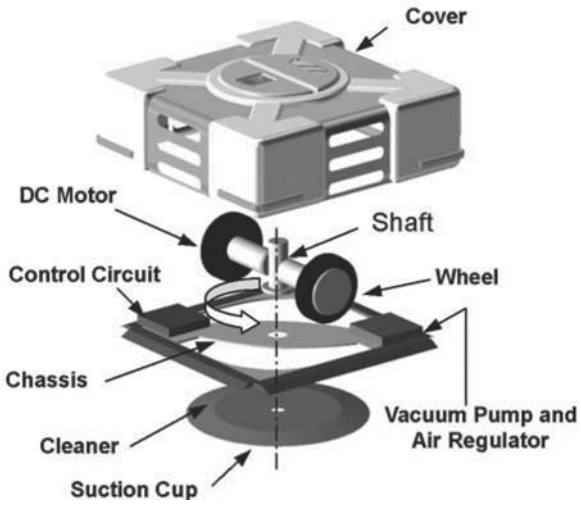


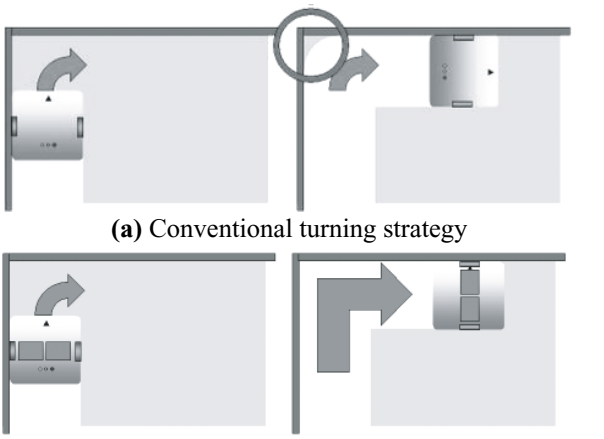
Fig. 2. Overview of small-size window cleaning robot

the friction coefficient of suction cup and tire against the adhering surface, e.g. high friction between the tire and the surface of window transmits the torque, and low friction between the suction cup and the surface of window. It achieves to move the robot with holding the body on the window. We selected PTFE (Polytetrafluoroethylene) for the materials of surface of a suction cup, and silicon rubber for the material of tires.

2.2 Turning Mechanism

Turning mechanism is a key to clean even at the corner of window. Figure 3 shows the scenes that the robot changes its traveling direction at the corner. Figure 3(a) shows a usual turning way like as turning of motorcars. In this case, since the robot changes a direction as tracing an arc, it can not reach the end of corner of window. It needs the complicated process as follows to clean the corner by such robot: first, the robot goes into a corner, next it moves back the distance to turn, then it changes its direction as tracing an arc. In case that the robot can change its direction at the end of corner as shown in Fig. 3(b), the robot can clean a corner easily and rapidly. Round-shape robot is easily able to turn at the corner, but it unable to reach the end of corner. On the other hand, a quadrangular robot can clean to the end of corner, but never turn itself there.

To get a function to change direction as shown in Fig. 3(b), we designed the mechanism that a mobile unit and a cleaning part are rotatably connected at the center shaft as shown in Fig. 2. Proposed mechanism consists of an adhering part, a cleaning part and a mobile part. The adhering part is constructed of a suction cup covered with PTFE and a vacuum pump. The



(a) Conventional turning strategy

(b) Novel turning strategy, which enables to clean a corner

Fig. 3. Turning mechanism at a window Corner

cleaning part is fixed to the adhering part. The mobile part uses two-wheel driving mechanism and is connected to the center shaft of the adhering part with suspension springs.

2.3 Suspension Mechanism

It is very important to press the tires on the adhering surface with the force enough to generate the friction to move itself. Because the suction cup deforms its own shape by the condition of vacuum such as a vacuum pressure, it is impossible to calculate the posture of robot against the adhering surface initially. That is, the force that the tire is pushed on the adhering plane must be adjustable to the adhering force.

WallWalker is introduced suspension springs into as an adjusting mechanism. They are placed between the mobile part and the adhering part, and enable to touch the tires on the adhering plane with a suitable force for the generating the friction.

2.4 Prototype of Locomotion Mechanism

Figure 4 shows the photograph of prototype developed to test the proposed turning mechanism. The size of prototype is about 300 mm × 300 mm × 100 mm and its weight is about 2 kg without batteries. The chassis that is made of aluminum alloy is formed square, and its inner area is hollowed to rotate mobile part at changing traveling direction. This contains two DC motors, suspension mechanism, a vacuum pump (−23 kPa) a suction cup which diameter is 150 mm, an air regulator and some electronic circuits. This robot is currently controlled from outside via cables and electric energy is also supplied by a power strip.

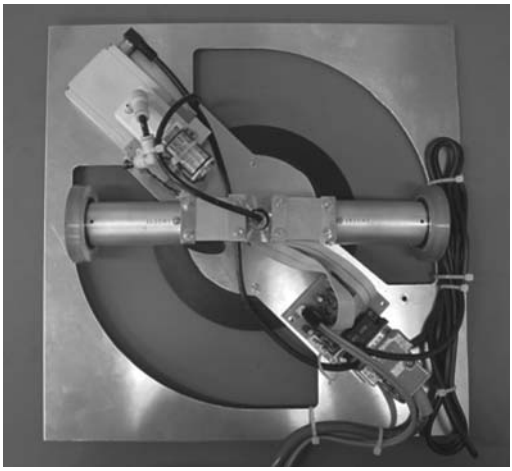
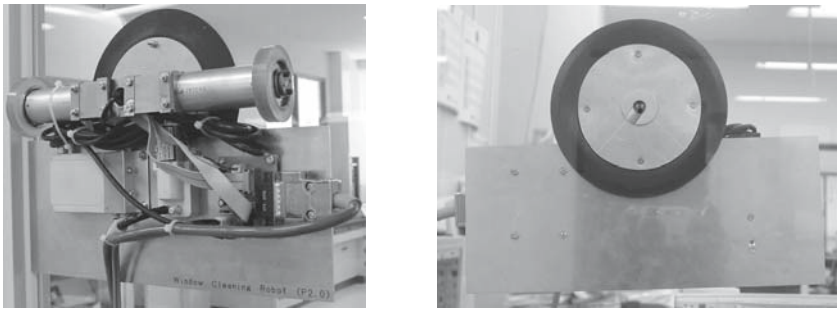


Fig. 4. Developed prototype

3 Experimental Results

At first the basic properties on a vertical smooth window glass have been tested. As the experimental results, traveling speed of going up direction was 0.08 m/s, one of going down direction was 0.14 m/s and horizontal direction was 0.11 m/s (Fig. 5). Also, the robot kept its body on the window stably and did not fall down during moving. These results proved its basic performance satisfies the specifications defined based on the field surveying.

Next, rotatability of prototype at the corner of window was confirmed by the experiment. Figure 6 shows sequential photographs when the prototype turns at the corner using turning mechanism proposed in this paper. As shown by these photographs, it was verified that the prototype can change its traveling direction at rights smoothly.



(a) Prototype is climbing up a window (b) Back side of prototype

Fig. 5. Mobility measuring of prototype

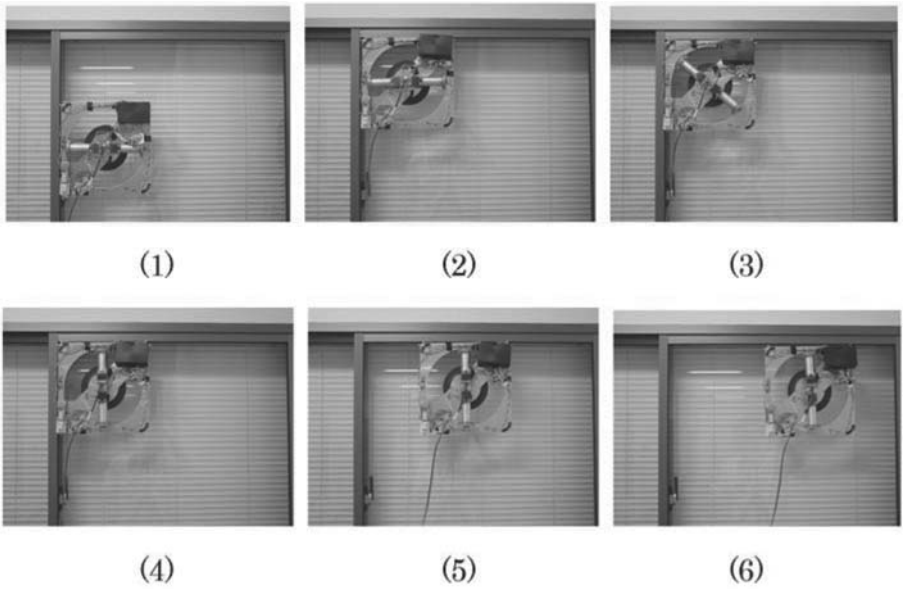


Fig. 6. Test of rotatability of prototype at the corner of window

4 Conclusion

Proposed WallWalker, which provides the continuous motion on the vertical windowpane and rotatability that the robot can change its traveling direction at the corner of window, was designed for cleaning the end of corner of window. In order to verify the basic properties about above abilities, the prototype was developed. Those results proved that the prototype fill the basic requirements mentioned in 1st chapter.

As the next development, the installations of control system and cleaning unit are planed. Sensors such as the posture sensor, e.g. gyro sensor, will be mounted and control scheme will be developed. Finally, the cleaning abilities will be tested with some cleaning units.

Acknowledgements

This research was supported by Foundation of Nankai-Ikueikai, Takamatsu, Japan. We greatly appreciate their support and encouragement.

References

1. Ryu SW, et al. (2001) Self-contained Wall-climbing Robot with Closed Link Mechanism. Proc. of the 2001 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, pp. 839–844

2. Fung HY, et al. (2000) Development of a Window-cleaning Robot, Workshop on Service Automation and Robotics, CIDAM2000, pp. 148–153
3. Longo D, Muscato G (2002) Design of a climbing robot for wall exploration – a neural network approach for pressure control onboard the Alicia II prototype. 5th Int'l conf. On Climbing and Walking Robots, pp. 1021–1026
4. Wang Y, et al. (2000) The study and application of wall-climbing robot for cleaning. Third Int'l conf. On Climbing and Walking Robots, pp. 789–794
5. Cepolina F, Michelini RC, et al. (2000) Gecko-Collie-homecleaning automation of floors, walls and cupboards. Third Int'l conf. On Climbing and Walking Robots, pp. 803–811
6. Yoneda K, et al. (2001) Development of a Light-Weight Wall Climbing Quadruped with Reduced Degrees of Freedom. Proc. of 4th Int'l Conf. on Climbing and Walking Robots (CLAWAR), pp. 907–912
7. Fukuda T, et al. (1992) Wall Surface Mobile Robot Having Multiple Suckers on Variable Structural Crawler. Proc. of Int'l Symp. on Theory of Machines and Mechanisms, pp. 707–712
8. Schraft RD, et al. (2002) Mechanical Design of an Autonomous, Lightweight Robot for window cleaning. Proc. of the 33rd Int'l Symp. on Robotics (ISR)

Behaviour Networks for Walking Machines – A Design Method

Jan Albiez[1] and Rudiger Dillmann[1]

Interactive Diagnosis- and Servicesystems, Forschungszentrum Informatik (FZI), Haid-und-Neu-Str. 10-14, D-76131 Karlsruhe, Germany
albiez@fzi.de

Summary. The high complexity of the mechanical system and the challenging task of walking itself makes the task of designing the control for legged robots a difficult one. Even if the implementation of parts of the desired functionality, like posture control or basic swing/stance movement, can be solved by the usage of classical engineering approaches, the control of the overall system tends to be very inflexible. A lot of different control approaches to solve this problems have been presented over the last years, but most of the time lacking the description of a design method for using them. This paper introduces a method for designing the control architecture for walking machines using behaviour networks.

1 Introduction

There have only been a few attempts to use behaviour based architectures on the lower levels of the control architecture for kinematically more complex robots like walking machines. The best known and most successful is the subsumption architecture [8, 10] used on several hexapods. A more biological inspired approach for a lobster-like robot is proposed in [6]. But there are several drawbacks to these architectures, among them a general tendency towards scalability problems, weaknesses when adding new behaviours or trying reusing existing ones and in most cases a highly problem specific approach (see [5]). Most of this architectures give an overview for the problem specific implementations they have been used on. Nevertheless all of this descriptions lack an overview of design guidelines or design patterns used to implement it. This makes it hard for other people to use the proposed approach for controlling their own robots.

Over the last years a new approach for a network-like behaviour based architecture has been developed at our Institute and successfully used to control a four-legged walking machine Bisam ([2] and [3] and Fig. 1). We started using this architecture on our other walking robots (Lauron III [11], Panter [4] and AirInsect [7]) as well. This reimplementations have been used

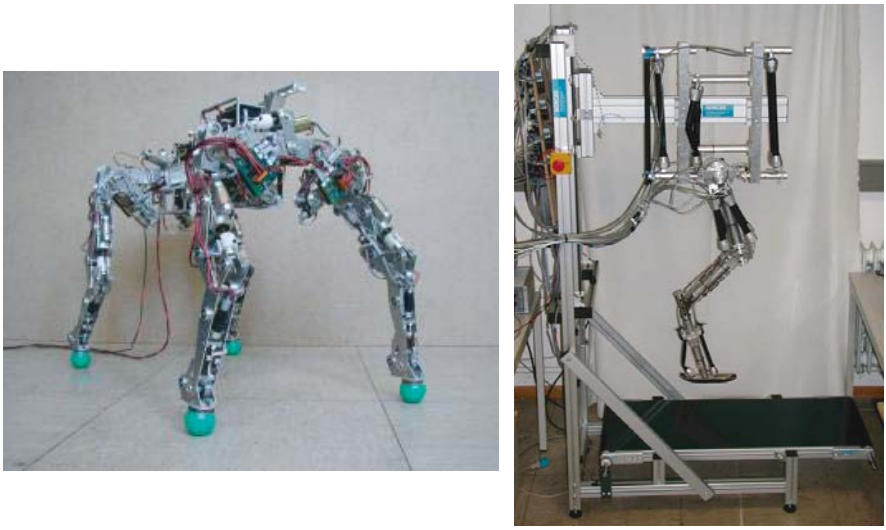


Fig. 1. (*Left*) the four legged walking machine BISAM, (*Right*) a leg of the walking machine PANTER in the testrig

to formulate several design rules and guidelines. In this paper we try to give an overview of this guidelines. First we give a short introduction to behaviour networks, after that follows the description of the design algorithm. The paper finishes with a short conclusion.

2 Behaviour Networks

Behaviour Networks have first been introduced in [1] and described in detail in [3]. This section gives only a short overview for a more detailed description refer to [3]. A behaviour or reflex B in the sense of this architecture is a functional unit which generates an output vector $\mathbf{u}$ using an input vector $\mathbf{e}$ and a motivation ι according to a transfer function $F(\mathbf{e})$. Additionally a target rating criterion (reflection) $r(\mathbf{e})$ and a behaviour activity a is calculated. Mathematically this can be combined to the 6-tuple

$$B = (\mathbf{e}, \mathbf{u}, \iota, F, r, a)$$

In short words the special I/O of a behaviour can be characterised as follows: ι is a fuzzy on-off switch for the behaviour, giving other behaviours the opportunity to control the behaviour and therefore its impact on the robot. r is a value between 0 and 1 representing an estimation of the actual robots state from the behaviours point of view, and a is an indicator of the actual work the behaviour is doing to change the robots state.

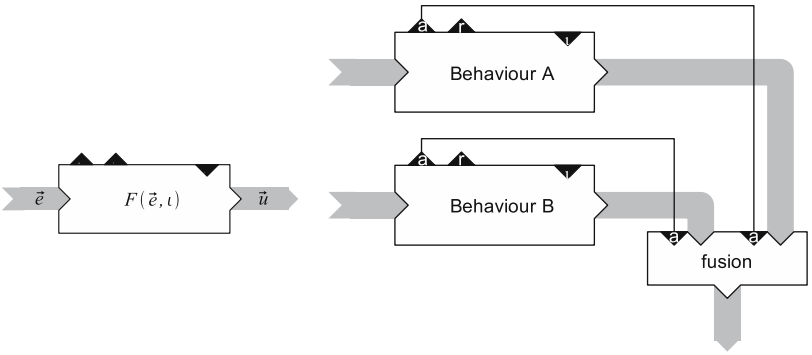


Fig. 2. (*Left*) a basic behaviour block, (*Right*) the concept of behaviour fusion via activities a

The activity a is also the key concept to solve the problem of behaviour coordination [13]. In the case that the output of several behaviours are connected to the same input of another behaviour or actuator, a is used to weight the output of the different behaviours. This scheme favours the output of behaviours having an unmet target (i.e. a high r) and a high activity a implying a ι greater than 0. Figure 2 shows the concept of the behaviour and the fusion.

The behaviours are placed in a network following the design rules introduced in the next chapter. Each behaviour in this network forms a region R. A region R is recursively defined as in (1), where $Act(B)$ is the set of behaviours being influenced by B via ι. It can colloquial described as all the behaviours which are directly or indirectly influenced by a behaviour (see Fig. 3).

$$R(B) = \bigcup_{B_i \in Act(B)} \{B_i \cup R(B_i)\}\,,$$
$$R(B) = \emptyset, \qquad \text{if} \qquad Act(B) = \emptyset \tag{1}$$

The affiliation of a behaviour to a region is not exclusive. Regions become quite important during the implementation phase of a network.

The most important concept in a behaviour network is the usage of lower behaviours ι as virtual actors and their a and r as virtual sensors. It is most likely, that a behaviour placed high-up in the hierarchy, like a gait, only acts on other behaviours ι and only reads their r and a as sensor information. This concept leads to a uniform interpretation of the robots system state and a uniform reaction to the environment. Especially the r guarantee that information generated in the system is not lost and only become more abstract moving up higher in the hierarchy.

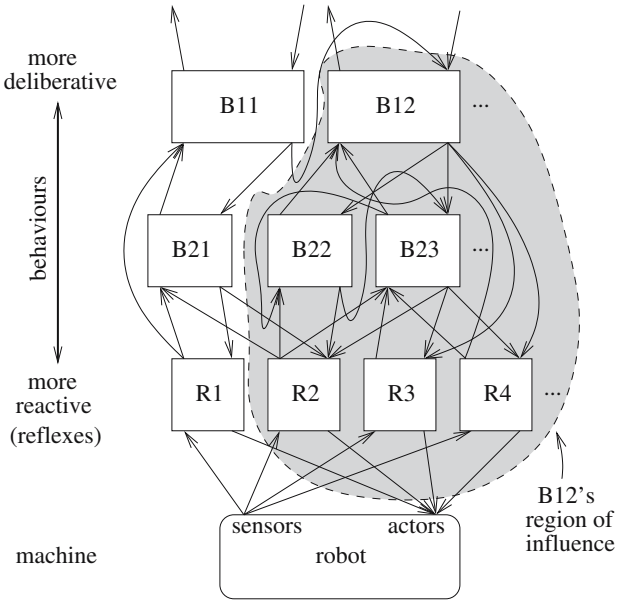


Fig. 3. (*Left*) an abstract layout of a behaviour network

3 Design Method

The proposed design method follows roughly a classical top-down specification and bottom-up implementation scheme. Since we are using a behaviour based approach the individual steps of the classical V-model can't be used due to the distributed approach of the behaviour networks. Also the design method has to take into account the special hierarchy system and the usage of ι, "a" and r as virtual actor and virtual sensors. In the following we first describe the specification and then the design phase step-by-step.

The specification phase is done, as mentioned, top-down in the following steps:

(1) **Identification**
Split the overall task you want to do in several behaviours and reflexes. This is a classical step when using behaviour-based robotics. When identifying the behaviour/reflexes use a top-down approach. For example the first behaviour for a walking robot will be the different gaits, then you need a swing and a trot behaviour. On most robots you are also forced to use reflexes for stabilisation and adaptation to the ground. Then you probably need reflexes for obstacle avoidance etc.

(2) **Analysis of kinematical structure**
The second step is the analysis of the kinematic structure. Find separate kinematic chains which can be controlled individually (e.g. legs).

These separate chains will form a tree structure with the machine local co-ordinate frame as root and the end effectors (most likely the feet) as the outermost leaves. The individual chains will further refer-ed to as *kinematic blobs*.

(3) **Distribution**
Distribute all the behaviours identified in step 3 to the kinematic blobs defined in step 3. A behaviour should reside in the outermost possible branch of the kinematic tree according to its function. This distribution is done from the actor point of view. For example resides a swing behaviour inside the leg blob since it only affects the actuators of one leg.

(4) **Sorting**
Hierarchical sort the behaviours in each kinematic blob by the level of planning they involve. The more reflex like, meaning less planning inside the behaviour is done, the lower it will reside in its kinematic blob. For each blob you will get a hierarchy from reactive to deliberative. This is the last step of the specification phase.

At this point the generic layout of the network should be clear. The connections between the behaviours will be created in the implementation phase. This phase is done bottom up, starting from the behaviours in the leaves of the kinematic tree and done for all the outermost kinematic blobs. Only if all the behaviours in a blob connected to an other blob higher up in the hierarchy are implemented and tested the implementation of the behaviours in this blob is started. Consider a kinematic blob from the outside as one "big" behaviour.

One of the key aspects of the behaviour-based methodology is embodiment [9]. In the direct consequence for the implementation of this architecture this means, that the following steps are done directly on or with the robot. The simulation of sensor systems or environment must only be done for testing purposes. If the robot can do it, let the robot do it.

For each behaviour the implementation is done in the following steps:

(1) **Specification**
Do a formal specification of the transfer function $F(\mathbf{e}, \iota)$ of the behaviour. To define F properly you need the answers to the following questions: Which sensor inputs e does it use, which output u does it generate. Is all the sensor information available as output from real sensors? Is the information extracted from the sensor data already needed and therefore by another behaviour? If this is the case gives this behaviours r and a sufficient information and could be used as virtual sensor? Pretty much the same applies for the actor output u: Can the ι of an underlying behaviour used as virtual actor instead of directly accessing the robots actors? The definition of the behaviour's r and a is done after the specification of F.

(2) **Identification of the region**
After the specification the region R of the behaviour has to be identified. Since the region is defined via the influence of one behaviour over others it only affects the actor (u). In the case that with the definition of the region a lower behaviour is part of the region of an other already implemented we have a conflict. This conflict is solved in (the last) step 3. For a smooth implementation process of the actual behaviour it is necessary to disable the other region until the last step.

(3) **Testing environment**
Before the actual implementation of the behaviour can be done a testing environment for this behaviour must be generated. This environment must be able to simulate all situations which a behaviour can encounter on the robot. The testing environment mainly consist of a software framework for embedding the behaviour in the network. Hardware test-platforms, equipment to fix the robot etc. should only be used if downright necessary. This could be the case in the early process (e.g. while implementing the blob for a leg).

(4) **Implementation and parametrisation**
Now it is time to implement the behaviour in software and do the necessary parametrisation by using the testing environment setup in step 3. The testing environment also gives the opportunity to test several implementation approaches for the behaviour (eg. PD-controller, fuzzy controller, neural network, finite state automata) and choose the most appropriate one.

(5) **Solving of conflicts**
As mentioned in step 3 the intersection of the regions of different behaviours is leading to a conflict. This conflict is solved by a fusion knot described in Sect. 2. Since this is the most critical point in the design of the network it has to be tested in detail. In some cases the definition of the activities from the behaviours in conflict have to be redefined. Is this the case each step of the implementation phase has to be done again for this behaviour.

The design process layout can and should be applied recursively if necessary. Therefore it is possible to expand the architecture with new functions whenever necessary without loosing the overview of the system.

A detailed description and evaluation of the implementation of this architecture can be found in [3] and [3]. Figure 4 shows the network controlling a leg of Bisam and Panter as example.

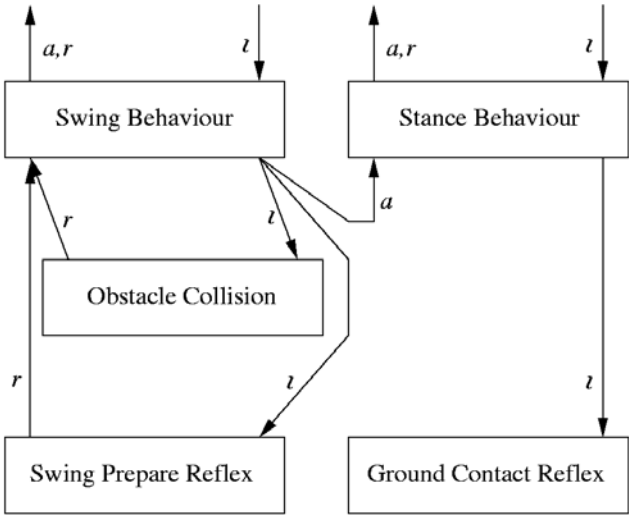


Fig. 4. The network controlling a leg of BISAM and PANTER [1, 2] with only the co-ordinating connections (ι, a, r) shown for clarity reasons

4 Conclusion and Outlook

In this paper we presented a method to design the behaviour networks used on the robots BISAM and PANTER. We introduced a method for identifying, placing and connecting behaviours. The critical points during the design process were stressed out and solutions to solve them introduced. Further research will a more complex implementation on the robot LAURON III allowing different gaits and a close interaction with a environment-model-database [12]. Another point of interested is to automate the development of a behaviour network on basis of the proposed design rules.

References

1. Albiez, J., Luksch, T., Ilg, W. and Berns, L. (2001) Reactive Reflex based Posture Control for a Four-Legged Walking Machine In *Proceedings of the 4th International Conference on Climbing and Walking Robots (CLAWAR)*, Karlsruhe, Germany.
2. Albiez, J., Luksch, T., Berns, K., and Dillmann, R. (2002). An activation based behaviour control architecture for walking machines. In *Proceedings of the 7th International Conference on Simulation of Adaptive Behaviour SAB*, Edingburgh, UK.
3. Albiez, J., Luksch, T., Berns, K., and Dillmann, R. (2003). Reactive reflex-based control for a four-legged walking machines. In *Roboticy and Autonomous Systems* 44 (2003), pp. 181–189.

4. Albiez, J., Kerscher, T., Grimminger, F., Hochholdinger, U., Dillmann, R. (2003). PANTER – prototype for a fast-running quadruped robot with pneumatic muscles In *Proceedings of the 6th International Conference on Climbing and Walking Robots (CLAWAR)*, Catania, Italy, pp. 617–625.
5. Arkin, R. (2000). *Behavior-Based Robotics*. MIT Press.
6. Ayers, J., Witting, J., Wilbur, C., Zavracky, P., McGruer, N., and Massa, D. (2000b). Biomimetic robots for shallow water mine countermeasures. In *Proc. of the Autonomous Vehicles in Mine Countermeasures Symposium*.
7. Berns, K., Albiez, J., Kepplin, V., and Hillenbrand, C. (2001). Control of a six-legged robot using fluidic muscle. In *International Conference on Advanced Robotics*, Budapest, Hungary.
8. Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23.
9. Brooks, R. (1991). New Approaches to Robotics *Science*, Vol. 253, September, pp. 1227–32
10. Ferrell, C. (1995). Global behavior via cooperative local control. Vol. 2, pp. 105–125.
11. B. Gassmann, K.U. Scholl, K. B. (2001). Locomotion of lauron iii in rough terrain. In *International Conference on Advanced Intelligent Mechatronics*, Como, Italy.
12. Gaßmann B., Frommberger L., Dillmann R., Berns K. Improving the walking behaviour of a legged robot using a three-dimensional environment model. In *Proceedings of the 6th International Conference on Climbing and Walking Robots (CLAWAR)*, 17–19 September 2003, Catania, Italy.
13. Pirajanian, P. (1999). Behaviour coordination mechanisms – state-of-the-art. Technical Report IRIS-99-375, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California.

Biological Inspired Walking – How Much Nature Do We Need?

Jan Albiez[1] and Karstan Berns[2]

[1] Forschungszentum Informatik an der Universität Karlsruhe (FZI), Interactive Diagnosis and Servicesystems, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany
albiez@fzi.de

[2] AG Robtersysteme, Fakultät für Informatik, Universität Kaiserslautern, D-67653 Kaiserslautern, Germany
berns@informatik.uni-kl.de

Summary. In the last decade a lot of research is done in the fascinating area of biological inspired walking machines. The idea of walking machines is based on the main method of land based propulsion found in nature: The legs. But do we need more inspiration from nature for building walking machines, or is using legs the only neccessary one and is it best to solve all further problems by classical engeneering means. This paper gives an overview of some principles transferred by researchers from biology to engeneering while building walking robots and tries to evaluate them. An overview of the research results will be given due to the subtask mechatronic system or morphology, control architecture and adaptive behaviour control. At the end of the paper several applications are listed for which biological inspired walking machines would be an optimal solution.

1 Introduction

The development of walking machines has been in the focus of world-wide robotic research since the early seventies of the last century. Since the idea of walking robots is to implement one of the best motion systems of nature for land based animals, it is obviuos that the research community has very often tried to mimic biology in more or less aspects. In this paper we would like to give an overview of different levels of biological inspiration in the design of walking machines of the last years. We try to look on the proposed solutions under the aspect of how much of the solution has been influenced by biology. With this knowledge we would like to give an estimation where it is reasonable to use the knowledge gained through biological research and where it is not.

In the following the term biological inspired walking machine, which belongs to the above described robots is introduced. This term can be defined as:

> Biological inspired walking machines contain all walking robots for which at least one of the system components like mechanical design, actuator system, control architecture or adaptive control is strongly influenced by design principles in nature. The development aims to reach similar locomotion capability as comparable animals under rough environmental conditions.

Examples of these kind of machines are an 8-legged ambulatory vehicle that is based on the lobster[a], the cockroach like machine Robot III Case Western University, Ohio USA [15], the humanoid robot Asimo[a], and the biped machine of the University of Munich, Germany [14]. Based on the research results of the neuroethology of the stick insect Carausius Morosus [8] in Germany several machines are developed like Lauron [10], Airbug, [4] both from FZI, Karlsruhe, Tarry from the University of Duisburg,[a] and Max of the University of Munich [18]. Examples for mammalian-like four-legged robots are Tekken I and Tekken II [13], BISAM [6] and PANTER [2].

2 System Requirements

Based on the above given definition of biological inspired walking machines the following system requirement must be fulfilled:

Autonomy The machine must be an autonomous mobile robot. Basic intelligence must exist because wireless communication from outside can not be guaranteed under rough environmental conditions.

Energy autarky The energy supply must be on-board.

Flexibility The number of active and passive degrees of freedom must be high so that the locomotion apparatus can adapt to ground. Bending forces or tenseness forces should be low to save energy and to allow smooth movements.

Reliability The robot must be reliable on all system levels starting from the computer architecture up to the behaviour control.

Adaptivity The control of locomotion must be implemented in a way that disturbances which arise e.g. from ground conditions can be managed.

For the realisation of biological inspired walking machines it is essential that knowledge from different research areas is used (see Fig. 1). Starting from research results of biologists concerning morphology or neuroethology of animals first an adequate mechatronic system must be designed and a sensor system which determines the inner state of the machine and detects the environment must be set up. Then an adequate control architecture and an adaptive behaviour control concept must be implemented.

[a]Northwestern University, MA, USA see http://www.dac.neu.edu/msc/burp.html

[a]Honda, Tokyo, Japan see http://ne.nikkeibp.co.jp/english/2000/11/1120asimo_d-ce.html

[a]see http://www.mechanik.uni-duisburg.de/publikationen/

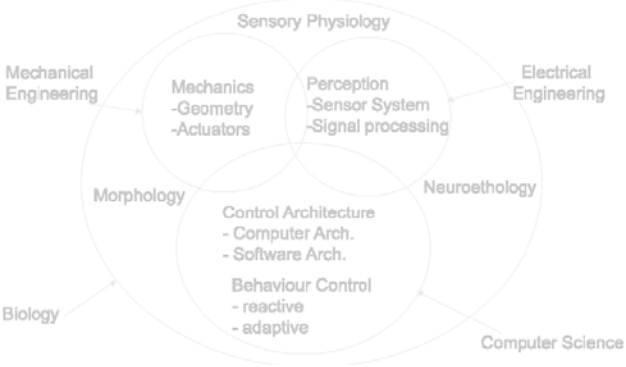


Fig. 1. Interdisciplinary solution for the development of biological inspired walking machines

In the following the state of the art of the different system components for biological inspired walking machines is discussed.

3 Advanced Mechatronic System

The physiology of the locomotion of creatures (drives and mechanics) are very different compared to walking machines constructed so far. It is very hard to transfer these concepts to walking machines. Mechanical design parameters from nature which are used for building biological inspired walking machines are the leg order, the proportion of the different leg segments (body parts), and the number of degrees of freedom. Because the machines are normally scaled up compared to their natural models the angle velocity is much smaller. In the case of the four legged machine BISAM [19] the biologically inspired aspects of the mechanics are the leg and the flexible shoulder and hip construction. Biologists had found out that most important for the locomotion of mammals is the hip, shoulder and spiral motion. Also a leg was constructed which is arranged in a rectangular zig-zag manner.

The actuators which are normally used to build up walking machines are not powerful enough to allow on one hand side fast movements and on the other side to generate necessary torques to move the machine with a payload. Even if control concepts with active compliance is implemented it is hard to cover impact problems. Therefore, actuators with passive compliance which behave similar than muscles in nature could be used for the development of walking machines. In literature there are several types of artificial muscles. Fluidic muscles with high forces, high contraction velocities and a good energy efficiency could be a solution for the actuator problem. E.g. in [9] a robot primate and in [17] a spider like machine is developed using different types of pneumatic muscles.

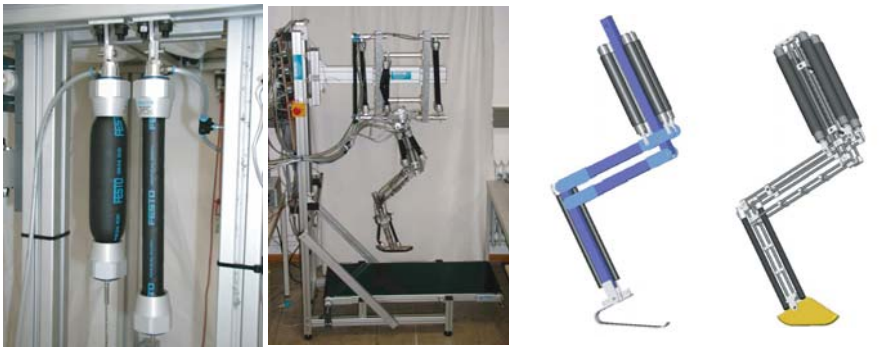


Fig. 2. (*Left*) the biological inspired pneumatic muscle actuator of the company FESTO. (*Right*) three different prototypes of the legs of the four-legged walking machine PANTER, driven by these muscles

4 Sensor System

Observations from nature have shown that the perception of creatures is based on a high amount of receptor cells (e.g. several thousands on the leg of the stick insect). A receptor cell delivers an analogue value or an on/of, value. In micro-mechanics research similar sensor components have been built. Up to now these sensors are only used in special robot systems. Walking machines should have at least the following internal and external sensors.

To the class of internal sensors belong shaft encoders (absolute or incremental), foot sensors (tactile or force sensors), inclinometers and gyroscopes to measure the state of the central body. Additionally, current and voltage in case of electrical motors and pressure in case of fluidic actuators must be determined. An indirect measurement of the torque and a protection against thermal overload can be performed with a known motor current.

For the control of the walking process in rough terrain it is essential to measure the distance to different objects and to determine soil properties which means to distinguish whether the ground is slippery, rather flat or strongly uneven and to find suitable areas for the foot points. Commercial sensor systems like laser scanner a normally to big and haevy to install them on small walking machines. Stereo camera systems combined with a laser which generates a structured light image seems to be a promising solution. The main advantage of such a sensor is that the calculation effort for the detection of an object is very low [12].

5 Control Architecture

Because biological inspired robots have huge number of actuators and sensors it is necessary for the reduction of the complexity problem to use a

Fig. 3. The six-legged walking machines Lauron III (*left*) and IV (*right*). The Lauron series is modelled after the stick insetc and uses electric motors as actuators

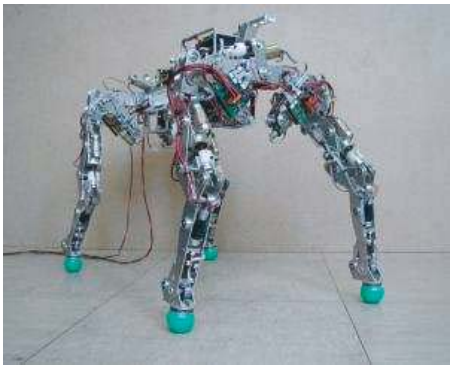
decentralised control architecture. The control architecture can be divided into computer architecture and software architecture.

In this example the computer architecture consists of three levels namely the micro-controller level, the PC level and the PC network level. The micro-controllers are directly coupled with a special power electronic card that allows the control of 4 motors. The micro-controllers are connected via Can-bus with an internal industrial PC. All sensors are directly coupled with the micro-controller. The sensors deliver both, analogue signals which are converted by the internal AD-converter of the C-167 and digital signals provided by joint encoders counted by the micro-controller. The second area provides the PC level with services including the management of the communication link to PC.

To handle the real-time control requirements a modular control architecture has been developed. Linux as well as Real Time Linux are used as operating systems. The selection has been performed because of the high number of available devices and the availability of source codes. The standard Linux kernel as a task of the RT-Linux kernel runs with a lower priority. For the efficient implementation of the different control levels the object oriented module concept MCA [16] has been implemented which enables a rapid development and the exchange of control algorithms on different control levels.

6 Adaptive Behaviour Control

Considering the features of neural networks like fault tolerance, the ability to process noisy sensor data and the possibility to learn from examples they seem to be an appropriate tool for implementing adaptive control mechanisms. In combination with Reinforcement Learning (RL), these approaches are very interesting because of their potential of online adaptability to changing environments. The main disadvantage is the huge amount of trials which are necessary to learn online a special control behaviour. Therefore, present re-



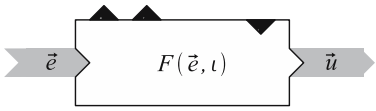


Fig. 4. (*Left*) the four-legged machine BISAM, (*right*) The basic behaviour building block for BISAMs adaptive control architecture

search deals with the integration of a priori knowledge into the learning concept. Similar as observed in nature neuro-oscillators are used to predefined rhythmic movements for the leg and gait control [11] and [13]. In combination with the neuro-oscillators the learning process can be used to learn and optimise the coupling weights and the influences of sensor information. For complex control behaviour it is still an open question how to determine the right weight.

Therefore, other approaches focused more on the realisation of adaptive control using reflexes which are coupled with traditional control methods. One of the most prominent approaches of this philosophy is the subsumption architecture [7] used on the six legged robots Attila nd Ghengis. In [1] a behaviour network architecture is introduced. In this approach all behaviours have a well defined interface and are placed in a network strucutre following the kinematic layout of the robot. Several experiments with this systems have been made on the robot BISAM.

7 Potential Applications and Evaluation

Based on the advantages of biological inspired walking machines applications can be found in all areas that require a locomotion in very rough terrain like:

- transportation of heavy loads along badly rugged country;
- emergency rescue works in the destruction zones with a possibility of utilisation of separate legs as manipulators;
- repair inspection works in rooms not suitable for mankind;
- moving on poor grounds, for example, moving in a forest, in the mountains;
- planatory exploration and submarine works;
- rehabilitation of people with restrictions of support movement apparatus.

However, up to now commercial and industrial use of walking is still in the status of developing prototypes. E.g. the Plustech company in Finland has developed a six-legged machine for forestry which was just sold a few times. The Odetics Inc. company in the US has built a small series six-legged machine for maintenance work in nuclear power stations. This leads to the question "what research problems have to be solved in future"? Main problems when building these machines arise in the case of mechanics from effective drive system, and the construction of very light but robust machines as well as a special foot construction, which absorbs the power stroke energy. The use of fluidic muscels offers a promising solution but also in this case a lot of improvements have to be done. Perception problems must be solved according to the interpretation of a huge amount of noisy and uncompleted sensor data and in the field of control from the fact that the locomotion behaviour can not completely determined a priori. In this area both new sensor systems as well as new methods for the interpretation of the measurements must be set up. In case of adaptive behaviour control several interesting approaches can be found in literature. These approaches are normally applied to simple control problems. It is still unclear how they can be used for complex control scenarios.Worldwide more and more projects have focused on the transfer of principles found in nature to develop powerful machines. The understanding of locomotion in nature will lead us to new concepts for walking machines, which will solve the above mention application problems.

References

1. Albiez J., Luksch T., Berns K. und Dillmann R. An Activation-Based Behaviour Control Architecture for Walking Machines. *The International Journal of Robotics Research*, Vol. 22, No. 3–4, March-April 2003, pp. 203–211.
2. Albiez J., Kerscher T., Grimminger F., Hochholdinger U., Dillmann R., Berns K. PANTER – prototype for a fast-running quadruped robot with pneumatic muscles In *6th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*, pp. 617–624, 17–19 September 2003, Catania, Italy.
3. Ayers, J., Witting, J., Wilbur, C., Zavracky, P., McGruer, N., and Massa, D. (2000b). Biomimetic robots for shallow water mine countermeasures. In *Proc. of the Autonomous Vehicles in Mine Countermeasures Symposium*.
4. K. Berns, J. Albiez, V. Kepplin, and Hillenbrand. Airbug–insect-like machine actuated by fluidic muscle. In *CLAWAR 2001–Climbing and Walking Robots and the Support Technologies for Mobile Machines*, september 24–26 2001.
5. K. Berns, K.-U. Scholl, W. Ilg, and R. Dillmann. Aspects of flexible control for mobile robots. In *ICRA 2000*, 2000.
6. Berns K., Ilg W., Deck M., Albiez J. und Dillmann R. Mechanical Construction and Computer Architecture of the Four-Legged Walking Machine BISAM. *IEEE Transactions on Mechatronics*, 4(1):1–7, März 1999.
7. Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23.

8. H. Cruse, J. Dean, U. Müller, and J. Schmitz. The stick insect as a walking robot. In *Proceedings of the 5th International Conference on Advanced Robotics: Robots in unstructured Environment (ICAR '91)*, volume 2, pp. 936–940, Pisa/Italien, 1991.
9. S.T. Davis and D.G. Caldwell. The biomimetic design of a robot primate using pneumatic muscle actuators. In *4th International Conference on Climbing and Walking Robots (CLAWAR)*, pp. 197–204, Karlsruhe, September 2001. Professional Engineering Publishing Limited.
10. B. Gassmann, K.U. Scholl, and K. Berns. Locomotion of LAURON III in rough terrain. In *International Conference on Advanced Intelligent Mechatronics*, Como, Italy, July 2001.
11. L. Jalics, H. Hemami, and Y. F. Zheng. Pattern generation using coupled oscillators for robotic and biorobotic adaptive periodic movement. In *Proceedings IEEE International Conference on Robotics and Automation*, pp. 179–184, 1997.
12. V. Kepplin, K. Berns, and R. Dillmann. Terrain and obstacle detection for walking machines using a stereo camera system. In *24th Annual Conference of the IEEE Industrial Electronics Society (IECON 98)*, 1998.
13. H. Kimura, Y. Fukuoka, Y. Hada, and K. Takase. Three-dimensional adpative dynamic walking of a quadruped robot by using neural system model. In *Proc. of the 4th International Conference on Climbing and Walking Robots (CLAWAR)*, Karlsruhe, September 2001. FZI.
14. K. Löffler, M. Gienger, and F. Pfeiffer. A biped jogging robot – theory and realization. In *Proc. of the 3rd International Conference on Climbing and Walking Robots (CLAWAR)*, pp. 51–58, 2000.
15. R.J. Bachmann, D.A. Kingsley. J. Offi, R.D. Quinn, G.M. Nelson and R.E. Ritzmann. Insect design for improved robot mobility. In R. Dillmann K. Berns, editor, *4th International Conference on Climbing and Walking Robots*, pp. 69–76. CLAWAR, Professional Engineering Publishing, September 2001.
16. Scholl, K.-U., Albiez, J., and Gassmann, B. (2001). Mca – an expandable modular controller architecture. In *proceedings of the 4th Linux Real Time Workshop*, Milano.
17. S. Schulz, C. Pylatuik, and G. Bretthauer. Walking machine with compliance joints. In *4th International Conference on Climbing and Walking Robots (CLAWAR)*, pp. 231–236, Karlsruhe, September 2001. Professional Engineering Publishing Limited.
18. H.-J. Weidemann, F. Pfeiffer, and J. Eltze. The six-legged tum walking robot. In *Proceedings of the 1994 IEEE International Conference on Intelligent Robots and Systems*, pp. 1026–1033, Munich, 1994.
19. H. Witte, R. Hackert, K. Lilje, N. Schilling, D. Voges, G. Klauer, W. Ilg, J. Albiez, A. Seyfarth, D. Germann, M. Hiller, R. Dillmann, and M.S. Fischer. Transfer of biological priciples into the construction of quadruped walking machines. In *Second International Workshop on Robot Motion and Control*, Bukowy Dworek, Poland, 2001.

Part IV

Sensors, Teleoperation and Telepresence

Results of Applying Sensor Fusion to a Control System Using Optic Flow

G. Martinez and V.M. Becerra

Department of Cybernetics, University of Reading, Reading, United Kingdom
sir00gcm@rdg.ac.uk,
v.m.becerra@rdg.ac.uk.

Abstract. The results from applying a sensor fusion process to an adaptive controller used to balance an inverted pendulum are presented. The goal of the sensor fusion process was to replace some of the four mechanical measurements, which are known to be sufficient inputs for a linear state feedback controller to balance the system, with optic flow variables. Results from research into the psychology of the sense of balance in humans were the motivation for the investigation of this new type of controller input. The simulated model of the inverted pendulum and the virtual reality environments used to provide the optical input are described. The successful introduction of optical information is found to require the preservation of at least two of the traditional input types and entail increased training time for the adaptive controller and reduced performance (measured as the time the pendulum remains upright).

1 Introduction

Maintaining the balance of an inverted pendulum is a well established problem in the study of adaptive systems. The pendulum is fixed by a hinge to a motorized cart that can move backwards or forwards along a track with the direction at any given time determined by a controller in response to measurements from the system at that time. The traditional inputs consist of the cart's displacement along the track, the angle between the pendulum and the vertical and the rates of change of those two values. The work described here consists of replacing some of these traditional inputs with novel inputs derived from the optic flow in the images registered by a camera fixed to the top of the pendulum.

The motivation for the novel use of visual information to balance a cart-pendulum system comes from research into the psychology of balance in humans. It has been demonstrated [8–10] that our sense of balance derives from the integration of several sense modalities: vision, mechanical information from the ankles, and the vestibular system. Experimental investigation with

human subjects showed that vision is the dominant modality [8–10], overriding inconsistent signals from the other modalities. These results were collected using a "swinging room" where the floor and the walls could be moved independently. A subject standing inside such a room would sway in response to small movements of the walls even when the floor remained still, i.e. visual information (the swaying walls) had more effect on their sense of balance than mechanical information (the stationary nature of the floor). Lee's experiments lead to the conclusion that the optic flow generated by an agent's egomotion contributed most to the subject's sense of balance. Furthermore Lee concluded that feedback from the ankle joints would be sufficient to provide the extra information needed to fully characterise the motion mathematically [19]. This work investigates these results from experimental psychology by using an inverted pendulum as a means of modelling bipedal balance. Reference [18] shows how bipedal systems can be modelled by a single inverted pendulum.

Sensor fusion is a process that aims to enhance both the reliability of the responses of a controller and its capacity to discriminate between different states of the system [23]. It is established methodology for augmenting the type and number of input signals to a controller. The move from mechanical information to optical information as input to the system reported here is not motivated by a need to improve the reliability of the existing controller. However, since sensor fusion provides a methodology for revising the nature and number of controller inputs it is used here to guide the process of moving to optical information.

2 System

The experiments were performed using a virtual reality simulation of a cart-pendulum system. The simulations were run in the Matlab 6.5 environment on a dual 2 GHz processor Intel Pentium Xeon workstation.

An example image from the virtual environment is shown in Fig. 1. In order to facilitate the processing of the optical information, the environment used contains many highly textured surfaces and has no moving elements other than the cart and the pendulum. The images taken by the virtual camera were in a SQCIF format with 128×96 pixels and a field of view of 40 degrees.

The cart-pendulum system consists of a cart that is free to move in one direction with the attached inverted pendulum able to swing around a hinge in the vertical plane through the same direction (see Fig. 2). The system was modelled using (1) and (2).

$$\ddot{\theta} = \frac{mg \sin\theta + \cos\theta(F + m_p l \dot{\theta}^2 \sin\theta)}{(4/3)\, ml - m_p l \cos\theta} \tag{1}$$

$$\ddot{x} = \frac{F + m_p l(\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta)}{m} \tag{2}$$

Fig. 1. The simulated pole in a virtual environment

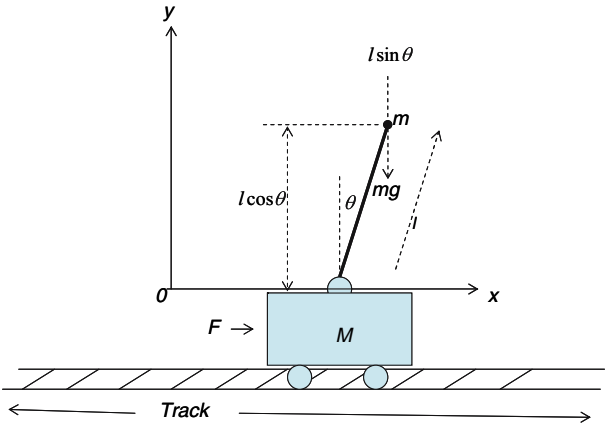


Fig. 2. The cart-pendulum system

where m is the combined cart and pole mass, g is acceleration of gravity, l is half of the length of the pole, m_p is the mass of the pole, F is Applied force, $\ddot{\theta}$ is the angular acceleration and $\ddot{x}$ is the linear acceleration.

The physical characteristics of the simulated cart-pendulum system were:

- Combined cart and pole mass = 1.1 kg

- Acceleration of gravity = $9.8\,\mathrm{m/s^2}$
- Half of the length of the pole = $0.5\,\mathrm{m}$
- Mass of the pole = $0.1\,\mathrm{kg}$
- Applied force = $\pm 10\,\mathrm{N}$

3 The Adaptive Controller

The movement of the cart is controlled by a bang-bang style controller which is assumed to cause a 10 N force to be applied to the cart in either direction. The direction to apply the force for a given input vector is determined by the adaptive part of the controller. This adaptive part consists of a CMAC (Cerebellar Model Articulation Controller) neural network that divides each dimension of the input space into possibly over-lapping associative units, and outputs a pair of probabilities describing in which direction the force should be applied. The CMAC learns an appropriate response to input vectors through a Q-learning process that penalizes any sequence of outputs that lead to the pendulum falling over [1, 21].

This controller is flexible and able to cope with different types and numbers of input variables. It has been used both with the traditional inputs applied to the cart-pendulum problem [6, 7, 15] and in the work reported here some of those inputs have been replaced by optical information.

Tables 1 and 2 summarise the default parameters for the artificial neural network (CMAC) and its learning rule. These parameters are kept for all experiments while no other value is stated.

Table 1. Default parameters for CMAC

Name	Value	Description
cmac.na	1	No. association units
cmac.pbits	12	Size of available memory = 2 pbits
cmac.np	2	Distance between each association units

Table 2. Q-learning values

Name	Value	Description
learn.alpha	0.50	Learning value
learn.beta	0.01	Noise added to the system
learn.gamma	0.85	Penalty, reinforcement learning value

4 Information Derived from Optic Flow

The optical information used as input to the controller was provided by the first order affine parameters of the optic flow field generated by the series of images taken from the top of the pendulum.

The optic flow field describes the movement of elements of a scene between two images as a plane of vectors. Each vector describes the size and direction of the movement of a particular picture element between one image and a subsequent image. The first order affine parameters of the flow field summarise all the vectors in just six values: translation of linear velocity (along axis x and y), dilatation (uniform expansion or contraction), rotation and shear (deformation, expansion in one axis and equal contraction in the other) [2, 19]. The optic flow was computed with a gradient-based algorithm as this is the fastest available algorithm for computing the particular affine values that were used here. (The other affine parameters were specifically excluded because the chosen algorithm could not determine them reliably. Although a more reliable matching-based algorithm was available, it was not as fast and lead to an unacceptable degradation of the performance of the simulations). See Tables 3, 4 and 5 for the default values used.

Table 3. Default input values

Inputs	Limit	Quantization	Description
x	1.20 m	3	Cart position (limit of the track to each side)
$\dot{x}$	0.50 m/s	3	Cart velocity
θ	1.00 rad	2	Pendulum inclination angle
$\dot{\theta}$	0.88 rad/s	3	Angular velocity
V_x	5	0	Relative displacement, X
V_y	2	0	Relative displacement, Y
dilation	0.01	0	Expansion or contraction
rotation	0	0	Rotation
shear1	0	0	Deformation 1
shear2	0	0	Deformation 2

5 Training Procedure

The controller operated at 50 input-output pairs per second. A single trial was deemed to have ended when the either the pendulum angle was at greater than 0.21 radians to the vertical or the cart reached one end of the track. Q-learning was then applied to penalize the most recent sequence of actions.

Table 4. Gradient Optic Flow Algorithm default values

Name	Value	Description
sigmaG	5	Mask standard deviation
mGaG	10	rotationally symmetric Gaussian low pass filter, mask size
rowsSampG	8	Number of samples for each row
colsSampG	8	Number of samples for each column

Table 5. Matching Optic Flow Algorithm default values

Name	Value	Description
sizeWin	3	Searcher windows size
radXse	3	Maximum expected displacement in X
radYse	3	Maximum expected dis placement in Y
rowsSampM	10	Number of samples for each row
colsSampM	10	Number of samples for each column
overlapX	1	Overlapping factor for rows
overlapY	1	Overlapping factor for columns
sigmaM	3	Mask standard deviation
mGaM	5	Gaussian mask size

6 Results of the Sensor Fusion Process

The first stage of the sensor fusion process was to add the six optic flow parameters to the four mechanical values as inputs to the adaptive controller. With the combination of these 10 inputs the adaptive controller was unable to learn to balance the pendulum for a significant period of time.

In order to determine which subsets of optic flow parameters could be successfully combined with the 4 mechanical inputs various combinations were tried. Table 6 shows the maximum time the pendulum remained upright using different combinations of inputs.

The elimination of shear values was found to improve performance. This was because the inclusion of the shear values entailed a large increase in the size of the input space. Hence, the process of reducing the set of inputs by eliminating particular values began with the adaptive controller receiving eight inputs: the four mechanical inputs and, from optic flow information, both linear velocities, dilation and rotation. The combinations investigated were as follows:

(1) Position on the track (x) + cart velocity ($\dot{x}$) + both linear optic flow velocities + dilation (not shown in Table 6)
(2) Position on the track (x) + cart velocity ($\dot{x}$) + angle of the pendulum to vertical (θ) + both linear optic flow velocities + dilation (not shown in Table 6)

Table 6. Maximum time that the pendulum remained balanced

Input Variables	Time
4 mechanical values	5.5 hours
4 values + dilation	5.5 hours
4 mechanical values + 2 linear optic flow velocities + dilation (learning rate = 0.5). Gradient algorithm	18 sec.
4 mechanical values + 2 linear optic flow velocities + dilation (learning rate = 0.2). Gradient algorithm	70 sec
4 mechanical values + 2 linear optic flow velocities + dilation (learning rate = 0.2)	27 sec
4 mechanical values + 2 linear optic flow velocities + dilation + rotation	17 sec
All 10 values	7 sec
2 mechanical values (x and θ) + 2 optic flow linear velocities + dilation	33 sec
2 mechanical values (x and θ) + 2 linear optic flow velocities	77 sec

(3) Position on the track (x) + angle of the pendulum to vertical (θ) + both linear optic flow velocities + dilation. (shown in Table 6)
(4) Position on the track (x) + angle of the pendulum to vertical (θ) + both linear optic flow velocities (removal of dilation improved performance, as shown in Table 6)

As Table 6 shows, this final combination was sufficient to keep the pendulum balanced for 77 seconds.

7 Discussion

These results show that it is possible to replace some of the inputs traditionally provided to the controller of a cart-pendulum system with optical information, thus providing a partial confirmation of results from the literature on the psychology of humans' sense of balance, and how these results can be translated to the case of balancing an inverted pendulum.

The displacement of the cart along its track has proved to be a necessary input value to prevent the adaptive controller from attempting to move the cart beyond the end of the track. However, this type of information is unlikely to be available to biological systems (such as humans) since it represents an absolute measurement. It would be desirable to use only the angle of the pendulum to the vertical and its rate of change, since these can be thought of as analogous to feedback from the nervous system about the state of a biped's ankle joint.

8 Further Work

Future work on this project will focus on possible ways of removing the displacement of the cart along its track as an input value. This value lacks biological plausibility and by analogy with the implied role of mechanical information from the ankle in human balance it would be desirable to use only information about the angle of the pendulum to the vertical. Possible solutions currently under consideration include implicitly constraining the cart to the length of the track or limiting the acceptable angle of the pendulum to the vertical during the earlier stages of training to reduce its movement along the track. The final goal of the project is to balance a real pendulum which has already been designed and built.

9 Conclusions

The results described here show that the first order affine parameters of the optic flow field can be successfully used as inputs to a CMAC neural network based controller that balances an inverted pendulum when two additional measurements are available. The optic flow field is obtained from images captured by means of a camera located at the top of the pendulum. See [11, 19] for more details of why additional information is needed to fully solve first order optic flow equation. The cart displacement and the pendulum angle from the vertical were used to provide this additional information. The process of sensor fusion described in this paper was successful. Future work will seek to replace the cart displacement with a more biologically plausible type of input. One further step will be to carry out experiments with a real pendulum.

References

1. Almeida P, Godoy M (2001) Fundamentals of a Fast Convergence Parametric CMAC Network. IEEE-INNS International Joint Conference on Neural Networks, Proceedings of IJCNN 2001. OmniPress, v. 3, pp. 3015–3020, Washington DC
2. Barron JL, Beauchemin SS, Fleet DJ (1994) On Optical Flow. 6th Int. Conf. on Artificial Intelligence and Information-Control Systems of Robots, Bratislava, Slovakia, Sept. 12–16, pp. 3–14
3. Boers EJW, Kuiper H, Happel BL (1993) Designing Modular Artificial Neural Networks. Proceedings of Computing Science in The Netherlands
4. Dean T, Basye K, Shewchuk J (1992) Reinforcement Learning for Planning and Control. Machine Learning Methods for Planning and Scheduling Book, ed. S. Minton, M. Kaufmann
5. Finton DJ, Hu YH (1995) Importance-Based Feature Extraction for Reinforcement Learning. Thomas Petsche, et al., editors, Computational Learning Theory and Natural Learning Systems, Volume 3, Chapter 5, pp. 77–94, MIT Press

6. Gomez F, Miikkulainen R (1998) 2-D Pole Balancing with Recurrent Evolutionary Networks. Proceedings of the International Conference on Artificial Neural Networks (ICANN-98, Skovde, Sweden), 425–430. Berlin
7. Hougen D, Fischer J, Johnam D (1994) A neural network pole-balancer that learns and operates on a real robot in real time. In Proceedings of the MLC-COLT Workshop on Robot Learning, pp. 73–80
8. Lee DN (1976) A theory of visual control of braking based on information about time-to-collision. Perception, 5(4), 437–459
9. Lee DN (1974) Visual information during locomotion. In: R.B. McLeod & H. Pick (Eds.) Perception: Essays in honor of J.J. Gibson (pp. 250–267)
10. Lee DN (1980) The optic flow field: the foundation of vision. Philosophical Transactions of the Royal Society of London, B 290, 169–179
11. Martinez G, Becerra V, Juarez J (2003) Investigation into the performance of an algorithm for deriving concise descriptions of optic flow fields for real-time control. 6th International Conference, Climbing and Walking Robots and the Support Technologies for Mobile Machines, Catania, Italy
12. Michie D, Chambers RA (1968) BOXES: An experiment in adaptive control. Machine Intelligence 2, E. Dale and D. Michie, Eds. Edinburgh: Oliver and Boyd, pp. 137–152
13. Miller WT, Glanz FH (1996) UNH_CMAC Version 2.1 The University of New Hampshire Implementation of the Cerebellar Model Arithmetic Computer – CMAC. Report
14. Miller WT, Aldrich C (1990) Rapid Learning Using CMAC Neural Networks: Real Time Control of an Unstable System. IEEE, TH0333-5/90/0000/0465
15. Pasemann F, Dieckmann U (1997) Evolved eurocontrollers for pole-balancing. Neuroscience to Technology, Proceedings International Workshop on Artificial Neural Networks, Lanzarote, Canary Islands, Spain, pp. 1279–1287
16. Smith RL (1998) Intelligent Motion Control with an Artificial Cerebellum. PhD Thesis, University of Auckland, New Zealand
17. Szabó T, Horváth G (2000) Improving the generalization capability of the binary CMAC. Proc. of the International Joint Conference on Neural Networks, IJCNN'2000. Como Italy. Vol. III. pp. 85–90
18. Vukobratovic M, et al. (1990) Biped locomotion: dynamics, stability, control, and application. Scientific fundamentals of robotics 7, Berlin; London: Springer
19. Young D (1994) First-Order Optic Flow. CSRP, no 313
20. Onat A (1998) Q-learning with recurrent Neural Networks as a Controller for the Inverted Pendulum Problem. The Fifth International Conference on Neural Information Processing, pp. 837–840, October 21–23
21. Sutton R, Barto A (1998) Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, Bradford Book, pp. 143–152
22. Brooks RR, Iyengar S (1997) Multi-Sensor Fusion: Fundamentals and Applications with Software. Simon & Schuster Publisher
23. Gros XE (1997) NDT Data Fusion. Elsevier Publisher, pp. 5–36

Novel Method for Virtual Image Generation for Teleoperation

R. Chellali, C. Maaoui, and J.G. Fontaine

(*) Ecole des Mines de Nantes-IRCCYN
4 rue A. Kastler 44000 Nantes
(+) Ecole Nationale Superieure d'Ingenieurs de Bourges
10 Boulevard Lahitolle 18020 Bouregs, Cedex France
ryad.chellali@emn.fr,
choubeila.maaoui@ensi-bourges.fr,
jean-gyu.fontaine@ensi-bourges.fr

1 Introduction

Tasks carried out remotely via a telerobotic system are typically complex, occur in hazardous or hostile environments and require fine control of the robot's movements. To operate effectively in the remote environment, the operator requires sufficient visual information to be able to interpret the remote scene and undertake the task effectively and efficiently.

User interfaces for teleoperated robots started basically with live video image coming from cameras. However, for an unknown environment, it is necessary to the operator to explore the environment visually before moving the robot. To do so, images issued from different points of view of the unknown environment are needed. Unfortunately, this is not always possible: remote cameras are only in Pan-Tilt control.

To facilitate the teleoperation process, we designed a new approach enabling the automatic generation of virtual image corresponding to new viewpoints from a minimal set of images acquired by static and uncalibrated cameras.

In this paper, we address the problem of synthesizing a new image from an arbitrary viewing position given two or three reference images without any 3D information about the scene and exploiting the constraints from imaging geometry. These geometry-based methods are often referred to as a reprojection or transfer method.

First, we will present an overview of methods allowing the generation of a new view, from existing images, which are the epipolar and trifocal transfers. Then, we will propose a new approach to generate a synthetic image based on Desargues configuration, knowing 6 points of scene, of which four

are coplanar. Finally, several examples on real and synthetic images are presented to illustrate the significant results.

2 Epipolar Transfer

Epipolar geometry is the basic constraint, which arises from the existence of two viewpoints. Faugeras et al. [1] describe how the epipolar geometry between a third view and each of the first and second views predicts the location and properties of corresponding elements in the third view. For a given element in one image the epipolar constraint reduces the possible location of the correspondence in another view from the whole image to a line in the image. Epipolar transfer exploits the situation where we have two views in correspondence and wish to produce a third view for which we know the fundamental matrixes relative to the originals images [2]. A correspondence between original images (p_1, p_2) constraints the point in the third image p_3 to lie on the lines $F_{13}p_1$ and $F_{23}p_2$ (Fig. 1). The point in the third view must be the intersection of these two epipolar lines and is given by:

$$p_3 = F_{13}p_1 \times F_{23}p_2 \tag{1}$$

Assuming, we know the dense correspondence between two original views and the required view defined in terms of fundamental matrices [2], a novel view can be produced by transferring all corresponding pixels to the new view using the (1). Given the fundamental matrix between the original views we can compute the fundamental matrices required for the epipolar transfer given the desired motion (R,t) (Fig. 2) [2].

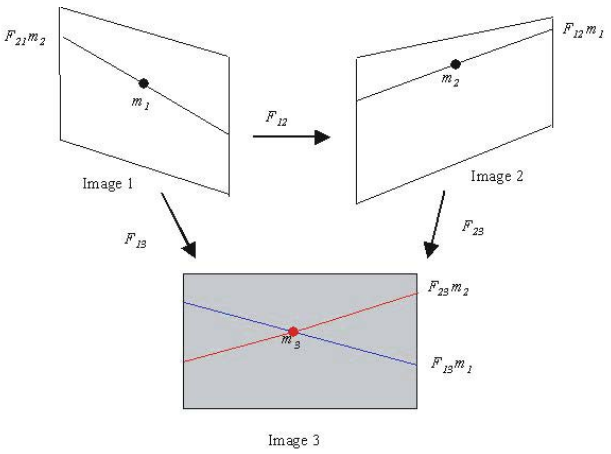


Fig. 1. Epipolar transfer

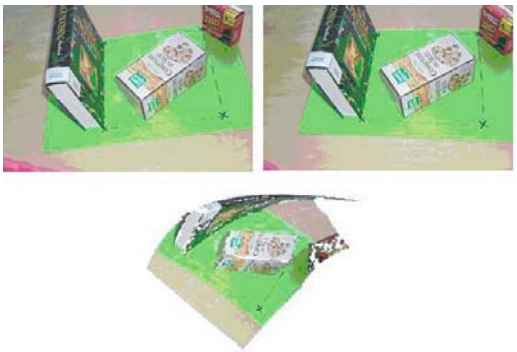


Fig. 2. Epipolar transfer of real images

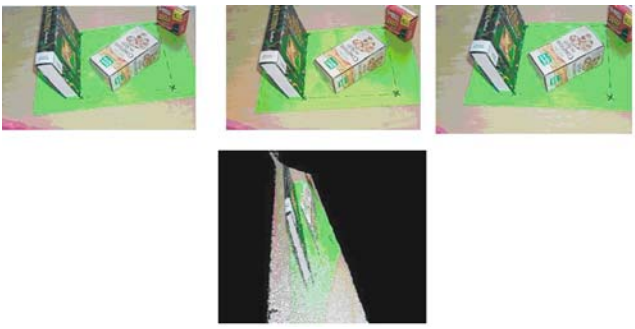


Fig. 3. Virtual image generation using trifocal tensor: *Top row*: original images (1, 2, 3). *Up row*: virtual image generated

The degeneracy case in epipolar transfer is when the epipolar lines are coincident (parallels) then the intersection is not defined [3]. This case occurs when the *trifocal plane* (the plane defined by the three camera centers) intersects the new image plane.

3 Trifocal Transfer

Novel view synthesis by trifocal transfer was introduced by Avidan and Shashua [4]. They have developed a method, which allows a parameterization of trifocal tensor with the displacement of virtual camera. Given two real images (1 and 2) and the tensor for views 1, 2, 3 then the transfer is accomplished by transferring all corresponding pixels to the new view, view 3. One way of obtaining the required tensor is by manually identifying number correspondences across all three views. The transfer is accomplished using the tensor constraint (2) to solve for the point p_3 given a correspondence points $p_1 \leftrightarrow p_2$.

$$p_3{}^k = p_1{}^i l_j{}^2 T_i^{jk},\ p_2{}^j l_j^2 = 0 \tag{2}$$

Where T_i^{jk} is the trifocal tensor defined for views 1, 2, 3.

For each point in the left image p_1 and a line l_j through its corresponding point p_2 use the transfer equation to obtain p_3. The trifocal tensor does not suffer from the singularities evident in epipolar transfer.

Avidan et al. [5] have also defined a new tensor for two distinct views named a seed tensor. This tensor is simply an extension of the fundamental matrix and is the tensor for the views (1, 2, 2). Given a pose for the new view (R, t) the relevant tensor is computed by updating the seed tensor:

$$\beta_i^{jk} = R_l^k \alpha_i^{jl} - t^k\ R_i^{'j} \tag{3}$$

Where α_i^{jl} is the seed tensor (generalized fundamental matrix), β_i^{jk} is the tensor valid for views (1, 2, 3) where view 3 is rotated and translated relative to view 2 by (R, t). The rotation matrix, R', between the two real views is required and can be obtained by standard decomposition of the fundamental matrix [2]. Therefore, given the dense correspondence between two real views and their fundamental matrix, the tensor for a novel camera pose can be computed.

4 Desargues Configuration

In this section, we present a new method for computing corresponding points using Desargues theorem starting from three images, knowing 6 points of which four are coplanar.

4.1 Desargues Theorem

"If the three straight lines joining the corresponding vertices of two triangles ABC and $A'B'C'$ all meet in a point O, then the three intersections of pairs of corresponding sides lie on a straight line".

We applied this configuration for 3 images and a reference plane Π.

Let C_1, C_2, and C_3 be the respective optical centers of three cameras. Let M be an observed point of the scene and $m\,(u, v)_{i;i=1,3}$ its three projections on the three images. The projections of $m\,(u, v)_{i;i=1,3}$ on the reference plane according to three optical axes are the points M_Π^i. The two triangles $Tr\,(C_1, C_2, C_3)$ and $Tr\left(M_\Pi^1, M_\Pi^2, M_\Pi^3\right)$ are in Desargues configuration.

The lines $(C_i M_\Pi^i)$ have a common intersection at the point M, the couples $C_i C_j$ and $M_\Pi^i M_\Pi^j$ are intersected at the respective points P_Π^{ij}, which are aligned. This alignment is the intersection line of the plane (C_i) with the plane Π.

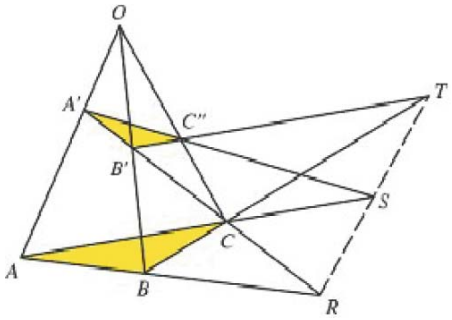


Fig. 4. Desargues configuration

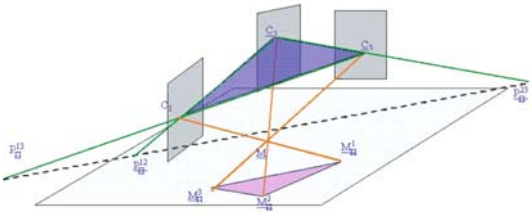


Fig. 5. Desargues configuration of three images

To find the points M_{Π}^{ij}, it is enough to know the respective projections of four reference points of the plane on images $Im_{i,i=1...3}$. Indeed, we can define the 3 homographies $H_{i,i=1...3}$ [6] by establishing the correspondence between the points of the plane and the point images.

M_{Π} represents the image homography of the point $M \in Im$ on the plane Π.

4.2 Desargues Invariant

Let P_1 be a new point in the previous configuration. P_1 also product a triangle on the reference plane. This triangle is in Desargues configuration with the 3 optical centers. We can demonstrate that the two triangles $Tr(M)$, $Tr(P_1)$ are in Desargues configuration (Fig. 6). It can also be noted that the intersection O of the line (MP_1) with the reference plane Π is Desargues top of the both triangles. Point O is invariant of the configuration. For the two points P_1 and M, O is independent of the position of the cameras. This property is very important and it has two interesting practical consequences:

1. If we know the point P_1, its homographic triangle and the triangle of an unknown point M, we can determine the position of the point M, which is the line OP_1.

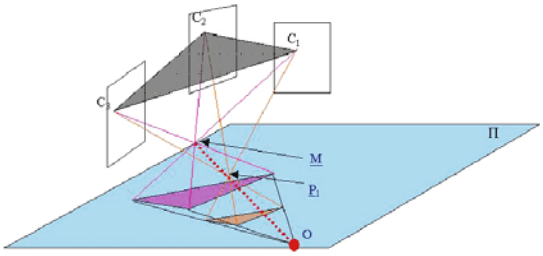


Fig. 6. Desargues invariants

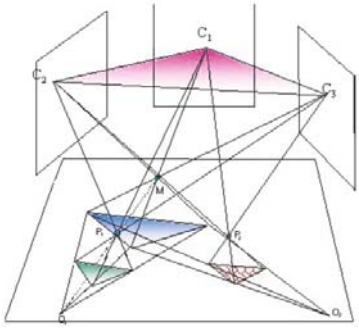


Fig. 7. Localization of point M with Desargues theorem

2. Conversely, if we know the point P_1, its homographic triangle and the point O, then any point M of the straight line OP_1 will have its triangle in Desargues configuration with $Tr\,(P_1)$.

5 Novel Image Generation with Desargues Theorem

In this section, we develop a new approach to synthesis new image using Desargues configuration from three existing images.

Assuming, we know the image of two reference points P_1 and P_2 of different heights, and 4 coplanar points of a reference plane in 3 images. The projection of the points P_1 (resp. P_2) on the reference plane are p_1, p_2 and p_3 (respectively p'_1, p'_2 and p'_3) from the three optical centers. The triangle $Tr(P_1)$ (resp. $Tr(P_2)$) are in Desargues configuration with the (C_1, C_2, C_3) triangle. These two triangles are known. The projection of given point of image i on the reference plane is computed using the corresponding homography (H_i) [6]. The homography is calculated using the four coplanar points of reference.

Let $Tr(M)$ be a triangle of unknown 3D point M. $Tr(M)$ is constituted from corresponding point in three images $m_1 \leftrightarrow m_2 \leftrightarrow m_3$. This triangle is in Desargues configuration with the two triangles $Tr(P_1), Tr(P_2)$. Therefore, there are two points O_1, O_2 defined by:

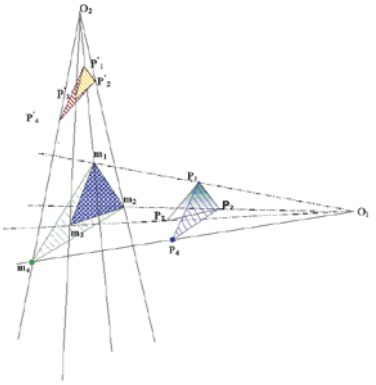


Fig. 8. Localization of the point M in the new image

$$O_1 = \cap(m_i p_i), O_2 = \cap(m_i p'_i) \quad i = 1:3 \tag{4}$$

O_1 is the intersection point of the line $(m_1 p_1), (m_2 p_2)$ and $(m_3 p_3)$.
O_2 is the intersection point of the line $(m_1 p'_1), (m_2 p'_2)$ and $(m_3 p'_3)$.

The points O_1 and O_2 are invariant for the point M, and don't depend on the position of cameras. In the new view, image 4, the points P_1, P_2 generate two news triangles $Tr'(P_1)$ and $Tr'(P_2)$. Also the point M generates a new triangle $Tr'(M)$ in Desargues configuration with previous triangles. So it is constituted from the two points (m_1, m_2) and the new point m_4, which is the intersection point of the line $O_1 p_4$ and $O_2 p'_4$ (Fig. 8). Knowing the projection of reference points in any new image, the projection of point M in the new image is the intersection point of the line $O_1 p_{new}$ and $O_2 p'_{new}$.

The Fig. 9 shows some results obtained with synthetic and real images.

6 Conclusions

In this paper, we have presented the transfer methods allowing the generation of new image from an arbitrary viewing position from images existing. These methods used only the geometric constraints, without 3D reconstruction. The epipolar transfer is used to generate a new image from two original images. This method fails when the cameras are parallel. However, the trifocal tensor is used two generate a new image from two or three originals images and it don't suffer from the singularities problem.

We have also proposed a new method of synthesizing of new image based on the Desargues configuration. Knowing 6 reference points of which 4 are coplanar and two points of different heights, we can compute the image of given point in the new view, by using a very simple principle which is the intersection of the straight lines. This approach gives good results and it is less sensitive to the noises than the classical methods.

Fig. 9. Results of generation of new image using Desargues theorem

All methods depend of the quality of the correspondence dense.

References

1. Faugeras O, Robert L (1994) What can two images tell us about a third one? In: Proc. of 3rd European Conference on Computer Vision, Stockholm, Sweden, pp. 485–492
2. Maaoui C, Chellali R (2002) Virtual Image for Teleoperation. ISMCR. Bourges, June 2002
3. Hartley RI, Zisserman A (2000) Multiple View geometry in Computer vision. Cambridge University Press
4. Avidan S, Shashua A (1998) Novel view synthesis by cascading trilinear tensors. IEEE Transactions on Visualisation and Computer Graphics, 4(4)
5. Avidan S, Shashua A (1997) Novel view synthesis in tensor space. Conference on Computer Vision and Pattern Recognition (CVPR '97). Puerto Reco. June 1997
6. Loop C, Zhang Z (1999) Computing Rectifying Homographies for Stereo Vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, CO Vol. 1, pp. 125–131, June 23–25, 1999

Intelligent Technical Audition and Vision Sensors for Walking Robot Realizing Telepresence Functions

V.E. Pavlovsky[1], S.A. Polivtseev[2], and T.S. Khashan[2]

[1] Keldysh Institute of Applied Mathematics of RAS (KIAM), Moscow, Russia, vlpavl@keldysh.ru
[2] Institute of Artificial Intelligence (IAI), Donetsk, Ukraine, {psa, tsk}@iai.donetsk.ua

Abstract. The paper presents audition and vision sensors, which serve as special locators for searching objects by mobile robot. These sensors provide robot with a data of objects locations during the mission where the robot has to explore the surroundings, find objects and run from one of them to another. Main aim of that sensors elaborating is realizing the mobile (walking) robots with telepresence support. The robot schemes, the structure of a sensors and mathematical features of objects searching technique are presented. Main focus of a paper is an AI-based algorithm for pelengating objects and localizing them as well.

The research was prepared under the partial funding of RFBR grants 02-01-00750, 01-01-00079.

1 Introduction

Development of modern mobile robots requires the elaboration of a set of different intelligent sensors to provide the collecting complete "picture" of surroundings. As main information channels of human beings are vision and hearing (it is known that vision provides about 80% and hearing provides approximately 20% of sensing data), it is reasonable to elaborate technical vision and audition systems for robots. This is especially important for robots that have to explore some unknown or complicated areas. Such complicated tasks and missions may occur, for example, due to some natural or man-caused collisions, or may arise during collection of research data in those zones where the human presence causes troubles or, more, is strongly impossible. Therefore, the robots are intended for executing exploration of such areas and realizing telepresence functions like transmitting essential data to the distant control consoles.

As it is mentioned above the main focus of research is elaborating intelligent audition and special vision sensors for mobile robots for realizing

telepresence functions. Walking chassis is used for ensuring high adaptation for that environment where the robot has to move. On elaborating the robot concept it was assumed that robot has moved in automatic or remote control mode to the complicated area with complicated obstacles (or, for example, to the contaminated zone) and transmits telemetric data to the remote console from that zone. It is assumed that main components of telemetric data are the sound "pictures" and special images of the surroundings, or results of processing of that primary data. Accordingly, robot is equipped with the systems of technical audition and vision. Transmission of such data to the remote console (computer) is executed via radio line. It is assumed that first of all the robot has to find "interesting" objects in the area of exploration. So, for solving such task the vision and audition sensors will serve as special locators capable to find objects and localize them. The paper deals with elaboration of prototypes of all described devices – walking chassis itself, on-board control system, intelligent technical audition and vision systems. The main attention will be paid to the algorithms of technical audition and vision systems. For tests wheeled robots were used as prototypes and sensors carriers as well.

Further let's introduce some definitions. The process of determining the direction to the object will be treated as pelengation of an object, otherwise the determining of object coordinates will be treated as object localization. The system of sensors which are under elaboration is intended for searching different objects in the examinated area. In the work it is assumed that searching may be produced by vision sensors (searching some landmarks or outlined objects), by special vision system to operate with infrared (IR) data (searching some sources of IR radiation), or by technical audition system (searching sound sources). The common feature of those sensors is using ray concept in the objects pelengation process. As there are problem of finding real objects in the set of virtual points that are points of pelengation rays intersection special AI-based algorithm is proposed for solving the problem.

So, the paper deals with a problems of elaborating prototypes of robots for area exploration, of elaborating vision and audition sensors for those robots, and with algorithms for recognizing absolute positions of vision or audio objects located inside the operating zone of the mobile robot. Structure of video, IR sensor, audition sensors which solve the problem, algorithms themselves, and results of mathematical and real experiments will be presented in the paper. Nevertheless, the main attention of the paper will be paid to the development of algorithms for such sensors.

2 The General Description of Robots

Walking chassis. Two types of walking chassis were elaborated. One robot uses the six-legged walking chassis (KIAM, [1]). That chassis is a six-legged walking machine, having a body of approximately 1 meter in length. All legs of the robot are the same and have 3 controlled DOFs. Legs of the robot possess

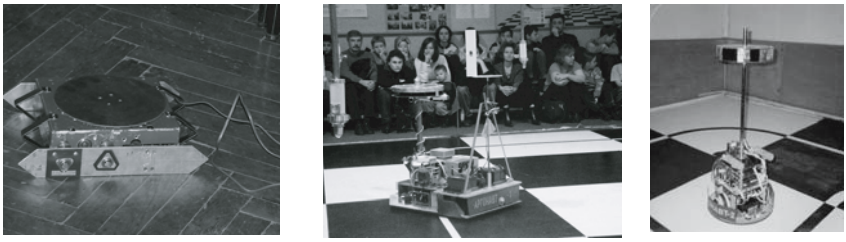


Fig. 1. Small walking robot and "Argonaut-1", "Argonaut-2" mobile robots

a so-called insectomorphic kinematics. In addition another walking robot was elaborated at IAI. It is small machine especially intended for working as a member of a team of such robots. It is shown in the Fig. 1 (photo at the left). That robot has only two "legs" and a special device for turning (special disks shown on the photo), with the help of this device the robot is able to turn around "on the spot". It is fully symmetrical referring to the middle horizontal plane, so it is able to move in regular manner and also after overturning. The walking chassis carry several sensors, among them there are small vision and audition sensors. Those sensors, especially for hearing, are mounted inside the robot body.

Mobile wheeled robots as carriers for elaborated sensors carrying are under research as well. They are also shown on a Fig. 1. The robot "Argonaut-1" (shown in the central picture) has 3 wheels, two wheels are active and one is free passive wheel, so robot has 2 electrically driven active DOF. Those wheels are driven independently and robot is able to go forward or back and able to rotate in any direction. It is also possible to set up speed of rotation of each wheel independently. The robot is shown while it participates in a robotic competition in the frame of International Festivals "Mobile robots" at Moscow State University. The 3-rd picture in the Fig. 1 shows the "Argonaut-2" robot. It has analogous scheme as "Argonaut-1" but is smaller. The photo shows the robot without its special "dress". The robot carries audition sensors with acoustic horns, they are two light cylinders mounted on the upper platform of a robot. Inside the horns the microphones are mounted, via ADC card they are connected to the central CPU of a system (MicroPC^{TM} cards from Octagon Systems^{TM}, USA, are used). The central CPU executes the set of parallel processes and one of them implements the sensor software.

Control system. As a whole, the control system of the robots will be elaborated in according to the client-server technology that is developed at KIAM for controlling mobile robots. The control system is implemented as two main modules – an on-board system (the control server) and a console application (client application), the last one is executed on the same on-board computer or on a remote computer. The control server and client application are connected via a wire or a wireless (radio) line, the system supports a number of different network protocols. As a main variant, the TCP/IP is used. For example, the

connection of a server and control client application may be realized via Internet with the radio-Ethernet end segment. The console application displays telemetric data of a remote robot and transmits commands to the robot, those commands may be of low-level type, or may be of high-level type. The solution for designing the control system hardware is a following. The system typically includes two computers and those computers are combined into onboard control computer network. The host computer of a network is PC-compatible high-reliable machine. This computer is intended for searching objects and for route planning and equipped with 8/16-channel ADC card for controlling the sensors. On a low level of control network modular control computer built on a 16-bit INTEL-80196 processor is used. System for motors (drives) control is digital one and implements optical encoders to obtain data of drives position and speed. The control system includes also PWM modules (cards) equipped with MOSFET H-bridge units to control wheels rotations.

Technical vision and audition systems. The technical vision system of the robot is implemented as a set of TV-cameras (the system supports from 1 to 4 cameras) and a special videoprocessor. The prototyped videoprocessor subsystem is composed of several cards, including a high-performance embedded CPU, a 4-channel framegrabber card, and a communication card (to provide communication to the central on-board computer). All cards are compact embedded devices, the videoprocessor and central computer use cards of a MicroPC^{TM} format from Octagon Systems^{TM}. The videoprocessor realizes algorithms of low-level image processing and several high levels of imaging from cameras. In particular, the videoprocessor may calculate the 3D-coordinates of distinctive objects that are located in a field of vision of a robot and is able to transmit that coordinates to the remote console computer. To calculate coordinates it executes algorithm, which is described below.

The "Argonaut" robots are equipped with IR-sensors to investigate methods for finding IR-objects. Those sensors are mounted on a mast on main desk of robots (as it is shown on a Fig. 1). There are also mounted on a robot desk such units as computer control system and power supply onboard system.

The technical audition system is realized by the set of microphones additionally equipped with special acoustic antennas and a software-hardware unit for audition data processing. The system is intended not only for transmitting the sound data to the remote console, but also for direction-finding to the typical sound objects, and for finding location of those objects in an area being investigated as well. The task is solved by special audition digital module, which is an element of intelligent audition sensor. First, in the frame of developing of this system main efforts were devoted to elaboration of the audition sensor itself and its digital module. Detailed computer simulation was performed, and experiments were made as well [3]. The sensor is a microphone equipped with a special acoustic antenna, which is a special horn device, on the set of experiments the form and sizes of horn antennas were chosen and optimized.

3 The Structure of Sensors Hardware Modules

As it was mentioned the aim of the sensor subsystem is to locate an object which lies somewhere around the robot. First, let's introduce here special IR-locator. This device is a sensor unit containing 8–12 detectors "looking" in different directions. Each detector works in an associated geometrical sector, detecting whether the object is within it. All sectors together cover whole horizontal plane, with some overlaps. So the hardware module tells the robot which sectors contain the object at any time. Note, that the regular case is the following: not one but two sectors "see" the object if there is one object outside the robot, and, in a case of two, or more, objects, two, three or more sectors can "see" the IR-radiation. For the case of 8 sensing detectors they are arranged into 8 independent measurement channels of identical electronic and optical structure shown on a Fig. 2 (left). Each channel is built of 4 main units including one part mounted on a robot mast as a "head part" of a sensor, and other part mounted on the robot platform. Each measuring detector (IR-diode) is mounted on a head in special box of sector-like shape bounded with non-transparent walls. The sensor is covered with a non-transparent cover to prevent a lot of disturbances of a very different nature.

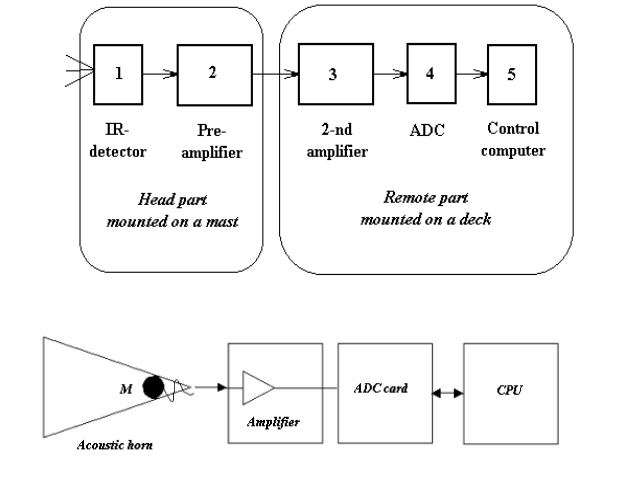


Fig. 2. Principal scheme of measurement channels of IR and audio sensors

It is also necessary to note that geometrical structure of measurements channels may not be identical. As it is shown on Fig. 3 below (central part), "Argonaut-1" robot has the following sensor – 6 front sectors (sectors organized into the 'front hemisphere of observing') are smaller than 2 rear sectors of a sensor. Ideally the observing angle of 6 front sectors is the same and is equal to 30°, and the observing angle of rear sectors is equal to 90°. On

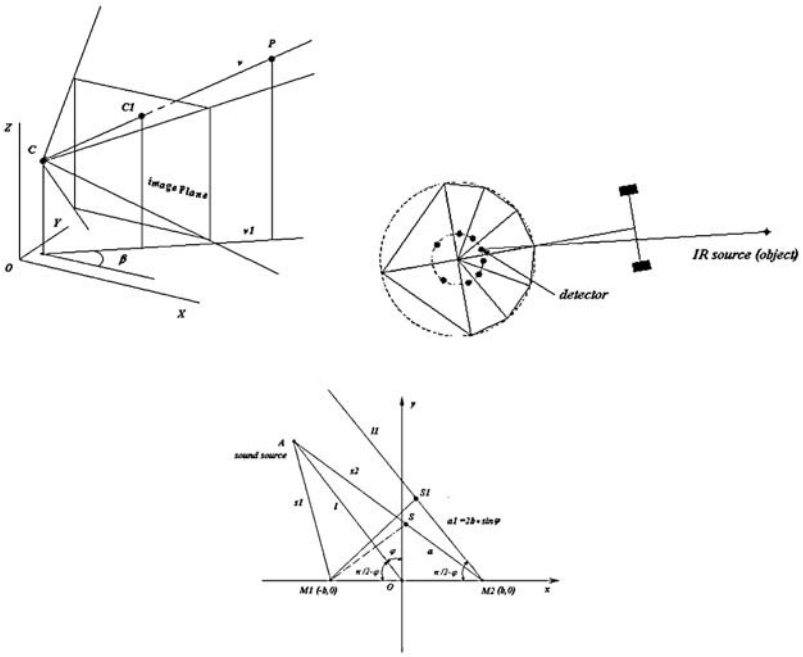


Fig. 3. Scheme of ray concept for vision, IR, and audition locators correspondingly

the contrary, the "Argonaut-2" has symmetrical sensor. In fact real values of sector angles slightly differ form those ideal values depending of manufacturing conditions of sensor building. The real values of observing angles were determined by special geometrical calibration procedures.

The measurement channel of an audition sensor is also shown on the Fig. 2 (right). It is necessary to mention that the special amplifier is needed to be installed in this measurement channel. The hearing system may include two such sensors, or a matrix of such sensors. Pair of "ears" (or more) is needed to find direction to sound object. The TV-vision sensor is, practically, traditional one and implement small TV-camera (cameras) of general type. On the first experiments those cameras were black-and-white ones.

4 The Sensor Data: Rays Concept

Of course, the detector information at any given moment doesn't contain proper information about the position of the object. All types of described sensors provide only rays (lines), which move from sensor to the object. Corresponding schemes are given on Fig. 3. For example, such situation is shown on a Fig. 3 (central), where objects "rising" and "setting" in the IR-locator are explained, on those events measuring of angle of ray is performed.

For all types of sensors the detector subprograms implement the specified idea collecting the ray information while the robot is moving, providing a list of rays found at any given time. For the first glance, it seems that it is possible simply calculate the common intersection point of rays. However, such points will be only virtual, and there is a problem of determining real objects in a set of virtual ones. Also, there are several possibilities for errors, which make the rays not to pass strictly through the same unique point that is not to pass over the object points. These error factors are the following:

1. Both objects and detectors are not points but have some physical dimensions;
2. The detector position in coordinate system related to the robot could not be measured with sufficient accuracy, instead there are some constant parameters describing robot geometry;
3. The robot position in absolute coordinate system is known with an error too, since the mechanics is not ideal and mechanical parameters are known also approximately;
4. The data are collected on discrete time intervals; as a result the system don't "know" the exact moment when the object left the sensor or rise in a sensor; the electronics also provides some delay;
5. There is a noise in a measurement channel, e.g. for IR-sensor it depends on infrared radiation nature (reflections and re-reflections, shadows from other objects, etc).

That's why the system has to use more intellectual algorithm for objects locating. Such algorithm named "grid algorithm" is described below. Let's note, that it could locate either separate object, or locate several objects simultaneously (the rays in such case intersect in a number of points, not in one point).

5 Calculating of the Objects Location: Probability Approach, Grid Algorithm

Let us first introduce the heuristic approach. We come here with the following task: find the object having a large amount of rays passing near to it. If this problem is given to a human, he will probably do the following: draw them on the sheet of a paper, and look for "bright" areas as objects. The "grid" algorithm is the implementation of this idea. Let's take a large matrix initialized with zeroes, and "draw" rays on it, incrementing values for cells through which the ray passes. Thus, for each cell it is possible to count number of rays passed through it, n_{ij}. Then the cells for which this number is large enough:

$$n_{ij} > \alpha \tag{1}$$

are treated as "bright" and organized into marked areas (see below). The boundary value α is chosen proportional to $\max_{i,j}(n_{ij})$. The relation (1) is treated as 1-th threshold for choosing the rays-"candidates" for determining the objects.

It is necessary further to note that this algorithm has a formal probability basis and it is possible to proof some foundation for the algorithm. Since there are errors in ray measuring and calculation it is not possible to treat measurements that the object lies *on* the ray, but it is possible to declare, that it is *probably near to it*. Mathematically this means a probability distribution $p(x, y)$ on the plane for object location, depending on distance from the ray

$$p(x, y) = p(d((x, y), l)) \tag{2}$$

decreasing when the distance $d((x, y), l)$ increases. This distribution, in its turn, induces distribution on grid, defining probability for cell Q_{ij} to contain an object:

$$\begin{aligned} p_{ij} &= \int\limits_{Q_{ij}} p(x, y)dx\,dy \\ &= \int\limits_{Q_{ij}} p(d((x, y), l))dx\,dy \approx p(d(Q_{ij}, l)) \end{aligned} \tag{3}$$

Then, if we have n rays, this probability should be multiplied:

$$\tilde{p}_{ij} = \frac{\prod_{k=1}^{n} p(d(Q_{ij}, l_k))}{D} \tag{4}$$

where D is normalization coefficient (sum of all probabilities must be equal to 1). By taking logarithms, we replace products by sums:

$$\begin{aligned} n_{ij} &= \ln \tilde{p}_{ij} \\ &= \sum_{k=1}^{n} \ln p(d(Q_{ij}, l_k)) - \ln D = \sum_{k=1}^{n} n(d(Q_{ij}, l_k)) - \ln D \end{aligned} \tag{5}$$

The relations (3) and (5) are continuous and discrete models of the basis of scheme of calculating objects locations. So the relation (5) is the main formulae implemented in the algorithm for searching the objects. Hence, if we treat "brightness" as logarithm of probability, the "bright" points will be the most probable "candidates" for object location. Note, that the implemented algorithm doesn't subtract $\ln D$, since it doesn't affect condition (1). Let's us also note, that the idea to take logarithms is reasonable by itself: the products will grow exponentially and lead to calculation overflow, or it will be necessary to normalize the calculation tables, which is rather slow operation. Note also that incrementing n_{ij} for that cells, the ray passes through, corresponds to distribution concentrated in the cells. More realistic distributions could be

implemented by addition of some complex pattern along the ray, affecting also the adjacent cells, for example, it is possible to use the following pattern shown on a Fig. 4. Numbers in cells are some weights adding to cells containing the rays. In the center of a pattern there is located a point with maximal value of a weight.

1	1	1
1	**3**	1
1	1	1

Fig. 4. Probability pattern for ray processing

This means that system will add 3 to each cell on ray, and 1 to all cells adjacent to it (in fact, all points on ray are incremented by 5, that is there will be added +3 by the point/cell itself, +1 by next cell, and +1 by previous cell). The corresponding example of 3 intersecting rays is shown below on Fig. 5. The cells with maximal total weights "15', "16" and "17" will be treated as object location.

0+2+0=2	2+2+0=4	5+1+0=6	3+0+0=3	1+0+0=1	0+0+0=0	0+0+0=0	0+0+0=0
0+5+0=5	2+6+0=8	6+4+0=10	7+3+0=10	3+2+0=5	1+1+0=2	0+0+1=1	0+0+2=2
0+3+0=3	1+6+0=7	3+6+0=9	7+6+0=13	6+6+1=13	2+4+2=8	0+2+4=6	0+1+6=7
0+1+0=1	0+2+0=2	1+3+1=5	4+4+2=10	7+6+4=17	4+7+6=17	1+6+7=14	0+4+6=10
0+0+1=1	0+0+2=2	0+0+4=4	2+1+6=9	6+2+7=15	6+4+6=16	2+6+4=12	0+6+2=8
0+0+2=2	0+0+5=5	0+0+7=7	1+0+6=7	3+0+4=7	6+1+2=9	3+2+1=6	1+3+0=4

Fig. 5. Example of grid state

Now let's turn to area creation. It is desirable to choose that points, which are "neighbors" in some sense, and unify them into areas. Then, if these areas are small enough, center of each will be declared as an object. Otherwise, the system will treat the situation as the accuracy is not achieved, and will wait for obtaining more numbers of rays. The algorithm presented, in fact, solves this task in some "opposite" way. First it creates areas upon criteria of "small-enough region", and later checks them for "neighborhood" property.

Let's now provide more detailed description.

1. *Initialization.*
   The system starts looking through the grid with empty set of areas.
2. First step: building of areas.
   Suppose, at some moment system has some built areas $S_1, \ldots, S_m$ and a new bright point p. If after adding it to some existing area S_k its size will be small enough, $\text{size}(S_k \cup p) < C_1$, it is necessary to set $S_k = S_k \cup p$, otherwise $S_{m+1} = \{p\}$. The operation "$\cup$" here also suggests taking of convex.
3. *Second step: testing of areas.*
   On the previous step some set of "small-enough" areas was obtained. Now it is necessary to check distances between them. If for some of them $\text{dist}(S_i, S_j) < C_2$, system will treat these areas as produced by a single object, whose position still could not be found with desired accuracy. Otherwise, the center of each area is declared as an object, with accuracy $C_1 \,.\, C_2$ should be chosen from condition $C_1 \ll C_2 \ll C_3$, where C_3 is minimal possible distance between objects. Practically, $C_1 \ll C_3$, and C_2 is easily chosen (operator "$\ll$" means "significantly less"). At the final stage of the algorithm the list of all objects around the robot will be set up.

6 Conclusion

The series of experiments were done on the stage of building the robots and during the tests. Those experiments show sufficiently high quality of proposed sensors and sensing algorithm. The total accuracy of object determining was enough for "touching" objects in the following sense. An absolute mean value of errors was equal approximately 10 cm on left-right angle (polar angle on achieving the object). In determining the distance the error was larger but it wasn't critical one. As a result sufficiently good quality of solving the exploration tasks was realized with the proposed devices and algorithm. It is possible to formulate the result, that passive sensors is able to provide adequate solution of problems of objects identifying being equipped with proper software system like those described above. The coordination of measurements and robot movements is necessary as well. It is planned that the elaborated sensor system will be mounted on board of other KIAM and IAI robots (e.g. see [2]).

References

1. Okhotsimsky D.E., Platonov A.K., Pavlovsky V.E., Serov A. Yu, Mishkin A.I. (2000) Study of Neural Control Algorithms for Walking Machine: Localization and Leg Motion Control. Proc. of 3rd Int. Conf. on Climbing and Walking Robots CLAWAR'2000. Madrid, Spain, 2–4 October 2000, pp. 355–362

2. Pavlovsky V.E., Shishkanov D.V., Alexeev A.D., Amelin S.V., Podoykin Yu N. (2002) Concept and Modeling of Legged-Wheeled Modular Chassis for High-Adaptive Rover. Proc. of 5-th Int. Conf. on Climbing and Walking Robots CLAWAR'2002. Paris, France, 25–27 September 2002, pp. 307–314
3. Pavlovsky V.E., Polivtseev S.A., Khashan T.S. (2004) Intelligent Technical Audution Sensors for Mobile Robots. Proc. of the Int. Workshop-Conference "Adaptive Robots – 2004", Russia, SPb, 2004, pp. 65–71. (in Russian)

Learning About the Environment by Analyzing Acoustic Information. How to Achieve Predictability in Unknown Environments? (Part II)

M. Deutscher, M. Katz, S. Krüger, and M. Bajbouj

Otto-von-Guericke – University of Magdeburg, Germany

Abstract. It is hard to describe in an analytical way all dynamical relationships of the ground and the robot. Unexpected environmental changes complicate a safe walking. Therefore it is obvious to develop a method what allows the evaluation of acoustic emissions depending on acting forces in the past and consecutive steps during walking. Practical investigations of steps on different ground deliver information which were recorded online and interpreted by an operator. The analytical evaluation of the relationship between robot body and the environmental reactions and a subjective interpretation delivers bad or good steps of these anticipations. That knowledge can be "leaved" for further explorations. The fusion of our data and evaluation of these relationships were done by our SVM (Support Vector Machine) a fast statistical classifier. The recognizable relationships between the empirical information and that of the SVM can help to generate adequate movements.

1 Introduction

As reported in [1] it is a difficult task to maintain a save system behaviour in unknown environments. Interdisciplinary investigations in biology in soil physics in acoustics and experiments with different material clarify that anticipation of sound waves from surface can support a safe walking. "Audiogenic Stepping" reported in [2] is a benefit for generation of adequate motions. We have studied significant acoustic emissions for different material of present step and the preceding steps. It is impossible to classify all materials of ground and to react sufficiently. Therefore we model "Audiogenic Stepping" as a technical approach.

Section 2 gives an overview about the concept of sensors and the information flow. Some data representations clarify the relationship of the most important parameters. Section 3 offers our approach how to classify the incoming signals generated by the interaction of robot and surface ground. Experiments will be present and evaluate in Sect. 4. In Sect. 5 real-time considerations were presented. Finally we will conclude and looking forward in Sect. 6.

2 Sensor Actuator Concept, Environmental Observations

Currently we are working on a low level regarding sensor readings and pre processing. As a technical platform we use standard PC104, CAN bus and Microcontroller environment like C167.

Figure 1 point to the relations and information flow between robot components, forces, and environment with the emitted waves from ground during walking.

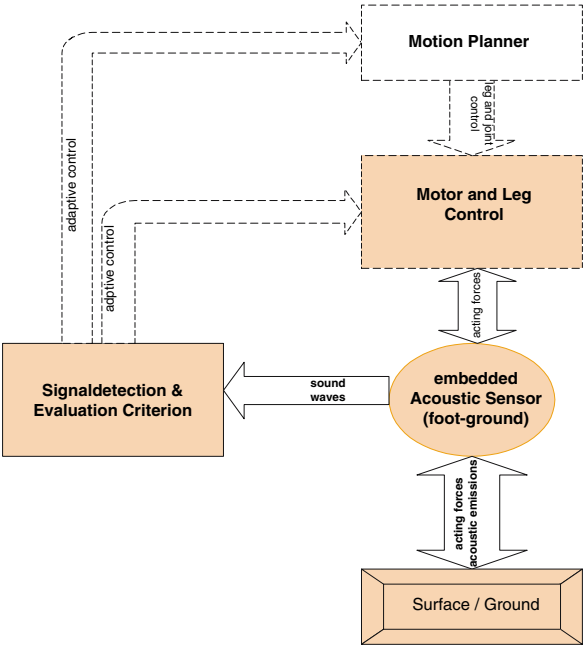


Fig. 1.

The Observation in Fig. 2 and Fig. 3 depict a sonogram of cracking plywood and the belonging force. During the cracking event at time 17–18 a short peak of broadband noises is emitted. This type of emission prefigures the beginning of cracking process and can help to predict the breakthrough and to react timely.

It is not possible to classify all the types of soil or ground by acoustic emissions only and to answer the question if the new step is a save one against the last one. Therefore we will inspect the acting forces, the soil dynamic and the corresponding sound emitted each point of time. Figure 3 prefigures for example the time to react before breakthrough.

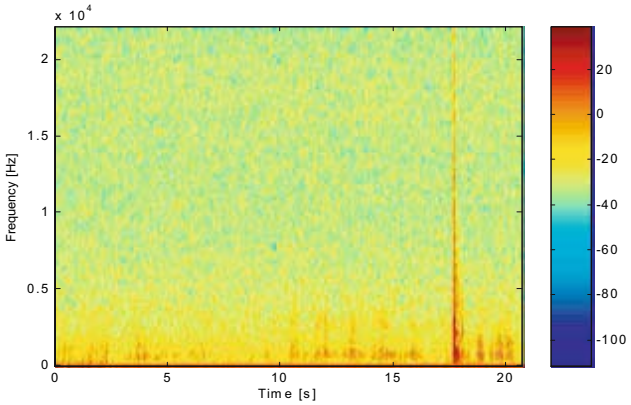


Fig. 2.

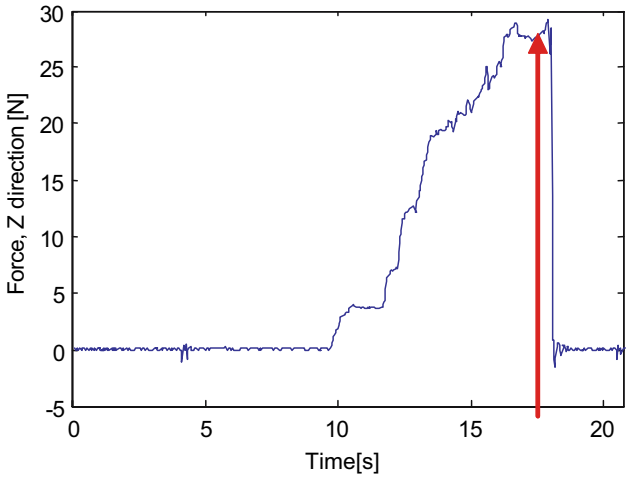


Fig. 3.

Sliding steps on gravely soil associated with a more powerful sound and varying directions of the leg forces to X-Y and Z direction of the leg (Figs. 4, 5) demonstrate also the importance of this approach.

The offline analysis of these data by a standard tool helps to understand the process of "Audiogenic Stepping". In our current work we want to percept *timely* (online sensing and evaluation) environmental changes and to act anytime. To achieve a good anticipation a specific module can evaluate the incoming signals and distinguish more and less dangerous situations using the different kind of noises and the forces on one leg of the robot. To get all the important information from the noise or sound we do a feature extraction as commonly used in speech recognition and by this get a feature vector for each

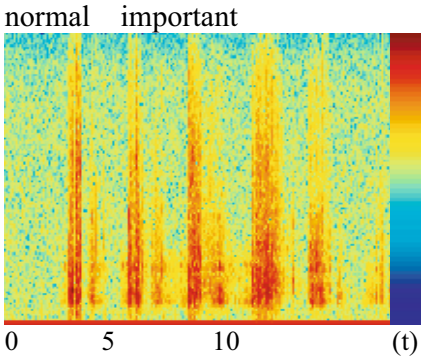


Fig. 4. Sonogram of gravely soil

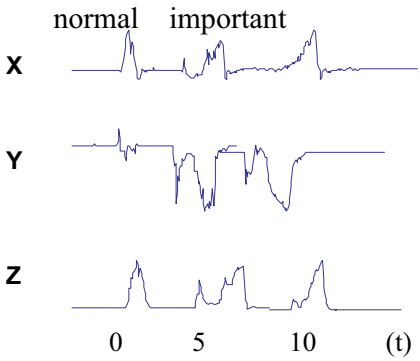


Fig. 5. Associated varying force directions

time frame. To these feature vectors from the noise we add dimensions which contain information of the force resulting in a feature vector $\mathbf{x}$. We want to do supervised learning and hence need for each feature vector $\mathbf{x}$ a discrete label describing the class c it belongs to. The classes may be defined as several dangerous and not dangerous situations.

3 The SVM Approach to Anticipate and to Classify

The application of SVM is motivated by applicability in processes like speech recognition were this algorithm works faster than comparable algorithms like the Percepton. The smaller amount of data play also an important role. In the further steps we use data generated by experiments online and processed offline. We use Support Vector Machines (SVM) [3, 4] with probabilistic outputs [5] to get the probability $p(c \mid \mathbf{x})$ that a feature vector $\mathbf{x}$ belongs to a class c.

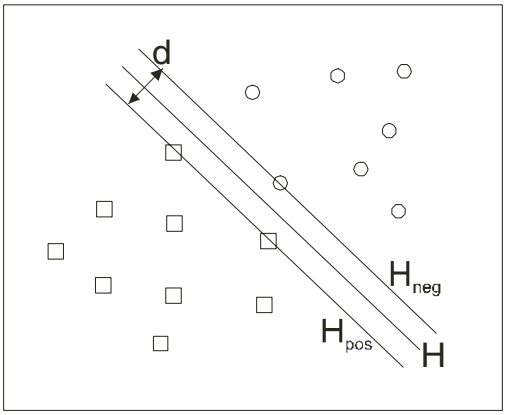


Fig. 6. The maximum margin separating hyperplane

Support Vector Machines (SVM) were first introduced by Vapnik and developed from the theory of *structural risk minimization* (SRM) [3]. We now give a short overview of SVMs and refer to [6] for more details and further references.

Given a training set $\{x_i, y_i\}$ of N training vectors $x_i \in \Re^n$ and corresponding class labels $y_i \in \{-1, +1\}$ there exist several hyperplanes $\mathbf{H}$ that separate the two classes. The vectors that lie on this hyperplanes satisfy the function

$$\mathbf{H} = (\mathbf{wx}) + b = 0 \tag{1}$$

where $\mathbf{w}$ is the normal to the hyperplanes and with $|b| \,/\, \|\mathbf{w}\|$ the distance from $\mathbf{H}$ to the origin normalized by the Euclidean norm of $\mathbf{w}$. We are now looking for a function $h : \Re^n \rightarrow \{-1, +1\}$ from a set of hyperplanes $\mathbf{H}$ which maximize the distance of the separating hyperplanes to the closest positive and negative training vectors. This distance is called the *margin* and all vectors satisfy

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \qquad \text{for } y_i = +1 \tag{2}$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \qquad \text{for } y_i = -1 \tag{3}$$

Every vector satisfying the equality conditions of (2) or (3) lies on a Hyperplane $\mathbf{H}_{\text{pos}}$ or $\mathbf{H}_{\text{neg}}$ respectively. These vectors are called the *support vectors*, see Fig. 1.

With the normal $\mathbf{w}$ and the perpendicular distance from the origin $|1 - b| \,/\, \|w\|$ for *support vectors* on $\mathbf{H}_{\text{pos}}$ and $|-b - 1| \,/\, \|w\|$ for *support vectors* on $\mathbf{H}_{\text{neg}}$. This results together with (1) in a *margin* $\mathbf{d}$ between $\mathbf{H}_{\text{pos}}$ and $\mathbf{H}_{\text{neg}}$ of size $2/\|w\|$.

So we need to maximize the distance $2/\|w\|$ which is equivalent to the minimization of $1/2\,\|w\|^2$. To solve this optimization problem the theory of Lagrange multipliers can be used which results in the maximization of the following dual formulation [6]:

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \tag{4}$$

under the constraints $\sum_{i=1}^{N} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C \; \forall \; i$. The parameter $C > 0$ allows us to specify how strictly we want the SVM to fit the training vectors [3].

We can expand the SVM to non-linear classification by transforming the training vectors into a higher dimensional *feature space* F using: $\Phi : \Re^n \rightarrow F$. Since the SVM algorithms use the training vectors only in the form of Euclidean dot-products $(\mathbf{x}_i \cdot \mathbf{x}_j)$ it is only necessary to calculate the dot-products in the *feature space*: $(\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_j))$. This is equal to the so-called *kernel-function*

$$k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_{xsi})\Phi(\mathbf{x}_j) \tag{5}$$

if $k(\mathbf{x}_i, \mathbf{x}_j)$ fulfils the Mercer's condition [4]. Using this *kernel-trick* we could compute the dot-product directly without knowing the explicit form Φ:

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i x_j) \tag{6}$$

Common *kernel-functions* are the polynomial kernel $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$ which maps into the space of all monomials up to degree d and the Gaussian radial basis function (RBF) kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2)$.

The output of the SVM for a test vector $\mathbf{x}$ is

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) \tag{7}$$

where f(x) > 0 means that $\mathbf{x}$ is classified to class $+1$. Only these vectors $\mathbf{x}_i$ for which $\alpha_i > 0$ are called *support vectors* so that the final set of *support vectors* should be very small.

3.1 Estimation of Probabilities from the SVM Output

As we have seen in the previous section the out of a SVM is a distance measure between a test pattern and the decision boundary. Because of the fact that we want an estimation of how dangerous the next step of the robot might be, we need a probability of the resulting class decision. But there is no clear relationship between the distance measure and the posterior class probability$^{p(y=+1|x)}$ that the pattern x belongs to the class$^{y=+1}$. A possible estimate for this probability can be obtained [5] by modelling the distributions$^{p(f|y=+1)}$ and$^{p(f|y=-1)}$ of the SVM output$^{f(x)}$ using Gaussians of equal variance and then compute the probability of the class given the output by using Bayes' rule. This yields to

$$p(y = +1|\mathbf{x}) = p(y = +1|f(\mathbf{x})) = g(f(\mathbf{x}), A, B) \equiv \frac{1}{1 + \exp[Af(\mathbf{x}) + B]} \quad (8)$$

The parameters of (8) are fit [5] using maximum likelihood estimation from a second training set (p_i, t_i) with $p_i = g(f(\mathbf{x}_i), A, B)$ and with the target probabilities $t_i = 1/2(y_i + 1)$.

The parameters A and B are found by minimizing the cross entropy error function

$$-\sum_i [t_i \log(p_i) + (1 - t_i) \log(1 - p_i)] \quad (9)$$

using the algorithm of Platt [5] which is based on the Levenberg-Marquardt algorithm.

4 Experiment and Evaluation of Acoustic Information

To achieve a good anticipation a specific module can evaluate the incoming signals and distinguish more and less dangerous situations using the different kind of emitted sound from surface and the forces of the leg of the robot acting on ground. To get all the important information from these sound we do a short-time *Fast Fourier Transformation* (SFFT) as commonly used in speech recognition. To these feature vectors from the sound we add dimensions which contain information of the force resulting in a feature vector $\mathbf{x}$. We want to do supervised learning and hence need for each feature vector $\mathbf{x}$ a discrete label describing the class $\mathbf{y}$ it belongs to. The class labels were done by hand and the different classes of sound were divided into five levels of danger.

4.1 Experimental Environment

We investigate relations between acting forces, emitted sound waves and subjective anticipation. We collect stepwise datasets of force and sound on wood and other materials with one leg of our robot mounted on a breadboard construction. The foot contains the microphone (Type Sennheiser KD97-12) to collect the signals and transforms the acoustic waves into electrical waves. These digitalized electrical waves by a sound card with a sample frequency of 44,1 kHz. This sample frequency will transfer the high quality recorded information of the microphone (33 Hz. 9 KHz [7]).

4.2 Data Evaluation

We observe 4 steps on plywood (Fig. 7 to 8). The step which is considered in Fig. 7 at fist ((btw: practically it is the second)) is very short and fast hit on the wooden surface. There are not that much changes in the vertical force $\mathbf{Z}$ but we considered it as a good step because of the high energy of the sound. The human interpretation says the higher the energy of the sound the better

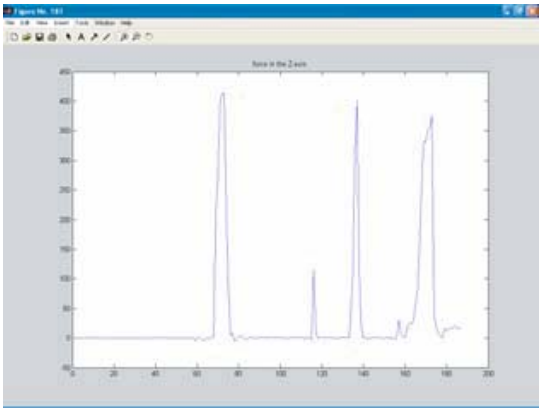


Fig. 7. Forces

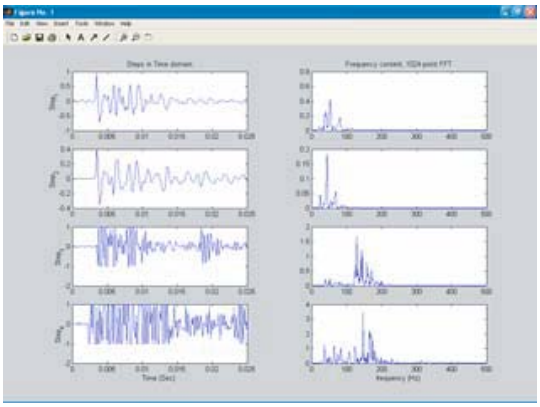


Fig. 8. Sound

and safer the step will be. If its sure that the step happened and finished, here comes the point which will force us to take the frequency in our consideration. Our evaluation should be also depend on the value of the frequency, clarified by different experiments when the break happens the frequency *and* the energy will be much higher than the normal steps. If we are just testing the normal steps, most likely the evaluation will be depending more on the energy of the sound and may be also depending on the force. For the material wood we find independent form the thickness in normal steps typical frequencies, whereby the band-with differs depending on the acting force. We have to compare the last step with the actual step regarding the energy (db) which is coupled on the investigated frequency area. So we are able to anticipate timely in the actual step a important deviation from these frequency band-with (crack).

4.3 Evaluation of Our Classification Method

In our classification experiments we could only use a small number of records and so the results are very preliminary. After training of the SVM and estimation of the parameter A and B of the probability-output of (7) we test the method by applying some feature vectors $\mathbf{x}$ to the SVM and calculating the posterior probability $p(\text{“danger”}|\mathbf{x})$, and comparing these probabilities with the previously known five levels of danger ($0 \ldots 4, 0$ meaning no danger) belonging to these feature vectors. The results are shown for one example of each level in Table 1.

Table 1. Human evaluation of measured parameters

Step	Energy	Freq. (Hz)	F_Z – Medain	Evaluation Number	Real Result
0	$8.^*10^{-6}$	37.2	1	0	Nothing, swinging leg
2	0.4241	52	116	1	Very short fast step
1	0.1855	43	413	2	Long normal step
3	1.6772	128	402	3	Cracking wood
4	3.4121	147	375	4	Break through

Table 2. Classification results for one example of different levels of danger of the recorded “wood”-data

Level of Danger of $\mathbf{x}$	Posterior Probability $p(\text{“Danger”}|\mathbf{x})$
0	0.08
1	0.16
2	0.40
3	0.83
4	0.94

These “soft” classification results show as expected that the posterior probabilities are approximately proportional to the level of danger (level 0 should correspond to $p(\text{“danger”}|\mathbf{x}) = 0$, and level 4 to $p(\text{“danger”}|\mathbf{x}) = 1$). Note that the probabilities are normalized as $p(\text{“danger”}|\mathbf{x}) + p$ (not $\text{“danger”}|\mathbf{x}) = 1$. As can be seen in Table 2 the vector which has the empirical level of danger (“4”) shows 94% of dangerous. The system should be warned!

5 Real-Time Aspects

In this paper we stress the role of recognition which is important for a correct response. Never the less we will elucidate the role of real time because of the efficiency of this response. The task of recognizing can be structured in subtasks like data recording and computing signals. Hereby the efficiency depends on the communication between shared components like sensor readings, bus transfer, CPU, and storing data. Therefore we will manage the computational power to handle the lot of tasks by TAFT (Time Aware Fault Tolerant) scheduling [8]. The ability to start, stop and continue any time, the tasks generated by sensors and actuators and the ability to schedule the sensor and actuator tasks depending on the importance of the active exploration tasks (Publisher Subscriber Protocol [9]). The SVM works faster than comparable algorithms like the Percepton. The smaller amount of data plays also an important role. In the process of technical recognition with SVM we find iterative processes [10]. At a first glance there exist no any-time capability. We will investigate these processes to improve them.

6 Conclusion and Future Works

It is hard to describe in an analytical way all dynamically relationships of the ground and the robot. Unexpected environmental changes complicate a safe walking. Practical investigation of steps on different ground delivers information which can be interpreted by an operator. The relationship between past robot body and the environmental reactions and the subjective interpretations can be learned by our SVM and support further anticipations for good steps or bad steps. In our classification experiments we could only use a small number of records and so the results are preliminary. Therefore we have to collect and to compute more data online in future works and to involve real-time aspects.

References

1. Deutscher M., Ihme Th (2003) How to achieve predictability in unknown environments – a senor based approach for a walking robot. 6th International Conference on Climbing and Walking Robots 2003. Professional Engineering Publishing Limited, Burys St Edmunds and London, UK
2. Stein P., Gillner S., Selverston A., Stuart D. (1999) Neurons, Networks, and Motor Behaviour. MIT Press
3. Vapnik V. (1995) The Nature of Statistical Learning Theory. Springer Verlag
4. Schölkopf B., Smola A.J. (2002) Learning with Kernels, MIT Press
5. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods.in A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, Advances in Large Margin Classifiers, MIT Press

6. Burges C. (1998) A tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, 2(2)
7. Dynamische Wandler: Sennheiser Electronic GmbH
8. Nett E. (1997) Real-Time Behaviour in a Heterogeneous Environment? In: Third International Workshop on Object-oriented Real-time Dependable Systems. Newport Beach, Ca
9. Rajkumar R., Gagliardi M., Sha L. (1995) The Real-Time Publisher/Subscriber Inter-Process Communication Model for Distributed Real-Time Systems: Design and Implementation. In: First IEEE Real-Time Technology and Applications Symposium
10. Christiani, Taylor (2000) An Introduction to Support Vector Machines. Cambr. Univers

Ultrasound Sensor System with Fuzzy Data Processing

J.A. Morgado de Gois and M. Hiller

University Duisburg-Essen, Campus Duisburg, Faculty of Engineering Sciences, Department of Mechanical Engineering, Institute of Mechatronics and System Dynamics, Lotharstr. 01 MD 220–239 47048 Duisburg, Germany
{morgado;hiller}@mechatronik.uni-duisburg.de

1 Introduction

The robot ALDURO [1] is a four-legged walking machine, supposed to operate in unstructured terrains. It has four degrees of freedom at each leg, what makes impossible to the onboard operator directly to control the whole robot movement; therefore this task was partially automated. Hereby the collision avoidance task follows as a natural need, which must be implemented accordingly to ALDURO's reality: large dimensions, slow and spatial movements, unstructured environment, and no need of a long term path planning (there is an onboard operator).

Considering such characteristics, it was decided to implement a reactive system, using a local map, based on the data from ultrasonic sensors. Such sensors are quite precise with respect to range measures, but suffer of intrinsically poor angular resolution [2], what conversely brings an advantage: they cover a whole volume at each measurement. Because of such angular inaccuracy, the inverse sensor model plays a relevant role to interpret each measure based on the sensor characteristics. The so formed information has to be added in an appropriated way to a base of knowledge (the local map), that is the so called data fusion process. Here is proposed a fusion by the use of a fuzzy inference system.

2 Inverse Sensor Model Data processing

Differently from other sensing medias to measure the distance to a target (like laser beam); ultrasonic sensors have quite large emission beams, which makes a priori impossible to know the exact direction.

Actually, conjugated use of many ultrasonic sensors or analysis of the shape of the returning echo allows a good approximation of the direction to

the detected object. However, such shape analysis requires high quality analogue sensors and a posterior, complex and time-consuming data processing. Besides, conjugated use of sensors requires the possibility to operate receivers and senders separately, what is not always possible, especially with cheap range finder units, where the whole detection is treated as a single operation.

For these reasons, a different approach is proposed here. We try to overcome the problem of not being able to directly measure the returning echo direction with a single sensor. It is important to note that, for the ALDURO application, the objective is not to locate an object at once, but to support the robot navigation. Therefore a map is constructed and many successive measures of each sensor are taken into account; but differently of the conjugated use of several sensors, where at least three different measures are used to directly localize a point, here the successive measures of the same sensor will be aggregated to the map to build knowledge about the surround.

A set of three coordinates is necessary to completely localize a point with respect to a frame fixed to the sensor. Using successive crisp measures and their intersection to find the real direction of the detected object would lead to a collection of (still imprecise) isolated points, which wouldn't really represent the surrounds. So, a natural choice to represent the point location is by means of fuzzy numbers.

These numbers are presumed to represent uncertainties of these coordinates, based on the many sensor characteristics (beam shape, gain, precision...) and the measured distance. In this way they describe the part of the sensor workspace containing the detected point. Several of these fuzzy sets together, through the appropriate inference, will lead to the more exact representation of the obstacles and thus the surroundings of the robot.

3 Target Coordinates

Considering a reference frame fixed to the sensor unit, with its X-axis along the sensor emission axis, a set of coordinates to locate the detected point can be stated as $\{\alpha, \beta, r\}$, where α represents the rotation around the Z-axis, followed by a rotation of angle β around the X-axis and r is the distance to the origin, as shown in Fig. 1.

A domain and a membership function over such domain define a fuzzy number [3]. This function represents how much an element of the domain satisfies a certain predicate. Through a fuzzification process, a measure d of the ultrasonic range finder is to be converted in a group of fuzzy numbers $\{A, B, R\}$, which are actually fuzzy sets and thus represent the detected point's coordinates and the uncertainty inherent to them. But a question arises: what would be the predicate to be attended in defining such sets? A good one could be just: "possible value", the set of the possible values for the target coordinates, given a sensor read value.

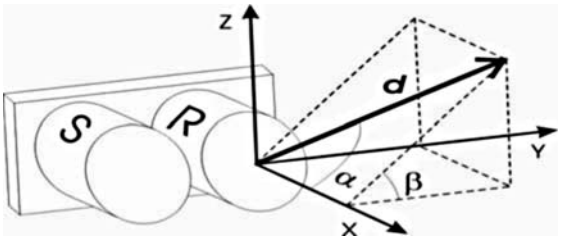


Fig. 1. Sensor Coordinates

4 Fuzzification

Suppose that the sensor imprecision about the distance measure is ε. It means that for a given measure d the actual distance lays in the interval $[d-\varepsilon.d; d+\varepsilon.d]$, respecting the ranging limits, i.e., the domain of this variable $[0; r_{\max}]$. Setting a membership function centered at the distance d indicates that d is the most "possible value", while other values out of the interval $[d-\varepsilon.d; d+\varepsilon.d]$ have membership grade null (it is not a possible value for the actual distance).

Let $I_{\min}$ be the minimal sound intensity measurable by the sensor, and I_d the intensity of a returning echo from an object at distance d, so can be defined the index $K = I_d/I_{\min}$. With $r_{\max}$ as the maximal detectable range (nominally at $\alpha = 0°$) and $G(\alpha)$ the gain of the receiver at an angle α, from the quadratic propagation law of the sound, we obtain (1).

$$K = \frac{r_{\max}^2}{d^2} \cdot \frac{G(\alpha)}{G(0)} \tag{1}$$

K is directly related to the possibility of a successful reading for an object at a given point, or conversely, the grade in which a value (the point location) can be taken as a "possible value".

In (1) is possible to see that the "possibility" decreases with square of the distance, but if the real maximal range of the sonar is large enough, it can be set up so that these decrease can be neglected.

The last term of (1) shows the dependence of K to the gain function and therefore to α. As the gain function (shown in Fig. 2) can be approximated with very good precision by a Gaussian (Normal) curve, a Gaussian membership function is adopted for α. For a measured d, the support is $D_\alpha = [-\alpha_{\max}; \alpha_{\max}]$, where $\alpha_{\max}$ is the maximal possible angle at the given distance.

Considering an error of $\pm 2\%$ in the range measure [4], membership functions can be stated to the coordinate r. Once again, the Gaussian membership function is adopted, now to the range. Using the product operation as T-norm, we obtain an inverse sensor model shown in Fig. 3, which is very similar to that proposed in [5]. The difference between these models lies on the fact that the model here proposed use a fuzzy logic approach, while the later uses a probabilistic one.

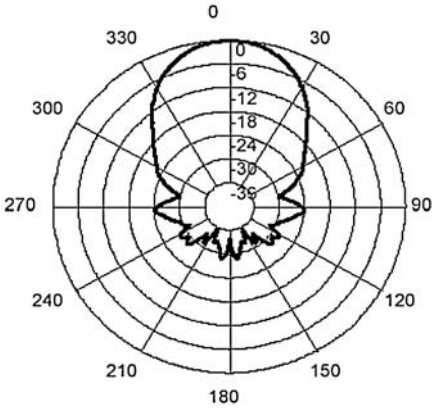


Fig. 2. Gain chart of sensor receiver

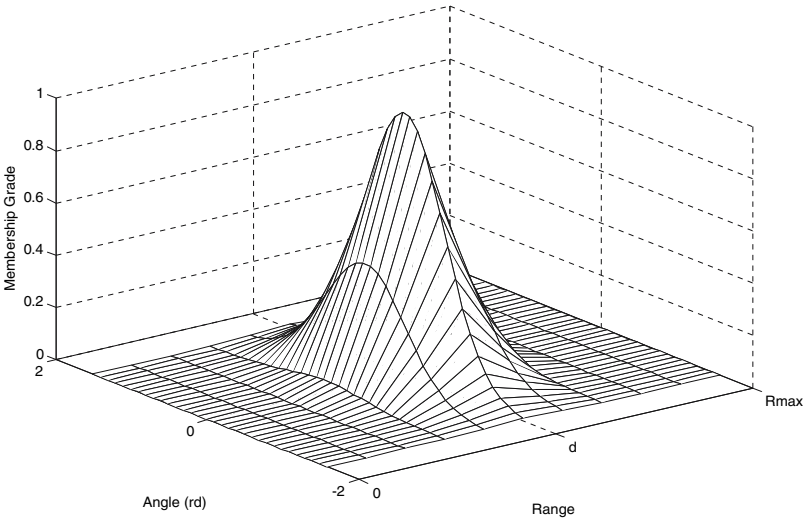


Fig. 3. Inverse sensor model

For the angle β the membership function is taken as unitary at the whole domain $D_\beta = [0; 2\pi]$ because of the simple interpretation of the predicate: any value there is as possible as any other.

In this way, based on a measured distance d, a point located by the coordinates $\{\alpha, \beta, r\}$, has its uncertainty expressed by the fuzzy numbers in (2).

$$\begin{aligned} A &= \{ (\alpha, \mu_A(\alpha)) / \alpha \in D_\alpha \} \\ B &= \{ (\beta, \mu_B(\beta)) / \beta \in D_\beta \} \\ R &= \{ (r, \mu_R(r)) / r \in D_r \} \end{aligned} \tag{2}$$

As a ranging operation is performed, it is possible through (2) to state the membership grade of the coordinates of the points contained in the workspace

of the sensor. But it is necessary to have a single membership grade for each point $s = (\alpha, \beta, r)$ and not one for each coordinate. So in (3) the Cartesian product is used to find the membership function of the point s, stating a new fuzzy set S,

$$S = A \times B \times R \Leftrightarrow \mu_P = \mu_A \cap \mu_B \cap \mu_R = \mu_A \cdot \mu_B \cdot \mu_R \tag{3}$$

The classical product operation is used as *T-norm* (the corresponding operator for intersection), what makes the calculation somewhat easier. Hence, as $\mu_B = 1$, the overall membership function remains depending on α and r, as in Fig. 3.

5 Data Fusion

To be used, the information from each individual sensor (or from the same sensor at different moments) has to be aggregated to the knowledge base, to the map. In order to do that is necessary first to transform the obtained measure and its membership functions to the Cartesian reference system of the local map.

Through the use of the Principle of Extension [3] and coordinates transformation it would be possible to obtain S in the local map's reference system. That would be very suitable to the construction of an occupancy grid as in [4], but here a simplified solution is proposed, where the whole spatial information is not necessary. The map will be generated by a Takagi-Sugeno-Kang (TSK) Fuzzy inference system [6, 7].

The XY-plane, on which the local map is based, is partitioned in many cells and linear functions are associated to each one. Here, instead of the usual crisp inputs used in TSK systems, a fuzzy input will be used. This input is the projection of the fuzzy set S onto the XY-plane, as shown in (4)

$$\mu_{S_{xy}}(x, y) = \sup_{Z} (\mu_S(x, y, z)) \tag{4}$$

Such function is quite complicated to obtain analytically, but considering that for each possible value of d, the support of R is much smaller than the corresponding length of S, we neglect the influence of R. Then, the support of S is reduced to a surface and is possible to obtain analytically its projection on the XY-plane.

6 TSK System

With fuzzy inputs rather than crisp ones, the activation values are found from the intersection of each partition with S_{xy}. One problem is to state suitable

membership functions for the partitions, because calculation of surfaces" intersection in general needs a lot of processing. We try to overcome that using singletons as membership functions.

In such kind of system, the activation values are used to make a weighted average of the output functions (trough the so called fuzzy rules). Here it was employed one rule for each partition of the map-grid.

7 Results and Conclusions

Two kinds of linear functions were tested as output functions: constants and planes. The first one has one parameter per rule while the second, three parameters. That makes the simulations much slower when using planes, because the dimensions of the matrices employed at the calculation are directly proportional to the number of parameters per rule.

To the adjustment of the parameters was used of Recursive Least Squares Method. Simulations were run with different number of partitions and different

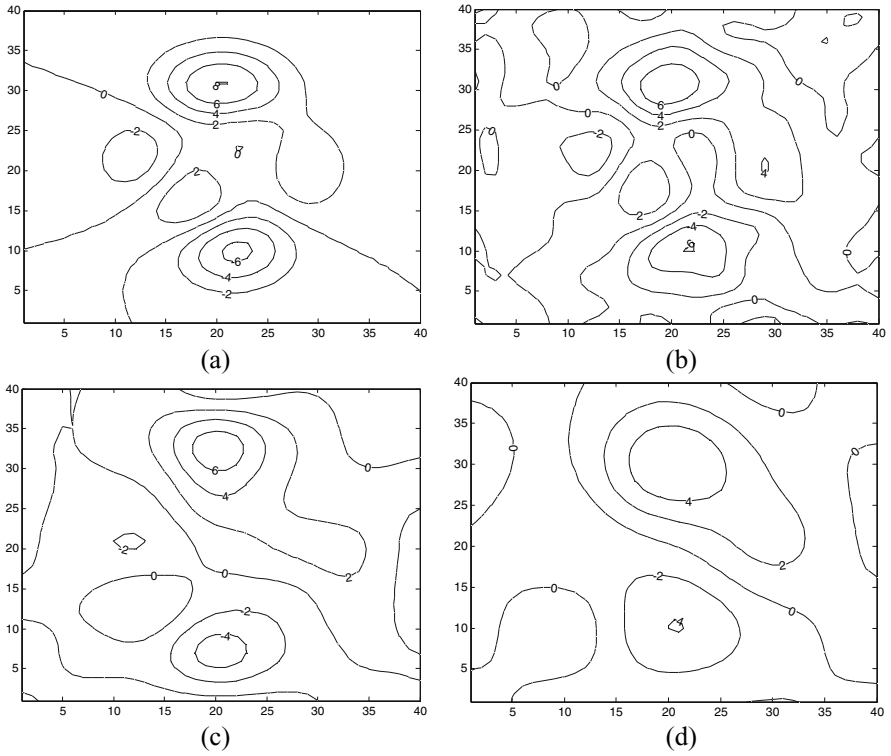


Fig. 4. (**a**) Input Terrain; Maps obtained by (**b**) TSK with planes as output functions; (**c**) TSK with constants as outputs and (**d**) Occupancy Grid

input terrain; as expected the system seems to be a little sensitive to different combinations of terrain and number of partitions, but the general performance holds.

In Fig. 4 are shown the input terrain (a), the results obtained through occupancy grids [4] (b), through TSK system with constants as output (c) and through TSK system with planes as output functions. All the simulations in Fig. 4 use the same number of sample points and to trace the contour plots, the same cotes.

The results obtained by TSK system are in general closer to the real terrain than the ones obtained by occupancy grid; much better when the number of partitions is appropriated to the topography of the terrain.

Planes as output functions have a smoother result than constants, having a superior performance when the number of samples is small. However the processing time is bigger, what is not a problem if the processor is fast enough to keep that smaller than the ranging time of the sensor.

References

1. Müller J. (2002) Entwicklung einer virtueller Prototypen des hydraulisch angetriebenen Schreitfahrwerks ALDURO. Fortschr.-Ber. VDI Reihe 1 Nr.356. Düsseldorf: VDI Verlag 2002
2. Risse W. (2002) Sensorgestützte Bewegungssteuerung kinematisch redundanter Roboter. Bericht aus der Robotik, Aachen: Shaker Verlag, 2002
3. Jang J., Sun C., Mizutani E. (1997) Neuro Fuzzy and Soft Computing. Prentice Hall, Upper Saddle River-USA
4. Morgado de Gois J.A., Germann D., Hiller M. (2003) Sensor-based ground detection in unstructured terrain for the walking-machine ALDURO. 6th International Conference on Climbing and Walking Robots CLAWAR, pp. 911–918, 2003, Catania, Italy
5. Moravec H.P., Elfes A. (1985) High Resolution Maps from Wide Angle Sonar. Proceedings of the 1985's IEEE International Conference on Robotics and Automation
6. Sugeno M., Kang G.T. (1988) Structure identification of Fuzzy model. Fuzzy sets and Systems, 28:15–33
7. Takagi T., Sugeno M. (1985) Fuzy identification of systems and applications to modelling and control. IEEE Transactions on Systems, Man and Cybernetics. 15:116–132

Finding Odours Across Large Search Spaces: A Particle Swarm-Based Approach

Lino Marques and A.T. de Almeida

Institute for Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
{lino,adealmeida}@isr.uc.pt

Abstract. This paper proposes an evolutionary-based search algorithm to find odour sources with robot communities across large search spaces. The characteristics of outdoor odour plumes and the main problems in detecting and finding them in real environments are described. An artificial olfaction system designed to carry out olfaction-based mobile robot experiments in realistic conditions is shown. This olfaction system is composed by intelligent gas sensing nostrils and a directional thermal anemometer. The searching algorithm proposed is inspired in the particle swarm optimization (PSO) method. This algorithm allows coordinating the movements of multiple robots searching for odour sources across large search spaces. The paper describes the algorithm and compares its performance against other searching strategies.

1 Introduction

The autonomous on-line search of hidden items across very large search spaces can be a very slow process. If the searched items release volatile chemicals, the search process could be highly accelerated by sensing and following the downwind chemical plume. Living organisms frequently use olfactory strategies to search and find mates, to find food and to follow chemical trails between a source of food and their nest. This type of sensing mechanism could be of high relevance to improve the autonomy of mobile robots, but their implementation still poses several technological problems. The first problem to be solved is related to sense a chemical plume far away from the source. After leaving its source, a chemical plume becomes shred in small filaments by the effect of turbulence. To detect such small chemical patches, a fast and sensitive chemical sensor should be used. The second problem is related with the control of the robot in order to follow a meandering and intermittent plume till the source using the sparse chemical cues sensed across the search space. When multiple robots are used, an additional problem of efficiently coordinating them while they search the odour sources arises.

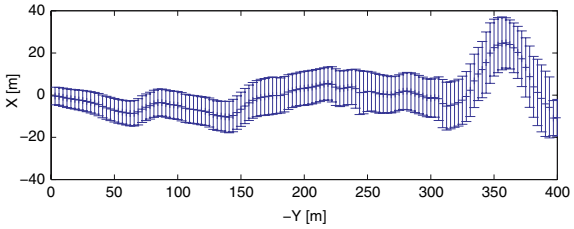


Fig. 1. Instantaneous centerline and width of a meandering chemical plume

1.1 Odour Plumes

Several algorithms have already been proposed to locate the source of odour plumes [6, 11], but those algorithms were tried under very simplified environmental conditions, namely: indoors with the robots travelling small distances across a constant airflow. The lack of smooth concentration gradients found in real outdoor atmospheric environments means that a simple gradient following strategy cannot be used to find an odour source. Far enough downwind of the source, odour plumes are filamentous and sporadic. The instantaneous concentration measured by a fast chemical sensor placed downwind the source will fluctuate with large intermittency periods and short concentration peaks that can be well above three orders of magnitude the average concentration value [8]. Under these circumstances, gradient-based navigation with an array of scattered gas sensors will be impracticable because the time-averaged local gradient is too small and the instantaneous gradient is time varying, steep and random-like. In this case, more complex strategies are required and additional information, such as airflow velocity and direction are essential [2].

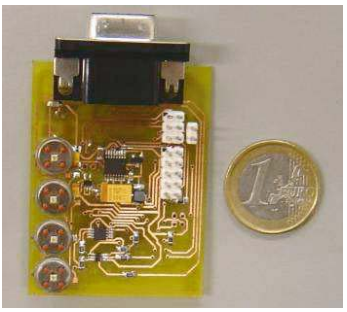
The experimentation of odour searching algorithms in natural environments is time consuming, requires outdoors mobile robots and appropriate olfactory sensing systems. A more convenient approach to develop and evaluate new algorithms can be to simulate the experiments, but in this case it is necessary to be able to generate realistic odour plumes. Figure 1 shows an instantaneous meandering plume generated with CofinBox[a] software package. This program allows to specify different environmental conditions concerning terrain type, gas released, wind characteristics, time and downwind distance and outputs to a text file the plume centre $y_0(x,t)$, width $w(x,t)$ and height $h(x)$ as a function of time t and downwind distance x.

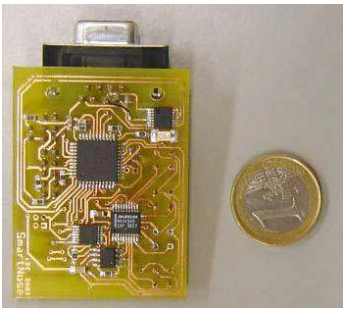
2 Olfactory Sensing System

From the above description, it becomes clear that an olfactory sensing system able to sense and track odour plumes until their source should have the following characteristics:

[a]This package was developed in the framework of European Union project Cofin [9].

(a)





(b)

Fig. 2. Wheeled mobile robot with an olfactory sensing system (**a**). Front and back pictures of a gas sensing nostril (**b**)

- It should be fast in order to detect small odour patches far away from the source.
- It should be selective in order to discriminate the odour searched from other similar odours that might be present in the environment.
- It should sense airflow intensity and direction in order to perceive where the odour comes from and what atmospheric conditions exist.

The olfactory system integrated in the robot of Fig. 2(a) is composed by a directional anemometer and smart sensing nostrils (see Fig. 2(b)) that are able to fulfil all of those condition. The thermal anemometer measures airflow intensity and direction using four self-heated thermistors placed around a square wind deflecting pillar. The power dissipated by each sensor depends from the difference between the device temperature and local air temperature and from the airflow around the device. Each gas sensing nostril possess an array of four different metal oxide gas sensors operated in temperature-modulated mode by a microcontroller-based signal conditioning circuit allowing to implement a large sensing space electronic nose [5]. The selectivity and fast identifica-

tion of the air mixtures is assured by the type of signal processing employed to the gas sensing array output, namely discrete wavelet transform (DWT) for transient analysis and principal component analysis (PCA) and artificial neural networks (ANN) for pattern recognition [1].

3 Searching Algorithm

Most population based search and optimization algorithms found in the literature start the population randomly across the search space. Whilst this type of initialization provides a good sampling of the space from the beginning, it is not very realistic to implement in robotic search agents, since usually those agents start searching from the same area, usually a corner of the search space. Common reactive search algorithms, like gradient climbing, cannot be used to search odours across large spaces because in most of the space no concentration can be sensed and so the agents become stopped without exploring unknown areas. Both the exploration of new areas and the exploitation of chemical cues eventually found are solved by means of a state-based search strategy [7]. When agents have no valid cue to exploit by means of local searching strategies, they switch to a global searching mode where they explore the environment by means of a random walk biased by the least known neighbour areas (see Fig. 3). In order to keep track of the lack of knowledge inside the search space, a rough grid map is used[a]. Each cell of this map contains the time spent exploring their corresponding area. Each time an agent finds a chemical cue, it exploits the cue by means of a local searching algorithm (e.g., PSO-based; Gradient Ascent; and chemical concentration Biased Random Walk).

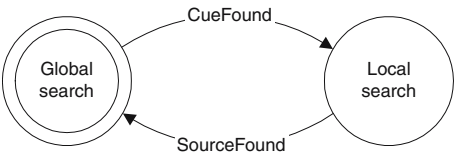


Fig. 3. State diagram representing the switching between global and local-searching modes

3.1 The Particle Swarm Optimization Algorithm

The Particle Swarm Optimization algorithm is an evolutionary technique originally proposed by Kennedy and Eberhart [3]. This optimization method takes

[a]In our tests, a 5×5 grid map was used for a searching area of $200 \times 200\,\mathrm{m}^2$

Listing 1. PSO-based searching algorithm

```
Initialize the population
Do
   Evaluate agent's fitness
   Compare each agent's fitness
   Calculate target positions using PSO
   Move till the target
Until finishing-criterium
```

inspiration from the dynamics of social organisms while searching for food, where social sharing of information takes place and individuals can profit from the discoveries and previous experience of all other companions (e.g., flocks of birds and herds of fishes). The original PSO algorithm assumed the existence of a group (or *swarm*) of searching elements (called *particles*) that move across a D-dimensional search space according to the following rules:

$$v_i(t) = \phi v_i(t-1) + \rho_1(x_{pbest\,i} - x_i(t)) + \rho_2(x_{gbest} - x_i(t)) \;, \tag{1}$$

$$x_i(t) = x_i(t-1) + v_i(t) \;, \tag{2}$$

where v_i and x_i represent the ith particle's velocity and position vectors respectively; $x_{pbest\,i}$ and x_{gbest} represent the particle previous best value and a predefined neighborhood best value respectively; ϕ is a constriction factor that allows to control the magnitude of the velocity; and ρ_1 and ρ_2 are two positive random values. The first equation updates the particle's velocity taking into account their previous velocity and the distances to the particle's best previous position and to the swarm's best experience. The second equation updates the particle's position by adding the calculated velocity to the particle's previous position. The ϕ value controls the explorative behaviour of the algorithm (higher ϕ values impose a behaviour more explorative). The ratio between ρ_1 and ρ_2 controls how collective and self experience influence future searching directions. More recent approaches of this technique change dynamically the magnitude of these coefficients, allowing to adapt the searching behaviour of the algorithm to the characteristics of the problem in hand [10].

3.2 PSO-based Robotic Searching

To search odour sources with robotic agents using a PSO-based algorithm, the agents should be able to localize themselves, measure local odour concentrations and keep the position of higher concentration and share this position with other neighbour agents. The PSO-based searching algorithm represented in listings (1) can use each agent's time-averaged concentration value as the fitness function in order to generate new target search positions. An odour source can be found by circumnavigating a suspicious area with closing spirals [4] or using a second detection mechanism to identify the source when

an agent passes sufficiently near (e.g., a vision system for a visible source). In this paper this last case is considered supposing the utilization of sensors with 1 m *radius-of-detection*. Each source found becomes collected and its respective odour field disappear. The searching process can stop after a pre-specified time or after a pre-specified number of sources have been found. The motion of the agents is commanded by means of a potential field-based method with target points representing attractive potentials and obstacles and other agents repulsive ones (see listings 2). Each time a target is reached or an agent velocity becomes too small, a new target position will be generated by the searching algorithm.

Listing 2. Agent's motion till a target postion

```
While Not_In_Target Do
  Calculate Attraction force to the target
  Calculate Repulsion force from other agents
  Calculate Repulsion force from static obstacles
  Update velocity
  Update position
End
```

4 Simulation Experiments

The proposed algorithm was simulated in Matlab with robotic agents moving in a large *natural* environment (200×200 m^2) composed by flat terrain with multiple scattered obstacles and multiple ground level odour sources. The robot's velocity and acceleration are bounded by 1 m/s and 0.1 m/s^2 respectively. Two very different environmental conditions were considered:

- Idealized conditions without airflow, where the dispersion of odour molecules through the environment was modelled by diffusion processes and the only existing noise was due to the sensing mechanism. This situation provides smooth test fields that allow evaluating the search algorithm's performance with common Gaussian shaped functions.
- Open field where the transport of odour molecules is mainly governed by turbulent airflow as was described in Sect. 1.1 and [7].

Figures 4(a) and 4(b) show example trajectories of BRW and PSO-based searching experiments described by 10 agents searching for 5 odour sources. The sources are diffusion fields with 5% Gaussian noise.

The performance of the proposed PSO-based search algorithm is compared against other commonly used algorithms, namely gradient ascent and biased random walk, employed in the same conditions (i.e., the algorithms were used

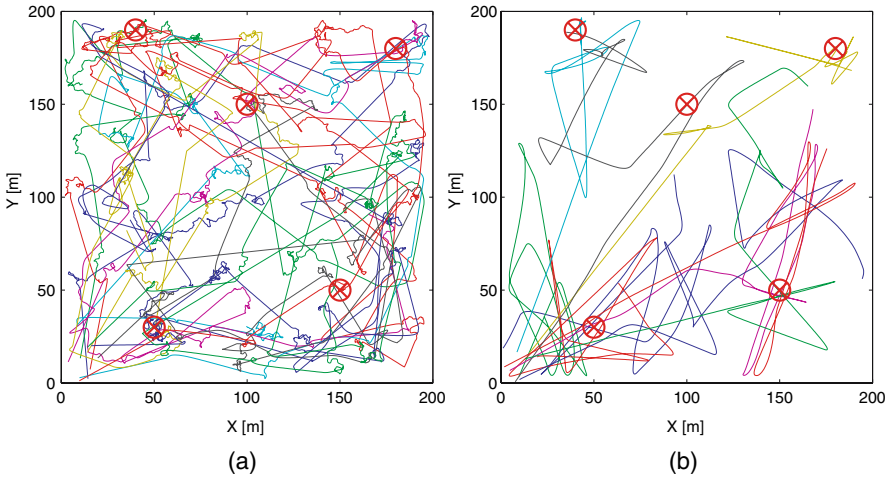


Fig. 4. Example with the trajectories described by a swarm of 10 robots searching for the source of meandering odour plumes. BRW (**a**) and PSO (**b**) local searching strategies

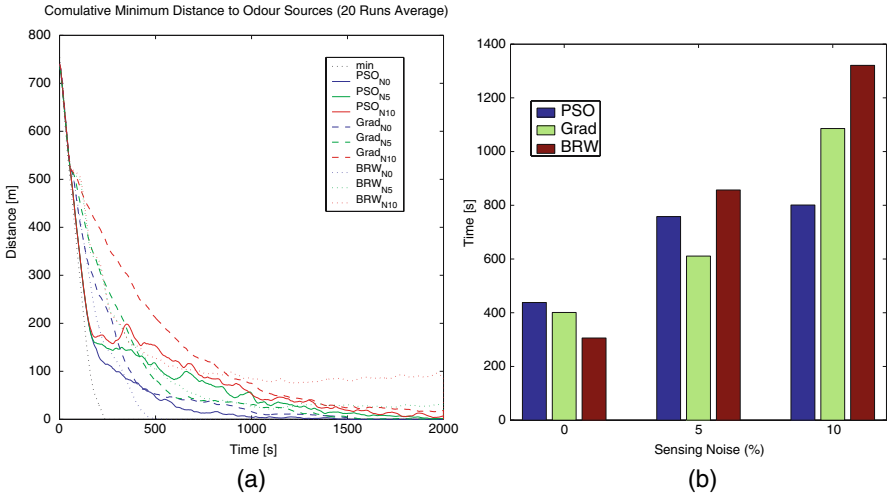


Fig. 5. Statistical analysis of the average time spent to find three odour sources in the simulation environment

as local searching algorithms of the state based search described in Sect. 3). Figures 5(a) and 5(b) show the statistical average obtained running 10 robots with different local searching strategies: PSO-based, gradient climbing and chemical concentration biased random walk[a] in noisy diffusion fields (0%, 5% and 10%). Each algorithm was run 20 times at three noise levels in a

[a]See [6, 7] for a description of BRW and gradient climbing algorithms

searching area containing five odour sources spread in fixed positions across the environment. Figure 5(a) shows the average cumulative minimum distance to the odour sources to be found (global proximity of the searching agents to the odour sources). This Figure demonstrates comparatively good running results for the PSO-based searching algorithm. Figure 5(b) shows the average time taken to find the first three odour sources. From this Figure it is possible to see that while PSO-based search was the worst algorithm for the no-noise situation, it becomes better when the noise level increases and it is clearly the best search algorithm for the highest noise level considered (10%).

References

1. N. Almeida, L. Marques, and A. T. de Almeida. Fast identification of gas mixtures through the processing of transient responses of an electronic nose. In *Proc. of EuroSensors*, 2003.
2. E. Balkovsky and B. I. Shraiman. Olfactory search at high Reynolds number. *Proc. Natl. Acad. Sci. USA*, 99(20):12589–12593, 2002.
3. J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *IEEE Int. Conf. on Neural Networks*, pp. 1942–1948, 1995.
4. L. Marques, N. Almeida, and A. T. de Almeida. Mobile robot olfactory sensory system. In *Proc. of EuroSensors*, 2003.
5. L. Marques, N. Almeida, and A. T. de Almeida. Olfactory sensory system for odour-plume tracking and localization. In *IEEE Int. Conf. on Sensors*, 2003.
6. L. Marques, U. Nunes, and A. T. de Almeida. Olfaction-based mobile robot navigation. *Thin Solid Films*, 418(1):51–58, 2002.
7. L. Marques, U. Nunes, and A. T. de Almeida. Odour searching with autonomous mobile robots: An evolutionary-based approach. In *Proc. IEEE Int. Conf. on Advanced Robotics*, pp. 494–500, 2003.
8. K. R. Mylne and P. J. Mason. Concentration fluctuation measurements in a dispersing plume at a range of up to 1000 m. *Quart. J. Royal Meteorological Soc.*, 117:177–206, 1991.
9. M. Nielsen, P. C. Chatwin, H. E. Jørgensen, N. Mole, R. J. Munro, and S. Ott. Concentration fluctuations in gas releases by industrial accidents – final report. Technical Report R-1329(EN), Risø Nat. Lab., Denmark, 2002.
10. K. E. Parsopoulos and M. N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1(2–3):235–306, 2002.
11. R. A. Russell, A. Bab-Hadiashar, R.L. Shepherd, and G.G. Wallace. A comparison of reactive robot chemotaxis algorithms. *Robotics and Autonomous Systems*, 45:83–97, 2003.

Vision Feedback in Control of a Group of Mobile Robots

Piotr Dutkiewicz and Marcin Kiełczewski

Institute of Control and Systems Engineering, Poznań University of Technology
ul. Piotrowo 3A, 60-965 Poznań, Poland
piotr.dutkiewicz@put.poznan.pl
marcin.kielczewski@put.poznan.pl

Abstract. This article is devoted to the vision system, which is an integral part of control system for mobile robots soccer team. The paper describes an experimental test bed used for real-time image acquisition, processing, and recognition of objects placed on the 2D surface. Description of specific features of objects being recognized, which are controlled and opponent team robots as well as a ball has been also included. Filtering methods used to improve quality of the image received from camera have been presented. The paper also shows a new heuristic algorithm, invented for objects recognition on the scene, playground in our case. The algorithm has been implemented, experimentally verified and quality of measurement has been estimated.

1 Introduction

Research results often lead to situation, where only narrow group of specialists is able to take advantage of the development outcomes. In recent years a concept arose to verify experimentally problems existing in complex systems, and to make them understandable for a group of people as broad as possible. Based on this idea, soccer competition between mobile robot teams were carried out [14]. Such training ground gives great opportunity for presentation complex problems in simple way, such as recognition of unknown environment, action planning and control of a mobile robot system [5, 6].

One of the important components of this is a vision system [3, 11], used for object recognition and measurement of position and orientation of objects placed on the playground. There are the following objects on the playground: three mobile robots of one team, three robots of opponent team and the ball. Measurement data, coming from vision system, are used as input to strategy and control algorithms, which produce set of commands sent to robots of the controlled team. Development of the visual feedback is the main goal of research presented in this paper, which is an extension of achievements

presented in [7]. Features of objects, vital for correct measurement, are examined. Also image filtering methods were selected as necessary for improving the quality of images received from camera. The main task was to develop recognition algorithm for specific objects on the scene. Implementation of extraction algorithm of specific features should satisfy criterion of speed, high reliability of operation and easy characterization of features for objects being recognized. Invented heuristic algorithm for objects recognition is based on the compromise between using widely known methods of filtering and image processing [2, 9, 10], and necessity of real time processing [1, 4].

2 Description of the Measurement Setup

A detailed description of the MiRoSoT (Micro-Robot Soccer Tournament) environment can be found in [11, 14]. According to the MiRoSoT regulations, CCD color camera has to be fixed above the playground. The video stream, coming from the camera, is then collected by a specialized PC card which enables image displaying on the user console and image data processing simultaneously.

The main part of the vision system is the framegrabber card Matrox Corona, achieving 25 frames per second (fps) for 768×576 image size with full soft real-time display and 24 bit color. As an input signal composite video, S-Video or RGB components signals, transmitted in PAL or NTSC standards, may be used with the framegrabber card. Additionally, the card has 24-bit digital input RS-422/LVDS. The library package MIL-Lite, supplied with Matrox Corona card, gives flexible and easy-to-use data processing and displaying framework available under Windows NT/2000 OS. Detailed description of the MIL library with application examples and Matrox Corona card can be found in [12, 13].

Very important part of the vision system, shown in Fig. 1, is a CCD camera. In our application Sony DCR-TRV 900E color camera was used. The camera is equipped with three CCD matrices, each for individual RGB components. Analog output formats available in the camera are composite and S-Video signals. Additionally, digital *i.Link* output, transferring compressed image in DV standard, is also accessible. During vision system development Microsoft Visual C++ 6.0 programming tool was used.

3 Features of Recognized Objects

In accordance with the MiRoSoT rules [14] two teams of robots participate in the game play. They play an orange golf ball, which is the easiest object to find on the playground. On image it looks like a color circular patch.

During the game a 8×8 cm marker with two color patches is placed on each player. One patch indicates team color (common for one team) and the

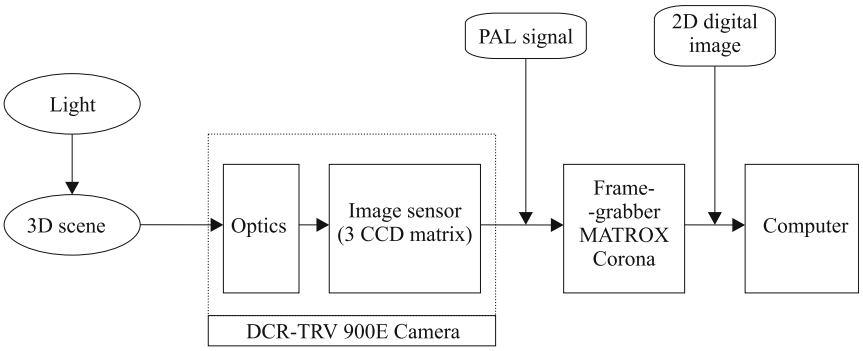


Fig. 1. General diagram of vision system

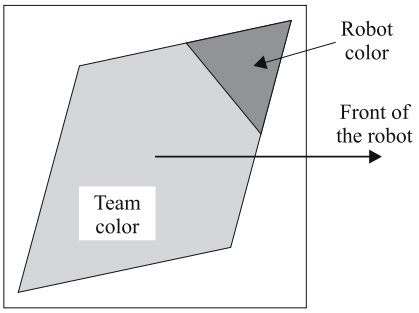


Fig. 2. Patch marker of our robot

other patch recognizes individual robot color. Team patch is a uniform spot square with minimum dimensions 3.5×3.5 cm. Yellow and blue colors are allowed. Individual robot patch identifies each robot inside team. The goal of the vision system is patch recognition and classification using color and associate individual and team colors of patches in pairs. After this, orientation and position of robots on playground are calculated using determined image coordinate of recognized objects. The marker system of our robot is presented in Fig. 2. Recognition of the robot (marked by color patches) depends on correct patch classification (according to defined colors) and finding two sharp corners of rhombus patch composed of individual and team patches. Based on determined points, position and orientation of robot are calculated.

4 Algorithm of Determining Position and Orientation of the Robot

In order to accelerate image processing algorithms, a technique based on a decreased image window [8] is often used. Unfortunately, it gives not always expected results, because in extreme case (after losing object) it is necessary

to increase processed window to the whole image. Moreover, object features extraction in color is more complicated comparing to black and white images.

It is assumed in this concept of objects recognition, that image is checked with quite large step in x and y directions in order to find a point inside robot patch marker or ball blob. Search step has to be selected to hit all regions of interest on image represented by defined colors and with the smallest amount of pixels to be checked. In this way significant acceleration of image processing is obtained. But with large search step it may occur, that a group of pixels representing a specific object will not be found and object cannot be recognized. To increase probability of hitting all regions representing searched objects, double image searching with the same big step has been introduced. Checked pixels between first and second searching are shifted by half fixed step. Second image checking is activated only if the first pass fails.

4.1 Color Segmentation

During image searching, detection of pixels representing objects is based on colors. Due to a noise on image and to make easier specification and allocation colors to objects, segmentation of available color space is used. This solution facilitates color classification of any object from available color palette and increases noise robustness. In this method, RGB color representation model [2] is used. Color segmentation form 24-bit color palette (R:G:B components have respectively 8:8:8 bits) is done by an indirect addressing algorithm. As a result of that, a three-dimensional version of a look-up-table technique is obtained. Data necessary to realize this are located in three arrays, filled with suitable values during initialization. First, one-dimension array is used for scaling color components. In this array, values of a function transforming discrete RGB components from 0-255 range to discrete values from 0-8 range (distribution of these values can be any function; also nonlinear). Indexing this table by the color component value gives new component value from narrowed range. This operation gives three new values of color components – they are three indices addressing elements in the second color array, which is three-dimensional (one dimension represents one color component). Elements in the second array are indices of colors, which are defined in the third table. Each element of this table is a structure consisting of few fields. These fields (connected with color and objects properties) contain information about color component value, color brightness, object identifier, and logical variable. The identifier indicates number of the object, to which has been assigned this color element, and logical variable defines, if this color has been assigned to any other object. Colors defined in this array are arranged from black to white, according to increasing color component value. In this way, using described arrays, it is easy to check to which object belongs examined point, described by RGB color components. As object a color set from a defined color palette (see Fig. 3(b) is understood. It is assumed, that there are six objects on the image: ball, patches with color of one team, opponent team and three individual

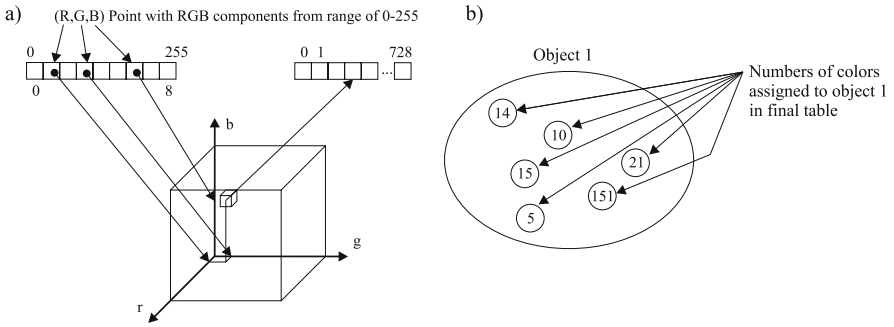


Fig. 3. Method of color space segmentation (**a**) and object defining (**b**)

color patches of players. Described method of color space segmentation is shown in Fig. 3(a).

4.2 Image Preprocessing Methods

During image searching for pixel of color that is assigned to objects, there are used two image preprocessing methods. First, contextless method (also called single-point filtering), is characterized by operations done only on single image pixels, without considering pixel neighborhood. Second preprocessing method is a logical context filtering, which is used to eliminate noise in the form of isolated pixels with object colors, which have not connection with neighbouring pixels. To check connection between examined pixel and neighbourhood pixels, convolution mask of 3×3 pixels size and center point of mask put on checked pixel is used. Logical filtering is based on checking, if all selected pixels under mask have the same color as the checked point. If so, that pixel is accepted as a point belonging to the object, which has assigned this point color.

In order to obtain independence from color interference, instead of checking team color for edges detection, information about pixels luminance was used. Luminance is transmitted practically lossless in PAL standard, contrary to color information. Pixels luminance of patch on image is similar, while color can change in wide range. Edge detection using luminance is possible, because light color patches are placed on dark background. If luminance of a particular pixel was above the threshold value and for the following checked pixel it was below the threshold, then current pixel was accepted as a border point. Because pixels in the image buffer are described by three RGB components, for checking luminance of each pixel it was necessary to recalculate from RGB space to Y luminance component.

4.3 Recognition of Own Team Robots

Simple comparison of pixel luminance with threshold does not take into consideration direction of edges on the image, so it is necessary to check how the rhombus is oriented before edge following starts. In this way it is judged, if the edge detection should be done along rows or columns of the image. Selection of processing direction is based on check, if longer diagonal of rhombus is oriented more along horizontal or vertical image lines. As it turned out in practice, it is good to distinguish situation, when angle between longer diagonal of the rhombus and image lines or columns is about 45°.

The described method of rhombus edge following and finding of sharp corners is based on the edge detecting along determined directions. The algorithm distinguishes four directions from which one is chosen, depending on approximate orientation of rhombus on the image. In Fig. 5 all four possible directions of cuts of rhombus are shown, which are used in edge detection, starting from approximate center of a rhombus. This method reveals some similarity to edge detection with use of the Sobel operator. The diagram of final determining one of sharp corners is presented in Fig. 4. An edge detection is based on test of image function gradient with use of first derivative. The pixels for which absolute value of gradient is big enough are chosen as edge elements. Sobel operator is designed to detect edges independently of their direction on the image.

When sharp corners of rhombus are found, on the basis of their coordinates accurate position and orientation of the robot are calculated. Position of robot is determined as arithmetic average of suitable coordinates of corner points. Orientation is calculated from the equation:

$$\Theta = Atan2(y_2 - y_1, x_2 - x_1) \,, \tag{1}$$

where pairs $(x_1, y_1), (x_2, y_2)$ mean coordinates of corner points of the rhombus.

After position and orientation are calculated, it is still necessary to determine, which robot has been detected and where is the front of the robot. For this purpose individual robot patch is utilized. This is done by checking

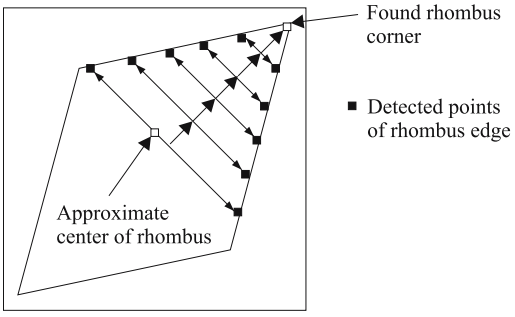


Fig. 4. Scheme of finding sharp rhombus corner

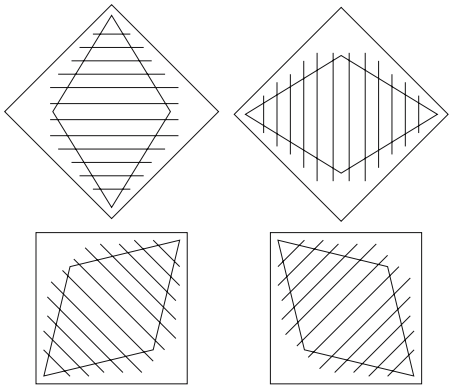


Fig. 5. Distinguished directions of rhombus edges detection

in which corner of rhombus individual patch is located and from this front of the robot is determined.

4.4 Determining Position of the Ball and of Opponent Robots Team

Second object, which has to be found by image processing algorithm is the ball. It is the easiest object to recognize on a playground. On image it is represented by circular group of pixels with orange–red color. To determine center of the ball, i.e. ball position on the image, we use a mask of cross shape with width of one pixel and length almost two times greater than diameter of the ball on image. Center of the mask is put on starting point, which was found during image search. After that, point with coordinates equal to arithmetic average of pixels coordinates under the mask, which has color assigned to ball, is determined.

Last group of objects, which should be found and recognized, are opponent team robots. Because in practice different patch markers for different teams are used, positions of robots are estimated by using opponent team color patches. Center of the opponent team patch is determined similar to searching of approximate center of rhombus patch of own team players. Edge detection is done on the base of opponent team color, rather than luminance, because for some configuration of patches results may be incorrect.

5 Measurement Results

During the test 100 measurements were done per several position of a recognized object. For example, the test rhombus patch on black square marker was arranged that longer diagonal of rhombus was at an angle of 45° to image lines. Based on real and measurement values, maximum measuring error as

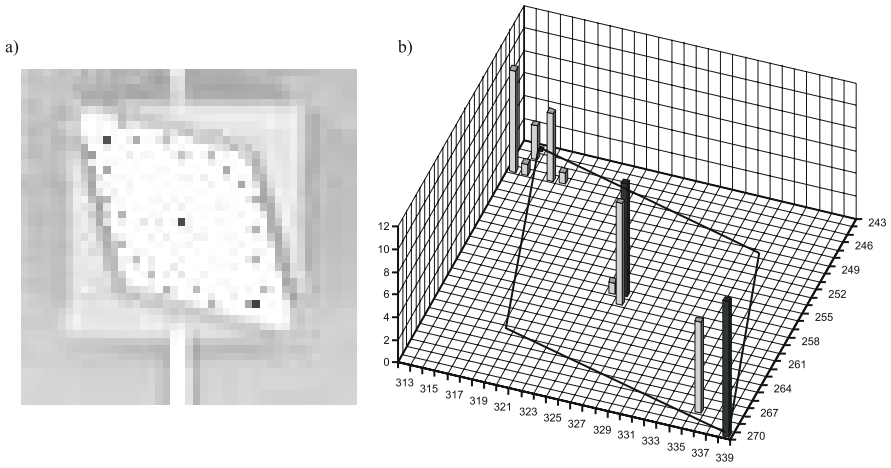


Fig. 6. Photo of patch marker with plotted points for example object position (**a**) and its measurement histogram (**b**)

the highest difference between right values using 100 done measurements were determined, which are:

- first rhombus corner – X coordinate: 3 pixels, Y coordinate: 3 pixels;
- second rhombus corner – X coordinate: 3 pixels, Y coordinate: 2 pixels;
- center of rhombus – X coordinate: 1 pixel, Y coordinate: 1 pixel;
- orientation: 0.08 rad.

In Figure 6(a) edge points (dark) and found corners and center of rhombus (black) are shown, superimpose on the photo of robot marker for example position and orientation measurement. Figure 6(b) present histogram of pixels found as corners and center of rhombus for 20 measurements. Dark shade represent real points of corners and center of rhombus.

References

1. Corke P.I. (1996) Visual control of robots. Wiley, New York
2. Gonzales R., Woods R. (1993) Digital image processing. Addison-Wesley Publication Company, MA
3. Haralick R.M., Shapiro L. (1992) Computer and robot vision. Addison-Wesley Publication Company, New York
4. Hutchinson S., Hager G.D., Corke P.I. (1996) A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pp. 651–670
5. Dobroczyński D., Dutkiewicz P., Jedwabny T., Kozłowski K., Majchrzak J., Niwczyk G. (2000) Strategy planning for mobile robot soccer. Proceedings of the World Automation Congress, 6 pp. (CD-ROM)

6. Dobroczyński D., Dutkiewicz P., Jedwabny T., Kozłowski K., Majchrzak J., Niwczyk G. (2000) Mobile robot soccer. Proceedings of the 6th International Symposium on Methods and Models in Automation and Robotics, pp. 599–604
7. Kiełczewski M. (2000) Hardware and software implementation of position and orientation measurement algorithms of 2D objects. MS Thesis, Chair of Control, Robotics and Computer Science, Poznań University of Technology, Poznań, (in Polish)
8. Kwolek B., Kapuściński T., Wysocki M. (1999) Vision-based impementation of feedback control of unicycle robots. Proceedings of the 1st Workshop on Robot Motion and Control, pp. 101–106
9. Pearson D. (1991) Image processing. McGraw-Hill Book Company, London
10. Sonka M., Hlavac V., Boyle R. (1994) Image processing, analysis and machine vision. Chapman & Hall, London
11. Shim H.S., Jung M.J., Kim H.S., Choi Kim J.H. (1997) Development of vision-basic soccer robots for multi-agent cooperative systems. Proceedings of the Micro-Robot World Cup Soccer Tournament, pp. 29–34
12. MIL-Lite version 6.0. User's guide and command reference. HTML page `www.matrox.com`
13. MIL/MIL-Lite version 6.0. Board-specific notes. HTML page `www.matrox.com`
14. HTML page `www.fira.net`

Vehicle Teleoperation with a Multisensory Driving Interface

Mario Maza, Santiago Baselga, and Jesús Ortiz

University of Zaragoza, Department of Mechanical Engineering, (CPS) C/ Maria de Luna, s/n; 50018 Zaragoza, Spain
mmaza@unizar.es

Abstract. This paper describes the technology developed in a recent research project for the teleoperation of wheeled ROV's from a multisensory driving interface. This driving interface is mounted on a motion base to generate the whole body motion sensation. To reach such objective, it uses the same motion sensation generation algorithms developed for driving simulators.

Key words: Teleoperation, motion sensation, driving interface.

1 Introduction

The exploration of hostile or extreme condition zones, polluted sites, or any mission which could be dangerous for men are activities which could be part of the field of application of teleoperated vehicles. Some systems for vehicle teleoperation can also be found in the entertainment sector, where the goal can be, for instance, virtual visits to underwater sites or distant famous ruins, etc.

The teleoperation technology described in this paper has been developed in the ***TELEDRIVE*** project, funded by the European Commission under the programme IST- 1999–57451, which intends the development of a highly efficient vehicle teleoperation system suitable for remote interventions, entertainment and research.

2 System Description

2.1 Introduction

The teleoperation chain implemented in ***TELEDRIVE*** is a complex system formed by different parts or subsystems (Fig. 1).

The **vehicle operator** is the person who tele-drives the remote vehicle (ROV).

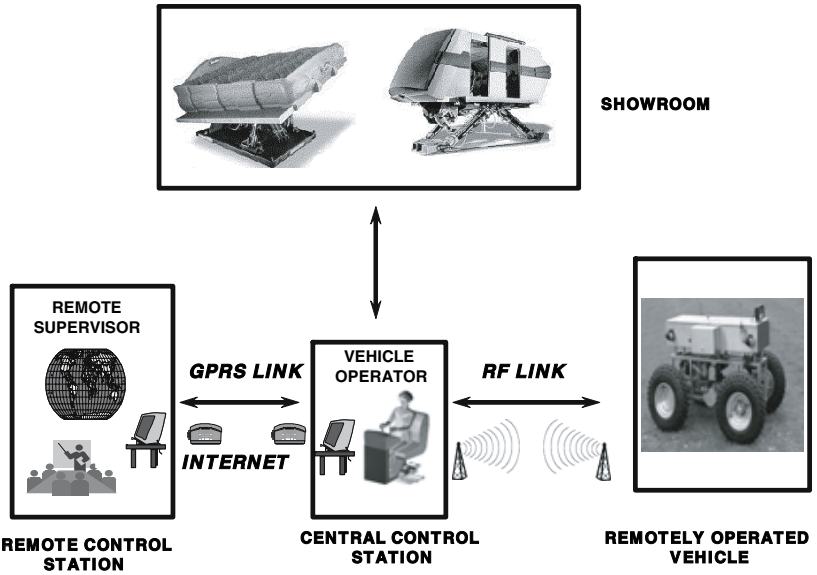


Fig. 1. ***TELEDRIVE*** layout

The **supervisor** is the person who controls the operation of the tool, system or end-effectors installed in the ROV (remote interventions) or just receives data from the sensors and cameras installed on the vehicle and checks operational safety (entertainment applications).

The **virtual passengers** are the people installed in a cabin named **showroom**. This cabin is mounted on a motion base. These virtual passengers receive images from the ROV and also motion stimuli by the replication of the ROV's motion done by the motion base.

The vehicle operator controls the motion of the ROV from a driving post called **central control station** (CCS). Such post is mounted on a motion base. In this way, the driver watches the images taken by the ROV's embarked cameras and at the same time feels the ROV's motion, increasing in this way the sensation of telepresence.

The **remote supervisor** in the remote control station is in charge of the surveillance and operational safety of the system.

The purpose of the **remote control station** (RCS) is the implementation of a post from where the general operation of all ***TELEDRIVE*** subsystems can be monitorised, mainly for safety purposes. The main information flow among the different subsystems is shown in Fig. 2.

2.2 Showroom

The showroom is another important part of ***TELEDRIVE*** system. It is formed by a closed cabin mounted on a motion base. The cabin has enough

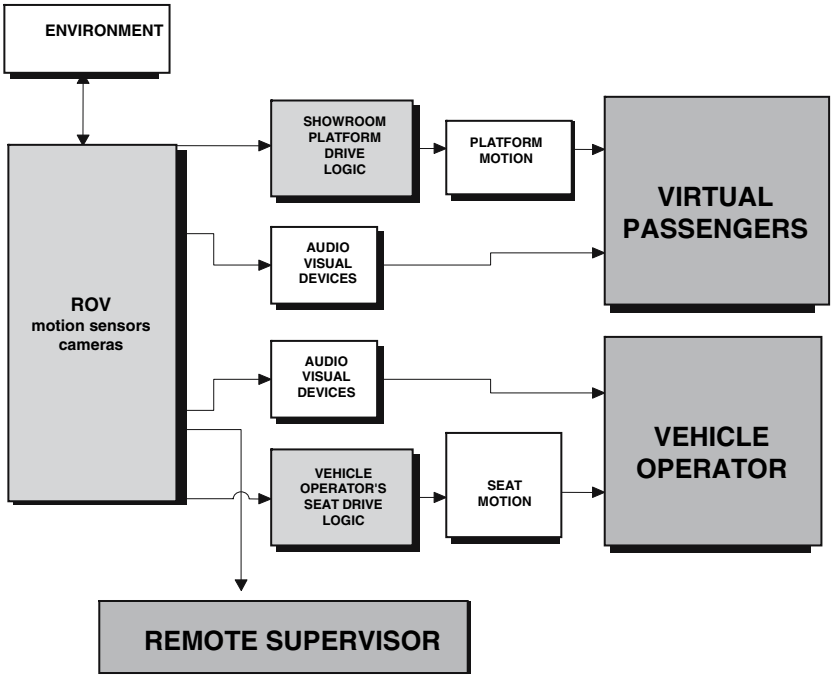


Fig. 2. Data flow among subsystems

capacity for five seated people, "the virtual passengers". It is made of light composite materials, such as glass fiber + epoxy resin.

The cabin is mounted on a motion base (hexapod or Stewart platform) and equipped with projection screens to communicate audiovisual cues to the virtual passengers. Its basic purpose is to induce people to the sensation of

Fig. 3. Showroom cabin

actually being riding the vehicle, and at the same time, project the images taken by the embarked ROV's cameras. Such real images are enhanced by means of augmented reality applications, making possible the overlaying of additional data or information, so it can be adequate for educational purposes.

2.3 Remotely Operated Wheeled Vehicle (ROV)

The wheeled ROV is a key part of the system. A general view of this ROV can be seen in Fig. 4.

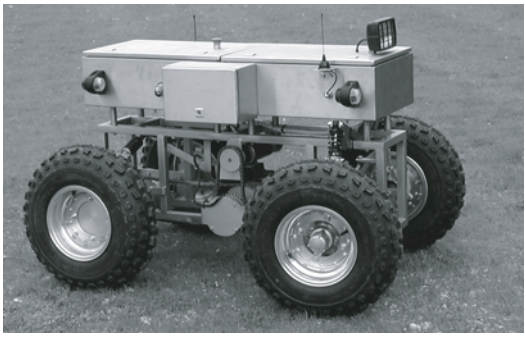


Fig. 4. Teleoperated wheeled vehicle

This vehicle has been designed as a compact exploration rover with important all-terrain and manoeuvrability capabilities. It is electrically driven and it is equipped with a 4-wheel-drive, 2-differentials transmission system. Driving control includes forward-backward motion selection, variable speed up to 12 Km/h, steering and 4×4 or 4×2 traction. Its suspension system is also adapted to rough terrains with important unevenness.

This vehicle has been tested in rough conditions showing remarkable motion capabilities. The wheeled ROV is driven from a control unit integrated in the CCS. This unit is a RF DTRII emitter (Fig. 5). It is formed by two joysticks (forwards-backwards, left-right), five buttons (horn, lights, and other auxiliary functions) and the emergency stop.

Inside the vehicle a receiver unit is installed. It is a DTRII unit composed of a high frequency receiver, a control system and a transmitter (Fig. 6). Its role is to give the information it receives from the combi-synthesized emitter to the vehicle drive and steering systems.

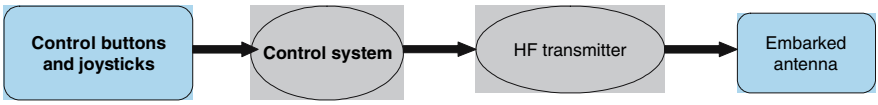


Fig. 5. Emitter scheme

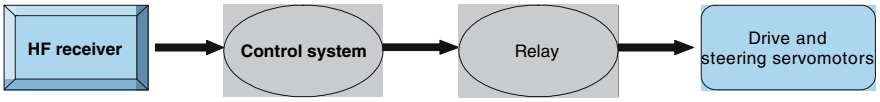


Fig. 6. Receiver unit

The wheeled vehicle is also equipped with a standard CCD colour camera F1.2/4 mm. Audio/video transmission is done using a wireless A/V sender, 2.4 GHz, range 130 m.

The ROV's motion reconstruction at the CCS and Showroom is a key part of this project. To make possible a realistic ROV motion replication, embarked sensors should therefore acquire exactly the relevant kinematic magnitudes: angular velocity (3-D) and linear acceleration (3-D). The sensor system selected for the wheeled ROV is formed by the following elements:

- 3-axis accelerometer. Range: 0-2 g's, 0-50 Hz. Sensibility 500 mV/g.
- 3 orthogonal angular-rate sensors. Range: 0-150 rad/s. Range $+-100°$/seg. Coriolis type.
- 2- axis inclinometer. Range $+-50$ deg.

The function of the 3-axis accelerometer and the 3 orthogonal angular-rate sensors is obvious. The function of the 2- axis inclinometer is to complement, calibrate and check the signals from the accelerometers.

2.4 Communication Links Among ROV's and Operator/Supervisor/Showroom Stations and Global Control System

Several parts and subsystems form ***TELEDRIVE***. There is a number of communication links amongst them. The main ones are represented in the following diagram (Fig. 7).

2.5 Central Control Station Motion Base

The CCS motion base has 4 degrees of freedom that can be easily identified in Fig. 8: Roll (rotation around x-axis), pitch (rotation around y-axis), yaw (rotation around z-axis), heave (rotation along z-axis).

There are two main points which define this spherical architecture:

- **The center point,** located at the driver's head. The location at this specific point is important to avoid linear accelerations of the head when the seat rotates.
- **The basis point.** This point moves on a spherical surface to provide roll and pitch rotations.

A pictorial view of the CCS is given in Fig. 9.

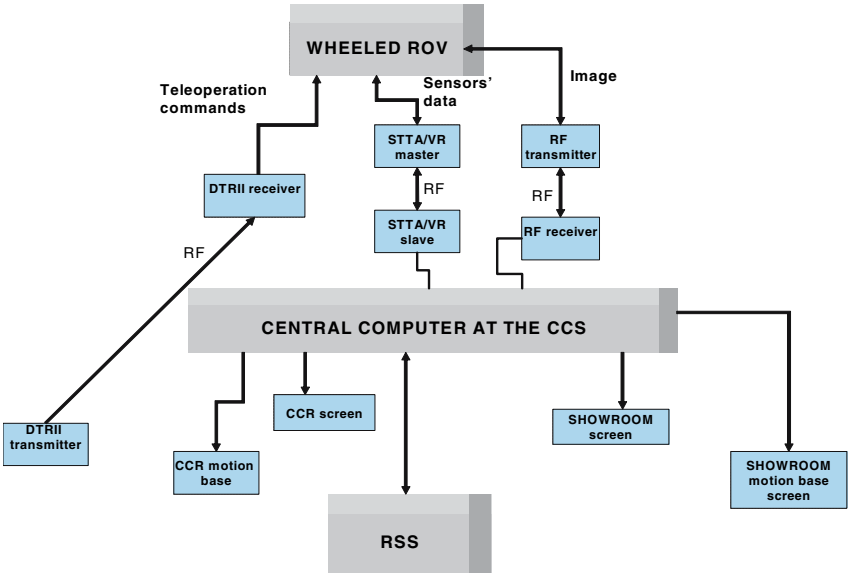


Fig. 7. Main communication links

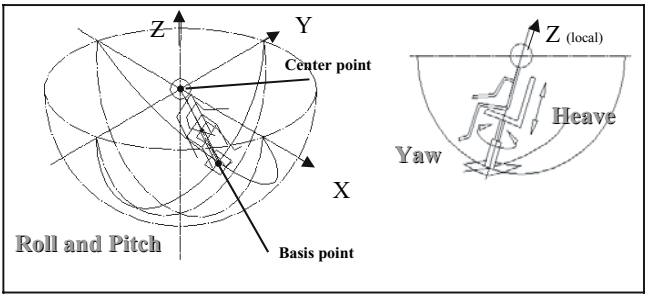


Fig. 8. Central control station's motion base architecture

2.6 CCS and Showroom Motion Based Drive Algorithm

The strategies used to generate motion sensation at the CCS and showroom are derived from those developed for driving simulators (Fig. 10). [1]

They are based on the way the human being perceives motion [2, 3] with the vestibular organ. This organ contains two main physiological sensors:

– *The otoliths, sensitive to specific forces.* The specific force is defined as the vector difference between translational inertial acceleration and gravity acceleration: $f = a - g$. As an example, riding a vehicle moving at constant speed on a level flat road, we sense the $\boldsymbol{f}$ vector pointing upward perpendicular to the road. Accelerating on the same road $\boldsymbol{f}$ points upward but with some forwards inclination.

Fig. 9. Central control station's motion base

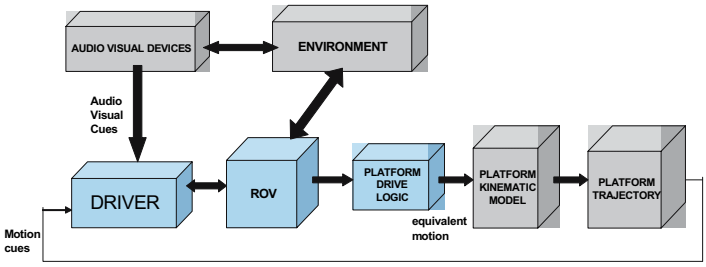


Fig. 10. Motion generation at the CCS and the showroom

– *The semicircular canals, sensitive to angular velocity.* The semicircular canals are not equally sensitive to the whole possible spectrum of angular velocities. Different authors have established perception threshold and bandwidth.

It is assumed then that human motion perception involves only specific forces and angular velocities. Thus, the CCS and showroom motion bases simulate only these two kinematic parameters from all defining the motion of the ROV. The way the simulator motion bases operate can be described as follows:

– The vehicle angular velocities are high pass filtered and then reproduced by the platform. In this way, only the frequencies inside the human motion perception bandwidth are considered.
– The specific forces are divided into different components:

(1) High frequency components of longitudinal and lateral specific forces which are associated to short displacements of the ROV, are simulated by

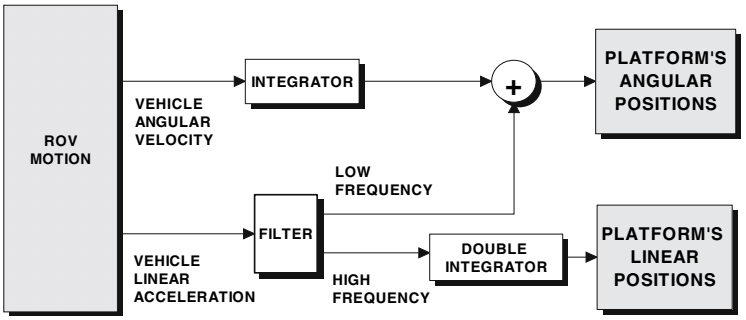


Fig. 11. The wash-out algorithm

means of longitudinal or lateral displacements of the platform. They are basically reproduced.

(2) Low frequency components of longitudinal and lateral specific forces, associated to ROV long displacements, can not be reproduced as they would exceed the excursion limits of the platform. They are then simulated by platform tilting. The tilting angle is achieved at a rotational speed below the human perception threshold.

This strategy, called the wash-out algorithm, is summarised in the following scheme:

3 Tests of the System

The system described in this paper has been extensively tested, with the ROV running on different environments. The quality of sensations in the CCS and the showroom proved to be fairly good.

One of the weakest points is the travel speed of the ROV since its maximum driving speed must be kept under 12 km/h. Satisfactory results have been obtained with the vehicle moving at 10 km/h. Once this speed has been surpassed, the expected synchronization between the motion bases movement and the images taken from the ROV embarked cameras is lost and a delay appears.

The appearance of this delay is mainly caused by the time required for acquiring and processing the ROV motion parameters in the PC as well as the time needed for the calculation of the platform motion and position control.

4 Conclusion

This paper describes an innovative architecture used for a vehicle teleoperation system, with special emphasis on the reconstruction of the ROV motion sensation.

For that purpose two motion bases are used: one for the vehicle teleoperator and a second one for virtual passengers or non-interactive users of the system.

The results obtained have revealed the feasibility of the reconstruction of the ROV motion in the teleoperation process.

References

1. Sinacori, J.B., "A Practical Approach To Motion Simulation", AIAA paper No. 73–931.
2. Young, L.R., Meyri, J.L., Newman J.S. and Feather, J.E., "Research in Design and Development of a Functional Model of the Human Nonauditory Labyrinths", Aerospace Medical Research Laboratory, TR-68–102, March 1969.
3. Meiry, J.L., "The Vestibular System And Human Dynamic Space Orientation", NASA CR–628, Oct. 1966.

Approaches to the Generation of Whole Body Motion Sensation in Teleoperation

Mario Maza, Santiago Baselga, and Jesús Ortiz

University of Zaragoza, Department of Mechanical Engineering, (CPS)
C/Maria de Luna, s/n; 50018 Zaragoza, Spain
mmaza@unizar.es

Abstract. This paper makes a review of the motion perception models and motion sensation generation strategies in VR systems, especially in driving simulators. It also presents how, by using similar strategies to the ones used in driving simulators, it is possible to make a reconstruction of the ROV motion in the teleoperation interface. Finally, it presents three different motion base mechanical architectures, to be used as motion generation platforms of the teleoperation interface.

Key words: Teleoperation, motion sensation, driving interface.

1 Introduction

Teleoperation, the remote operation of a vehicle or robot (ROV: remotely operated vehicle), is traditionally used in situations where a human controller is required but their presence at the location may be hazardous. Some examples include tactical urban reconnaissance, underwater activities, surveillance, bomb disposal, planet exploration, contaminated site cleanup, etc. Fig. 1 shows a typical control station for mining equipment [1, 2]. Figure 2 shows a control station for unmanned aircraft [3].

A teleoperation interface, commonly known as central control station (CCS), can be made more natural and easier for the operator by creating a feeling of presence, for example, by using stereo video cameras that follow the movements of the user's head [4]. In this way, the operator's vestibular and kinesthetic sensations are coupled to the visual information from the ROV.

The current teleoperation interfaces commercially used are relatively unsophisticated, having two basic components: the vision system and the control interface. The control interface is often an adaptation of the earlier generation of remote controllers and the vision system consists of usually two screens giving simultaneous rear and front views. The replication of the ROV's motion

Fig. 1. NUMBAT surface control station

Fig. 2. PREDATOR control station

at the control station, for instance by sitting the operator on a motion base or active seat, is an issue hardly reported in the technical literature.

This paper analysis the way the human being perceives motion (without visual cues) and how motion sensation on the CCS can be generated.

2 Human Sensory System and Motion Perception

Previous to the development of any ROV motion replication algorithm at the CCS, it is necessary to have models predicting the sensation of actual physical motion by humans (without visual cues). The main perception models were established in the sixties and seventies, completed with some recent studies. Although many sensors throughout the body play a role in this sensing process, it is apparent that the vestibular system located in the head (internal ear) provides the dominant signals [1]. It was established that human motion perception involves only angular velocity and linear acceleration (specific force).

2.1 Angular Velocity Perception

The primary sensing organs for rotational motion are the semicircular canals, located in the internal ear.

Table 1. Rotation Motion Sensation Model Parameters

	Pitch (about y-axis)	Roll (about x-axis)	Yaw (about z-axis)
T_L(s)	5.3	6.1	10.2
T_S(s)	0.1	0.1	0.1
T_a(s)	30	30	30
Threshold (deg/s)	3.6	3.0	2.6

For the frequency range of interest in the present paper, the model of Young et al [5] was found the most adequate. The corresponding transfer function for the linear portion of the model is (for any one channel):

$$\frac{\hat{\omega}}{\omega} = \frac{T_L T_a s^2}{(T_L s + 1)(T_S s + 1)(T_a s + 1)} \tag{1}$$

where:

- ω is the angular velocity input about one of the three axis
- $\hat{\omega}$ is the perceived angular velocity.

A number of past experiments have been performed to determine the values for the parameters in this model. They are listed in the following Table 1: [6].

It can be seen that this system is a good sensor of angular velocity in the frequency band 0.2 rad/s to 10 rad/s. Its response $\hat{\omega}$ tends to zero for both steady-state angular velocity and steady-state angular acceleration inputs.

2.2 Acceleration Perception (Specific Force)

Specific force is felt by the otolith (Fig. 3) (the other major sensor contained within the vestibular system) complemented by the perception performed by muscles and tendons through contact pressure.

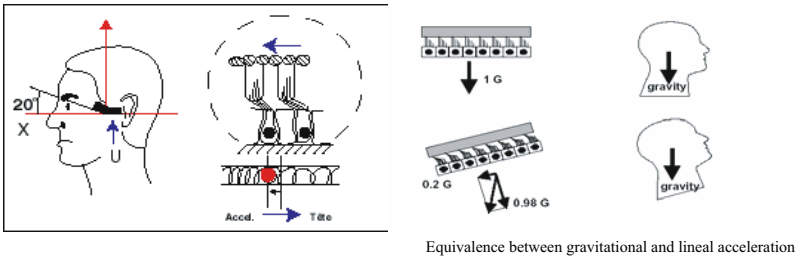


Fig. 3. The otolith specific force perception

Table 2. Specific Force Sensation Model Parameters

	Surge (along x-axis)	Sway (along y-axis)	Heave (along z-axis)
$\tau_L(s)$	5.33	5.33	5.33
$\tau_S(s)$	0.66	0.66	0.66
$\tau_a(s)$	13.2	13.2	13.2
K	0.4	0.4	0.4
Threshold (m/s^2)	0.17	0.17	0.28

Specific force $\vec{f}$ is defined as the vector difference between translational inertial acceleration $\vec{\gamma}$ and the acceleration due to gravity $\vec{g}$, that is:

$$\vec{f} = \vec{\gamma} - \vec{g} \tag{2}$$

In terms of forces, specific force is the sum of the vehicle's external forces divided by the vehicle mass less the gravitational components.

The corresponding model and transfer function for the linear portion of the model is (for any one channel):

$$\frac{\hat{f}}{f} = \frac{K\left(\tau_a s + 1\right)}{\left(\tau_l s + 1\right)\left(\tau_S s + 1\right)} \tag{3}$$

where:

- f is the specific force along any one of the three axis.
- $\hat{f}$ is the perceived specific force along any one of the three axis.

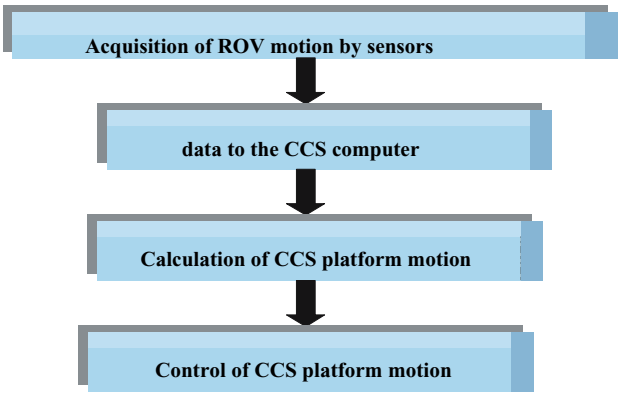
Based on this work Mairy and Young [7] produced the parameter values listed in Table 2:

It can be seen that the system is a good sensor for specific force in the frequency band of 0.2 rad/s to 2 rad/s. It has a steady response to constant values of $\vec{f}$. Because this system will respond to the gravity vector alone, when the translational inertial acceleration of the head is zero, it can be used to determine one's tilt orientation with respect to the local vertical.

3 Transmission of Motion Sensation in Teleoperation

The process followed can be summarised as:

The replication of the ROV (remotely operated vehicle) motion at the CCS (central control station) can exclusively be made by using a motion base or platform. As the CCS will not exactly replicate the ROV motion, it is necessary to generate a motion sensation similar to the sensation the driver would have being in the actual ROV. To reach such objective, the use of driving simulators motion base algorithms is a valid strategy.



Ideally, any ROV motion, to be properly replicated, needs a full 6-DOF motion at the CCS platform. It is important however to consider the motion replication fidelity that can be achieved with a reduced number of DOF's of the motion base.

The applicability of such simplified systems for flight simulators has already been analyzed in detail in some publications [8–10]. The platform architectures considered were always parallel-type. The conclusion achieved by Pouliot et al. [8] was that, in most cases, a 3-DOF's motion base was capable of producing a motion sensation quality comparable to that obtained with the classical 6-DOF's Stewart Platform [11].

4 Development of the CCS Motion Base Drive Logic Algorithm

In motion sensation reconstruction, two are the main objectives:

Let S_{cpc} a frame attached to the ROV and S_{cps} a frame attached to the CCS.

$$\left[\overrightarrow{f} \text{ on the ROV}\right]_{S_{cpc}} = \left[\overrightarrow{f} \text{ on the CCS}\right]_{S_{cps}} \tag{4}$$

That is:

Specific force at the origin of frame S_{cpc} (head of ROV driver), in frame S_{cpc}

Equal to

Specific force at the origin of frame S_{cps} (head of CCS driver), in frame S_{cps}

$$\left[\overrightarrow{\Omega}_{Abs \text{ on the ROV}}\right]_{S_{cpc}} = \left[\overrightarrow{\Omega}_{Abs \text{ on the CCS}}\right]_{S_{cps}} \tag{5}$$

ROV angular velocity in frame S_{cpc}

Equal to

CCS angular velocity in frame S_{cps}

By keeping these two objectives it is possible to reconstruct the ROV motion in the CCS.

Fig. 4. Wheeled ROV used in the tests

5 Architectures for the CCS Motion Base

This section shows three different architectures used to provide motion sensation in the CCS. The first one is a classical 6-DOF system. The other two are simplified 4-DOF systems able to provide a motion sensation quality similar or even greater than that given by the complete 6-DOF systems. The three platform architectures described below have been tested with optimal results for the teleoperation of the wheeled ROV shown in Fig. 4.

6 DOF Stewart Platform

This is the most classical architecture. The Stewart platform is a universal 6-DOF motion base, used in flight-driving simulators, robotics, earthquakes research, etc. It consists of a moving platform attached to a fixed base by means of six legs. Each of the legs consists of – from base to top – a fixed unactuated Hooke joint, an actuated prismatic joint and an unactuated spherical joint, attached to the moving platform. The mentioned prismatic joints are typically hydraulic or electric linear actuators. The architecture of this classical six degree-of-freedom parallel mechanism is showed in Fig. 6. It was first presented in 1965 [11]. Figure 5 shows a driving simulator using the Stewart platform motion base, which was tested for the teleoperation of the ROV shown in Fig. 4.

6.1 Spherical Platform

This is a 4-DOF motion base specially adapted to the replication of wheeled ROV's motion. There are two main points which define this spherical architecture (Fig. 6):

Fig. 5. Implementation of the Stewart platform

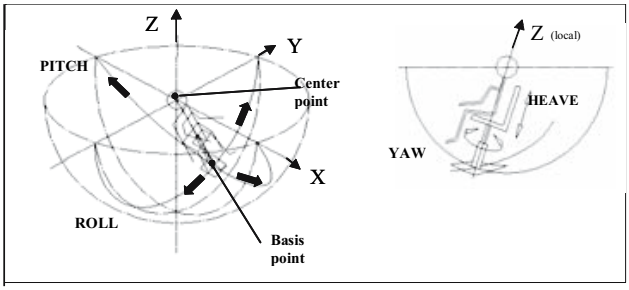


Fig. 6. The Spherical Platform architecture: heave displacement and yaw rotation

- **The center point**, located at the driver's head. The location at this specific point is important to avoid linear accelerations of the head when the platform rotates.
- **The basis point**. This point moves on a spherical surface to provide roll and pitch rotations.

In this way, the Spherical Platform has 4 DOF (Fig. 7):

- Roll (rotation around x-axis)
- Pitch (rotation around y-axis)
- Yaw (rotation around z-axis)
- Heave (displacement along z-axis)

A practical implementation of this architecture is shown in Fig. 7.

6.2 Parallelogram Platform

This 4-DOF platform represents, as the previous case, an innovative approach to the problem of motion sensation generation. The system is formed by:

Fig. 7. Implementation of the spherical platform

– two four-bar mechanisms, the first one on the floor, supporting the whole system, the second one mounted on top of the first one and perpendicular to it. These systems provide roll and pitch inclination.
– A rotation mechanism based on a slew ring, to provide yaw rotation.
– A scissors-type mechanism, to provide heave motion.

A pictorial view of this system is shown in Fig. 8.

Fig. 8. Implementation of the parallelogram platform

6.3 Comparison of the Motion Generation Capacity of the 3 Architectures Presented

The sensation of being accelerating on a motion base is proportional to the tilting angle, that is, using the gravity vector as it was previously explained. Table 3 compares the motion capacity of the three above described architectures:

Table 3. Platform comparison

Maximum Excursion			
Dof	Hexapod	Spherical	Paralellogram
Roll	±22 deg	±90 deg	±22 deg
Pitch	±22 deg	±90 deg	±22 deg
Yaw	±22 deg	±180 deg	±180 deg
Surge	±0, 3 m	0	0
Sway	±0, 3 m	0	0
Heave	±0, 2 m	±0, 3 m	±0, 3 m

It can be easily appreciated that the spherical architecture provides much greater angular excursion, thus being able to generate bigger acceleration sensation.

7 Conclusion

This paper explains why most of the technology developed for the driving simulators can be directly applied in vehicle teleoperation systems, especially all that concerns the algorithms used to replicate the ROV motion using a motion base, as well as the different architectures of the motion base itself.

The paper presents and compares three different motion base architectures used for vehicle teleoperation, the first one a complete 6-DOF system, the other two simplified 4-DOF systems. The three of them have proved their functionality to perform this task.

References

1. D.W. Hainsworth "Teleoperation User Interfaces For Mining Robotics" IEEE International Conference on Robotics and Automation. San Francisco – California. April 2000.
2. J.C. Ralston and D.W. Hainsworth. "The Numbat: A Remotely Controlled Mine Emergency Response Vehicle", *Proceedings of the International Conference on Field and Service Robotic*s, pp. 48–55, Canberra, Australia, December 1997.

3. Capt Mark H. Draper, Ph.D. Heath A. Ruff "Multi-Sensory Displays and Visualization Techniques Supporting the Control of Unmanned Air Vehicles" IEEE International Conference on Robotics and Automation. San Francisco – California. April 2000.
4. "Improving Pilot Dexterity with a Telepresent ROV". Philip J. Ballou, Senior Vice President, Deep Ocean Engineering, Inc., USA
5. Young, L.R., Meyri, J.L., Newman J.S. and Feather, J.E., "Research in Design and Development of a Functional Model of the Human Nonauditory Labyrinths", Aerospace Medical Research Laboratory, TR-68-102, March 1969.
6. Gum, D.R., "Modelling Of The Human Force And Motion Sensing Mechanisms", AFHRL-TR-72–54, June 1973.
7. Meiry, J.L., "The Vestibular System And Human Dynamic Space Orientation", NASA CR-628, Oct. 1966.
8. Pouliot Nicolas A., Meyer A. Nahon, Clément M. Gosselin "Analysis And Comparison Of The Motion Simulation Capabilities Of Three-Degree-Of-Freedom Flight Simulators". AIAA paper No. 93-3474-CP, 1993.
9. Shiabev, V.M., "New concept of the motion system for a low-cost flight simulator: Development and design", AIAA FST Conf. 1993.
10. Kurtz R., Haywarda V., "Multiple Goal Kinematic Optimization of a Parallel Spherical Mechanism with Actuator Redundancy" IEEE Trans. On Robotics and Automation, Vol. 8, No. 5, 644–651. October 1992.
11. Stewart D., "A Platform with Six Degrees of Freedom". Proceedings of the Institution of Mechanical Engineers. Vol. 180, No. 5, pp 371–378. 1965.

Virtual Platform for Land-Mine Detection Based on Walking Robots

A. Ramirez, E. Garcia, and P. Gonzalez de Santos

Industrial Automation Institute (CSIC) 28500 Madrid, Spain
aramirez@iai.csic.es

Summary. This work presents a virtual platform created for the SILO6, a six-legged walking robot focused on the localization of antipersonnel landmines. The virtual platform is a Java-based module which includes the new concepts about human – machine interfaces design, and it's main purpose is to satisfy three basic requirements: the robot's state monitoring, the robot's task definition and monitoring and the robot teleoperation.

Key words: Virtual Platform, Humanitarian De-mining, Virtual Reality.

1 Introduction

Today, human-machine interfaces are designed based on the user's necessities incorporating effectiveness, efficiency and productivity, being the machine the one who comes closer to the human necessities. Robotics applied to the detection of antipersonnel landmines is a technological field in which friendly human-machine interfaces are quite important. The nature of the task requires much more participatory interfaces focused on the user's and the task's requirements.

The virtual platform exposed in this paper is part of the DYLEMA project held by the IAI-CSIC [1], which is intended to develop a de-mining system carried out by a six-legged robot called SILO6. The virtual platform created for this project is aimed at achieving an efficient, intuitive and pleasant communication between the human and the robot. For this purpose, the virtual platform consists of:

- A three dimensional virtual image of the robot. It simulates the robot's real movement, although the robot is far away from the operator station, the user can visualize each movement of the robot's joints and links.
- A two dimensional view of the robot and the minefield. It shows the robot's trajectory by marking the robot's covered area and the localized mines during the robot's task performance.

- A data base. This data base stores all the information related to the task (mined land, robot's sate, and localized alarms) and it is accessible to the user at every time.
- Communication protocol based on TCP/IP. The protocol is in charge to comunicate the robot and the operator station.

This paper focuses on the visual feedback of the virtual platform which uses non inmersive virtual reality and has been created using Java 3D, which is a top-down approach for building 3D interactive programs. Programs written in Java 3D can be run on several different types of computer, in a web browser or as stand-alone applications. Built on top of the Java programming language, it uses a scene graph hierarchy as the basic structure. Various geometrical representations, animation/interaction, lighting, texture, collision detection, and sound are also supported [2].

The SILO6 virtual platform's visual feedback has two main parts:

- The 3D View: monitories the robot's state.
- The Planar View: defines and monitories the robot's task.

In this paper, Sect. 2 presents the design and development of the 3D view and Sect. 3 presents the construction of the planar view. Finally, Sect. 4 shows some experiments and conclusions.

2 Robot's State Monitoring: 3D View

Figure 1 shows the 3D View into the virtual platform. It includes a three dimensional view of the SILO6 robot and it is able to show the movement of each one of the SILO6 joints as soon as the value of their angles are received through the communication protocol.

The 3D View contained in the virtual platform has been designed following 3 main steps:

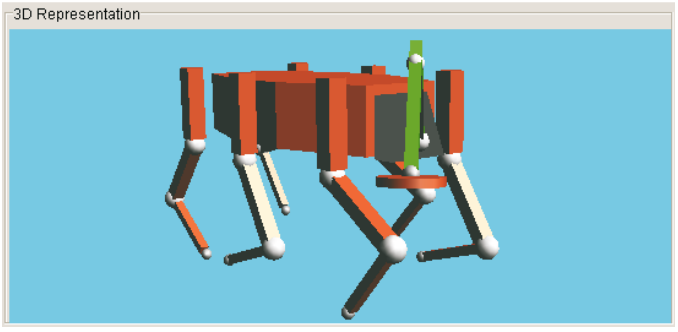


Fig. 1. 3D View of the SILO6 robot into the virtual platform

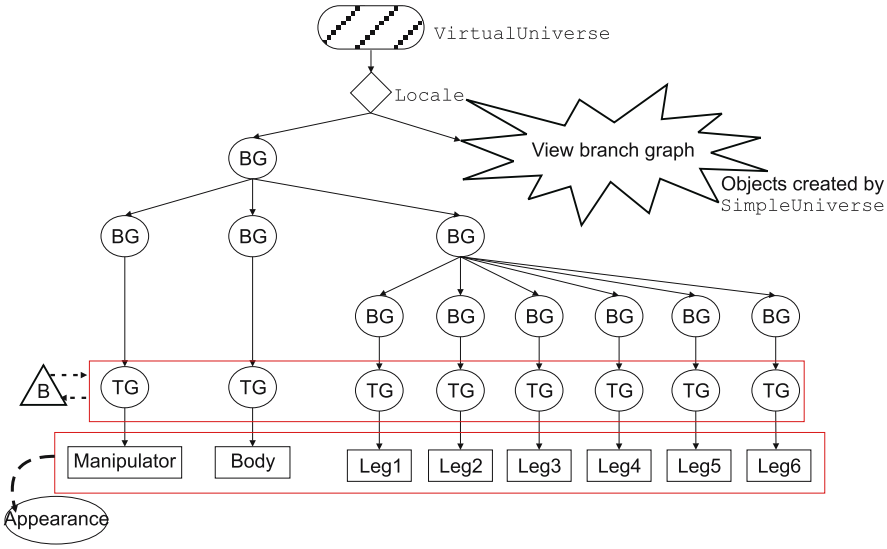


Fig. 2. Scene graph for the virtual platform: Application SILO6's 3D View

1 Building the scene graph.
2 Constructing the forward kinematic model based on the Denavit-Hartenberg representation.
3 Getting objects to move in a scene by adding animation behavior.

2.1 Building the Scene Graph

Java3D uses the scene graph approach. A scene graph is a hierarchical approach to describe objects and their relationship to each other. The scene graph holds the data that defines a virtual world, it includes low-level descriptions of the objects geometry and their appearance, as well as higher-level, spatial information, such as specifying positions, animations, and transformations of the objects, and additional application-specific data.

Figure 2 shows the scene graph created in the virtual platform for the 3D View. The view branch graph, which is represented by the large star will be described in following paragraphs. In Fig. 2, BG stands for `BranchGroup` and TG for `TransformGroup`, both classes are used for spatial transformation of geometry shapes.

2.2 Forward Kinematic Model Based on the Denavit-Hartenberg Representation

To derive the kinematic model of the SILO6 leg and the manipulator the Denavit-Hartenberg homogeneous matrix representation has been used. The

Table 1. Denavit-Hartenberg link parameters

SILO6's leg

Link	$\mathbf{a}_i$(mm)	$\mathbf{d}_i$(mm)	α_i(rad)	θ_i
1	84	0	$\pi/2$	θ_1
2	250	0	0	θ_2
3	250	0	0	θ_3

SILO6's manipulator

Link	$\mathbf{a}_i$(mm)	$\mathbf{d}_i$(mm)	α_i(rad)	θ_i
1	0	$\pi/2$	0	θ_1
2	350	0	0	θ_2
3	370	0	0	θ_3
4	0	-$\pi/2$	0	θ_4
5	170	0	0	θ_5

relevant Denavit-Hartenberg parameters for the SILO6 leg and for the SILO6 manipulator are given in Table 1. Both are obtained from the kinematic parameters of the leg and manipulator, which can be obtained from Fig. 3 and Fig. 4 respectively.

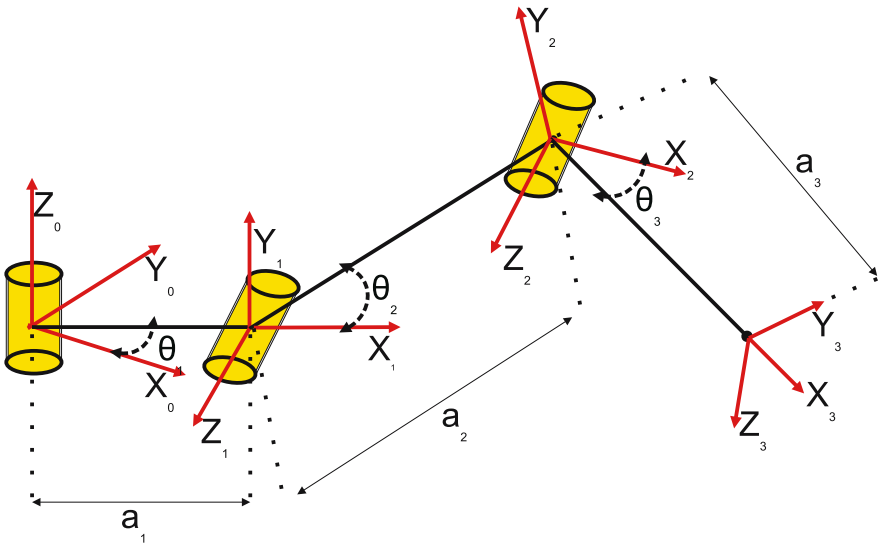


Fig. 3. The first links of a kinematic chain for any of the Silo6's legs

Finally, considering the Denavit-Hartenberg link parameters for a SILO6's leg and manipulator (Table 1) the transformation matrix will be (1) for a SILO6's leg and (2) for the SILO6's manipulator, where the terms C_i and S_i represents $\cos(\theta_i)$ and $\text{sen}(\theta_i)$ respectively, in the same way, the terms C_{ij} and S_{ij} represents $\cos(\theta_i + \theta_j)$ and $\text{sen}(\theta_i + \theta_j)$ respectively.

$$T = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & S_1 & C_1(a_3C_{23} + a_2C_2 + a_1) \\ S_1C_{23} & -S_1S_{23} & -C_1 & S_1(a_3C_{23} + a_2C_2 + a_1) \\ S_{23} & C_{32} & 0 & a_3S_{23} + a_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

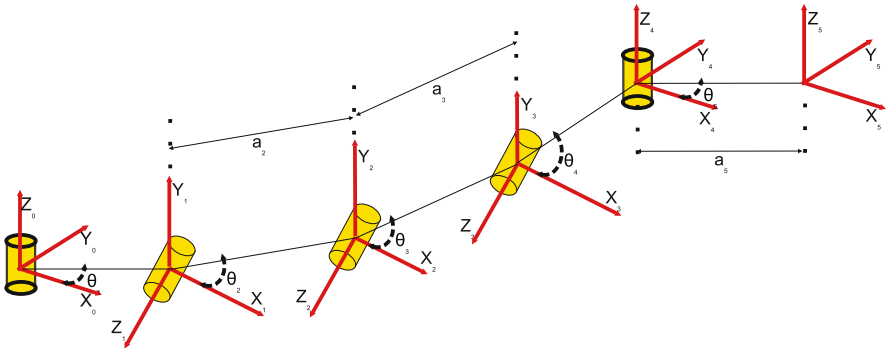


Fig. 4. The first links of a kinematic chain for the Silo6's manipulator

$$T = \begin{bmatrix} n^0 & s^0 & a^0 & p^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$p^0 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (a_5C_{234}C_5 + a_3C_{23} + a_2C_2)C_1 - a_5S_1S_5 \\ (a_5C_{234}C_5 + a_3C_{23} + a_2C_2)S_1 + a_5C_1S_5 \\ a_5S_{234}C_5 + a_3S_{23} + a_2 S_2 \end{bmatrix} \tag{3}$$

$$n^0 = \begin{bmatrix} C_{234}C_5C_1 - S_1S_5 \\ C_{234}C_5S_1 + S_5C_1 \\ S_{234}C_5 \end{bmatrix} \quad s^0 = \begin{bmatrix} -C_{234}S_5C_1 - C_5S_1 \\ -C_{234}S_5S_1 + C_5C_1 \\ S_{234}S_5 \end{bmatrix} \quad a^0 = \begin{bmatrix} -S_{234}C_1 \\ -S_{234}S_1 \\ C_{234} \end{bmatrix} \tag{4}$$

2.3 Getting Objects to Move in a Scene by Adding Animation Behavior

Based on the Denavit-Hartenberg kinematic representation (see Sect. 2.2), the next step is to translate those homogeneous transformation matrices into Java 3D code which execute the movement. The `Behavior` class is an abstract class that provides the mechanism to include code to change the scene graph, or objects in the scene graph, in response to some stimulus.

Particularizing to the SILO6 robot, each one of the legs and the manipulator have their own behavior previously defined by the rotation angles of the joints. To achieve the joints and links proper movement a custom behavior class is added, this class implements the `initialization` and `processStimulus` methods from the abstract `Behavior` class, then a behavior acts refereed to the object of change, it means the robot's links and joints.

Figure 5 shows the scene graph diagram for the content branch graph of each of the robot's legs and the robot manipulator. Both represent the geometric arrangement of the links of the robot, the *Shape Leaf* nodes represent the geometries and visual properties of the individual links that compose the

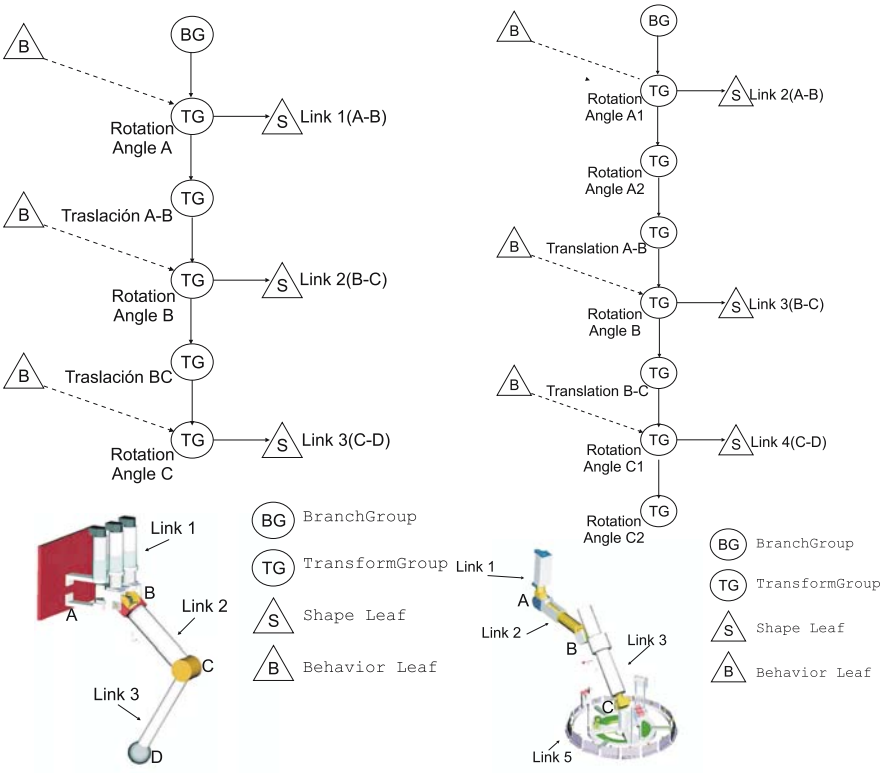


Fig. 5. Java 3D's Scene graph diagram for the content branch graph of (**a**) each of the SILO6's legs; (**b**) the SILO6's manipulator

legs and the manipulator, the `BranchGroup` and `TransformGroup` nodes describe the structure of the tree that defines the geometric relationship among them, and the *Behavior Leaf* nodes give the movement/interaction necessary to solve the kinematic problem at a virtual environment level.

3 Task Monitoring: Planar View

Figure 6 shows the SILO6 robot and the minefield in a planar view into the virtual platform. In the figure, the robot is represented by a red rectangle, the minefield by a green rectangle and the covered area into the minefield is represented by a gray shadow on the minefield. The localized alarms appear just when they are localized at the registered coordinate by the system and they are represented by a black point on the minefield, considering the nature of the DYLEMA's project, "alarms" would be considered as potential mines in this application. The figure also shows how the input parameters are given,

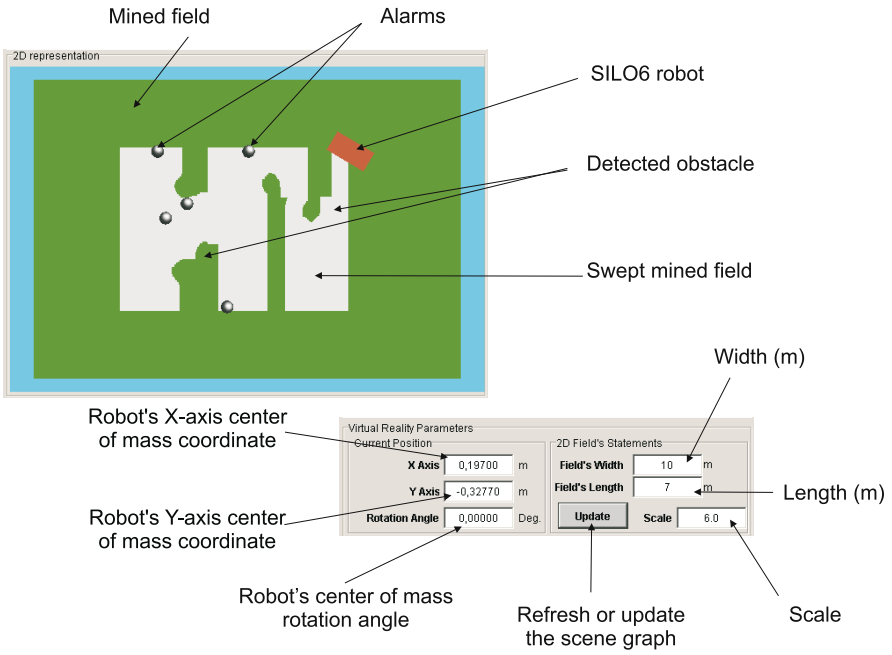


Fig. 6. The SILO6 robot and the minefield Planar View window into the virtual platform

those are the field's width and length data and the scale preferred by the user to see the task monitoring. In the same frame to the left the application shows each the robot's center of mass X and Y axis position.

This 2D View has been created almost in the same way as the 3D View has been created, it means that the scene graph has been created following the same steps and method, the difference takes place in the way how the movement into the scene has been achieved.

The movement has been created using `threads` from the `java.lang` class instead of the behavior class used for the robot state monitoring window.

4 Experiments and Conclusions

Focused on the purpose of creating a friendly, intuitive and flexible virtual platform for the operator station, 2 main experiments have been designed.

4.1 First Experiment: Task Monitoring

The firts experiment has been designed to see the virtual platform performance at monitoring the task. The robot's navigation algorithm used is fully

described in [3]. Basically the robot starts covering the minefield until it detects an obstacle, then it starts to cover the obstacle's contour.

Figure 6 besides showing the planar view into the virtual platform, it is also a picture taken while the experiment is on execution. In the picture the robot can be seen as a red rectangle covering the minefield, the minefield as a green rectangle, the covered area by a gray shadow and the localized alarms by a black spheres.

The result obtained by this experiment was a friendly, intuitive and flexible way to monitor and define the task. Through the planar view the user can continuously visualize the robot's task state, besides having an updated information about the localized alarms' positions inside the minefield.

4.2 Second Experiment: Robot's State Monitoring

The second experiment was designed to see the virtual platform performance at monitoring the robot's state and it's capacity to show at any time the robot's link and joint movement.

Figure 7 is a picture taken during the execution of this experiment, showing the main window of the virtual platform and the way how the user can monitor the robot's state, so if any abnormal movement appears while the robot is on walking it is immediately shown by the 3D View window (zoomed part) into the virtual platform and registered.

The result obtained at the end of the second experiment as well as the first experiment was a friendly, intuitive and flexible way to monitoring the robot's task and state. Through the 3D View the user can follow every robot-joint and link movement although she/he is quite far away from the robot.

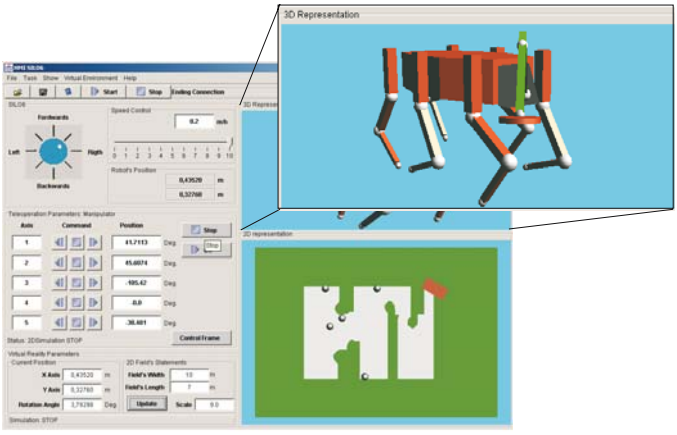


Fig. 7. SILO6's 3D View while the second experiment

References

1. P. Gonzalez de Santos, E. Garcia, J.A. Cobano, and A. Ramirez, "Silo6: A six-legged robot for humanitarian de-mining tasks," in *10th International Symposium on Robotics and Applications, World Automation Congress*, June 2004, Seville, Spain.
2. Sun Microsystems Java 3D Engineering Team, *Java 3D API Tutorial*, Sun Microsystems, 2000, http://www.unibwmuenchen.de/campus/LRT6/PDF/HumInt.pdf.
3. E. Garcia and P. Gonzalez de Santos, "Mobile robot navigation with complete coverage of unstructured environments," *Robotics and Autonomous Systems*, vol. 46, no. 4, pp. 195–204, 2004.

Vision Computer Tool to Improve the Dependability of Mobile Robots for Human Environments

C. Salinas, L. Pedraza, and M. Armada

Automatic Control Department, Industrial Automation Institute, Spanish Council for Scientific Research, Madrid, Spain

Summary. The design and realization of dependable robots for human environments, is a grand challenge the people working in robotic world have to meet. Dependability is the property that integrates attributes such as safety, security, survivability, maintainability, reliability and availability. And this is not a feature that can be added after the functional design. This work presents a tool based on principles that lead to an improved dependability of intelligent robots. This tool was developed with the main objective of aiding in the design of the vision computer system of a mobile robot performing activities related to vigilance and security control.

1 Introduction

Dependability is a concept that integrates several attributes previously mentioned [1]. In summary, the goals of the concept of dependability are the ability of a system to deliver a service that can justifiably be trusted, the capability to avoid the more frequent or severe failures and also the ability to prevent the outages which duration is longer than the acceptable to the user(s).

These concepts are needed especially for service and personal robots. This is because these robots are intended to operate in unpredictable and unsupervised environments and closed to, or in direct contact with, people moving around. There could also be people who try to harm them by disabling his sensors or try to confuse the robot.

The artificial vision can be used as a tool for analysing, evaluating and identifying the external perturbations or incidents during the performance of the activities of the robot. In this way the visual tool can help the robot to understand and interpret the events occurred in its environs and consequently the system should avoid the ambiguities of the information, mistakes that carry out the robot to take wrong and dangerous control decisions.

The external perturbations could be generated due to the illuminations problems of the environment, causing an inappropriate images acquisition and

errors in the images data. The disturbances also appear due to the brightness and the shadows formed by objects of the background.

This paper is organized as follows. The Sect. 2 is a brief overview of the computer vision systems. The Sect. 3 describes the design of the visual tool. In the Sect. 4 the experiment applying the visual tool is shown. And finally in the Sect. 5 a brief conclusion is shown.

2 Overeview of the Vision Computer System

In analogous form as in the human vision, the computer vision converts the luminous energy into electric current. This signal can be sampled and digitized for subsequent processing. The sampling provides a matrix of elements which corresponds to luminous energy values. These elements are called ***pixels*** and its value is the ***intensity*** or ***grey level***.

The image acquisition is the initial procedure of the vision systems, and as it was explained previously, it is necessary to control it, because this is the processes where more failures can take place. And these errors can not be corrected later.

The main function of the image acquisition is the digitalization. Here, the analogical output signal of the vision sensor is sampled and quantified providing a digital image. This procedure has two important parameters, which allow adjusting the range of digitalization. One is the "**offset**", which adjusts the minimum voltage or black level, adding a continue signal to the video signal. The second parameter is the "***gain***", which controls the bandwidth of digitalization, working over the gain of the amplifier. The Fig. 1 represents the signal of a visual sensor converted by an AD converter [3].

The importance of these parameters can be observed in the next figures. The Fig. 2.a. and the Fig. 2.b. display respectively, the situation of white saturation and black saturation in the images. In both cases, the information out of range of digitalization gets lost completely. The Fig. 2.c. is another example of erroneous digitalization, because it is not made a good use of

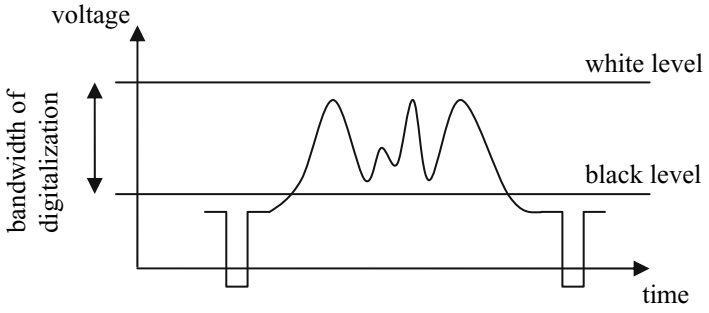


Fig. 1. Range of digitalization in an AD convertor

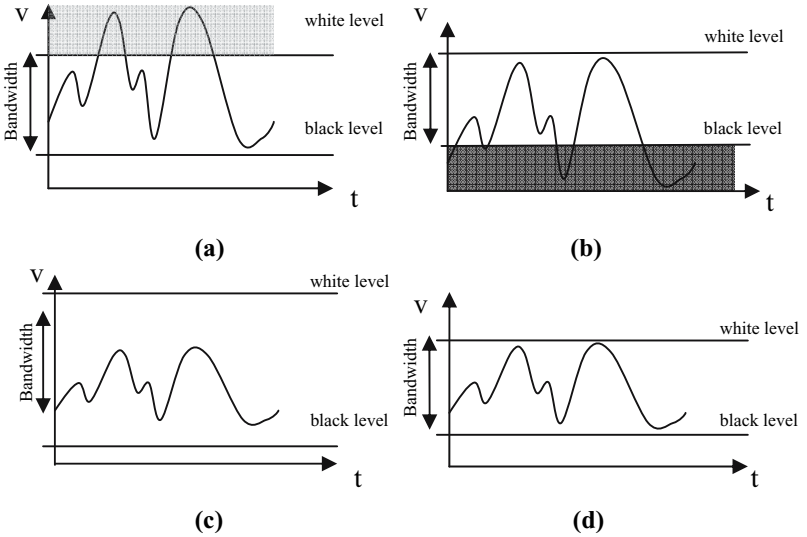


Fig. 2. (**a**) white saturation of the image, (**b**) black saturation of the image, (**c**) incorrect digitalization and (**d**) correct digitalization

the bandwidth. Consequently the image will have a low contrast. And finally the Fig. 2.d. gives an idea of a correct digitalization, which can be obtained controlling the gain and offset parameters.

It is possible to observe that the adjustment of the offset is equivalent to operating on the brightness of the images, where the brightness in terms of the image frame, is the medium value of the grey levels. And the adjustment of the gain is equivalent to modifying the contrast of the image. The contrast in terms of image frame is the medium deviation of the grey levels of the pixels.

3 Desing and Development of the Visual Tool

To develop this visual tool, we have followed some strategies and guidelines in the design process:

1. To learn from the study of the environment how to make the design to contribute to the robustness and safety of the robot.
2. To make available the adequate information to enable the robot to take intelligent decisions. This allows a natural interaction between the robot and its environment.
3. The functional design has to be oriented to obtain a flexible and adaptable tool.

First, we must ensure that the system is continuously digitizing correctly the acquired images and these images have a good quality. In this order, we

established an automatic controller based on the parameters of the digitalization. The objective of this controller is to provide an image with optimal conditions. In this way, the controller must be able to identify the disturbances and so, correct the control parameters for the following acquisitions.

For this work the following nomenclature was utilized:

$\mathbf{B_f}$: is the value of the brightness related to the image frame, which is the value of the grey level of the each pixel obtained by the digitalization. And the range of its value is [0, 255].

$\mathbf{B_{mf}}$: is the medium value of B_f.

$\mathbf{C_f}$: is the value of the contrast of the image frame. As we mentioned before it is equivalent to the medium deviation.

$\mathbf{B_h}$: is the value of the offset and the range of its value depends on the hardware or frame grabber, utilized. In this case is [0, 100].

$\mathbf{C_h}$: is the value of the gain and the range of its value depends on the hardware used. In this case is [0, 100].

We make use of the empiric method to study and evaluate the variables which represent the quality of the images acquired, such us histograms, brightness and contrast. The experiment was performed submitting the images acquired to different perturbations, changing the values of $\mathbf{B_h}$ and the $\mathbf{C_h}$. The testing consisted on oscillating the values of one of these parameters, holding constant the other parameter and computing the values of the $\mathbf{C_f}$ and $\mathbf{B_{mf}}$.

In this way, we could determine the behaviour of the variables of the images, related to fluctuations of the parameters of digitalization. And using this information, it is possible to obtain a decision criterion about the goodness of the images.

The problem of this evaluation is addressed in the measuring of the goodness of the images. For this, we used the analysis of the curves of $\mathbf{C_f}$, obtained of the experiment, versus $\mathbf{B_h}$ and $\mathbf{C_h}$ which is displayed in the Fig. 3. And in the same way, the curves of $\mathbf{B_{mf}}$ versus $\mathbf{B_h}$ and $\mathbf{C_h}$, which is displayed in the Fig. 4. First, the characteristic equations were determined. These equations describe the behaviour of $\mathbf{C_f}$ and $\mathbf{B_{mf}}$.

Then we looked for critical points on the curves and evaluated their corresponding histograms. Identifying the dangerous zones where the images turn into saturated, or when they have a low contrast. It was necessary a heuristic comparison of the characteristics of the histogram, such as number of peaks, their amplitude and location and so on, to define the behaviour of these features of the histograms when the parameters of the digitalization are changed. For this evaluation, we take an approach analogous to that used by Weszka and Rosenfield in [4] to solve a similar problem.

Using all the information obtained, we developed an algorithm based on methods of theoretical decision to find the optimal combinations of $\mathbf{C_h}$ and $\mathbf{B_h}$, to provide good quality images. Using as input parameters $\mathbf{B_h}$, $\mathbf{C_h}$ and extracting the features of the histogram of given image, the algorithm finds

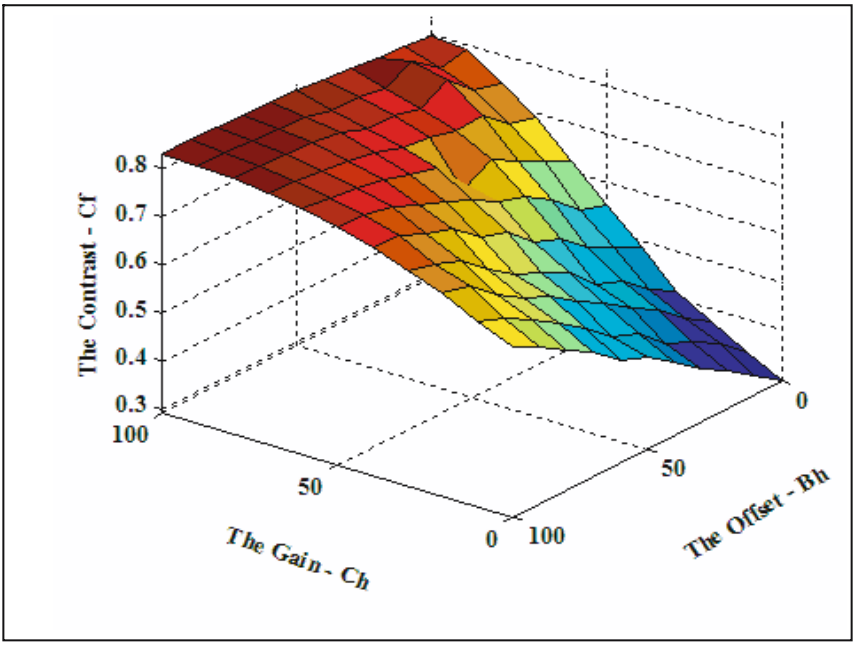


Fig. 3. C_f versus the parameters of digitalization B_h and C_h

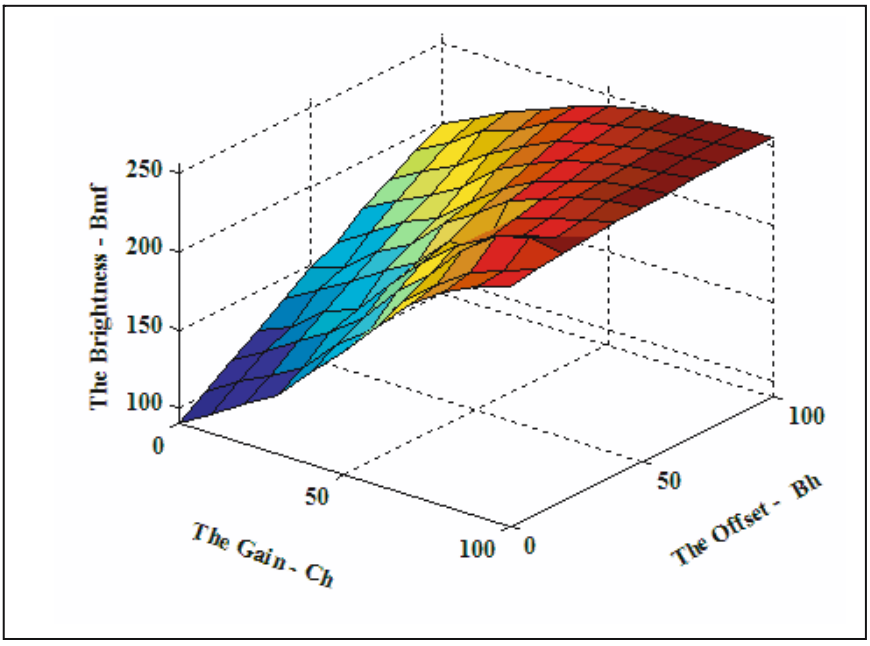


Fig. 4. B_{mf} versus the parameters of digitalization B_h and C_h

the optimal features of the histogram which represent the best contrast and quality that can be obtained for this image.

For the next procedures to extract the interesting information from the environment, we employ basically the operations used by [5]. And also we included an operation for cleaning the perturbation of the frames, such as shadows or noise generated by the illumination conditions. The algorithm developed for this operation is based in the concepts given by Wu and Manmatha in [7] for cleaning up images of documents. But in our case, we must add some criterion, which enables the system to identify the sort of noise, according to the direction of the trail of these perturbations.

In this way, we must ensure that the system is not eliminating any part of the objects or removing an incorrect area. The Fig. 5 shows two images of the

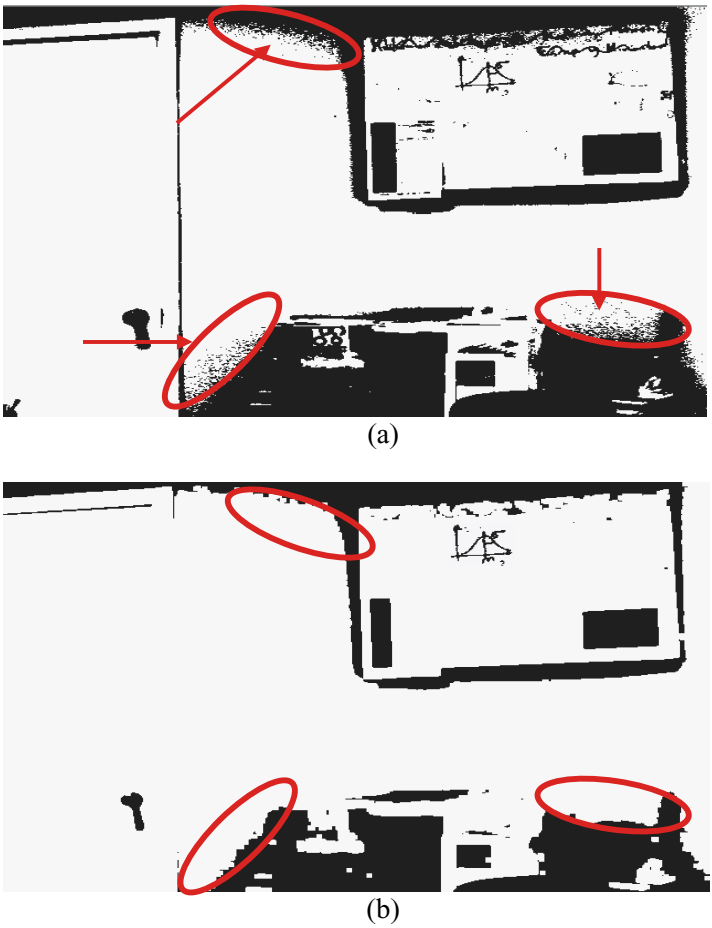


Fig. 5. (**a**) original processed image and (**b**) processed image with cleaning up operation

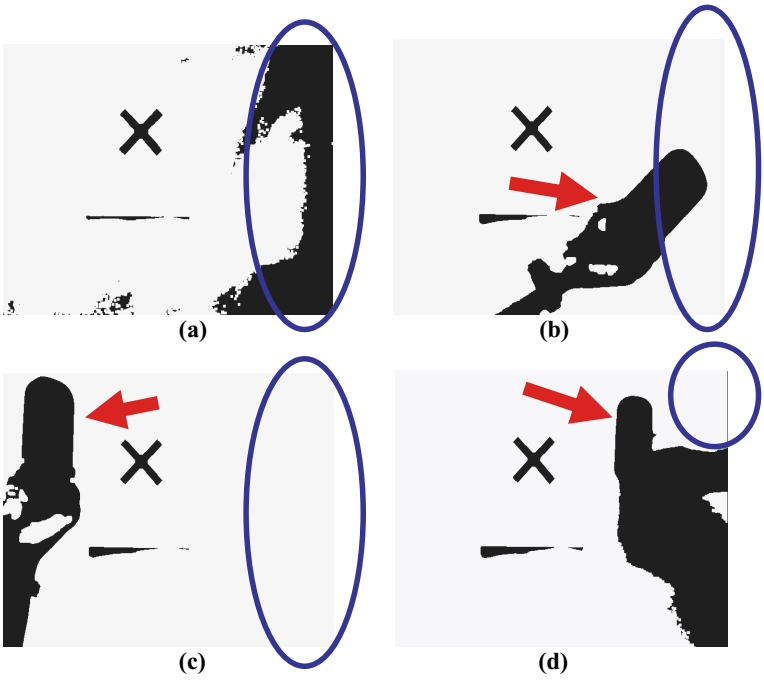


Fig. 6. Images processed using algorithm to detection of new object

same scene. The first is the image processed without applying the cleaning operation and the second is captured when the clean up algorithm is applied. The difference between the two processed images can be easily observed. The last has a clear image, the shape of the objects is not lost and the noise is not interfering with the scene.

And finally, using all the information provided by the developed operations, we create a function which is able to identify when a new moving object is entering in the scene, providing to the system the main features of this new object. In some way, the system separates the new object from the background to control it.

The images of the Fig. 6 show a sequence of frames taken when a new object appears in the scene of the robot. The system detects the change in the scene and as it has the digitalization controller and the cleaning function, it can determine that the origin of the perturbation is a new object. And avoiding the ambiguities, the system separates this object from the background.

In other words the system reacts when a new object is passing over the scene. The image 6.a is the initial frame, showing some characteristic objects of the environment being processed. In the next images the system separates the new object from the background. In this way only the new object and the target of tracking are displayed, avoiding the parts of the background. By means of that, the control and vigilance will be done correctly.

4 Experiments

As we mentioned before, this tool was developed with the objective of being added to a mobile robot performing activities related to vigilance and security control. For this reason, one of the experiments performed was to simulate a situation when the robot must look after a target which is in a human environment, where people are moving and passing over the scene.

In this way the robot must control the position of the target, even in the case in which an external object is covering this target. And also the robot must use the information, given by the vision system, to take an intelligent decision about the security of the target, its own security, and alert about possible risks or evade the trajectories of the moving objects.

The above sequence shows a situation, when a new object of similar dimensions to the target's is passing over the scene (Fig. 7.b), and at some time is covering the visibility of the target to look after, represented by a cross.

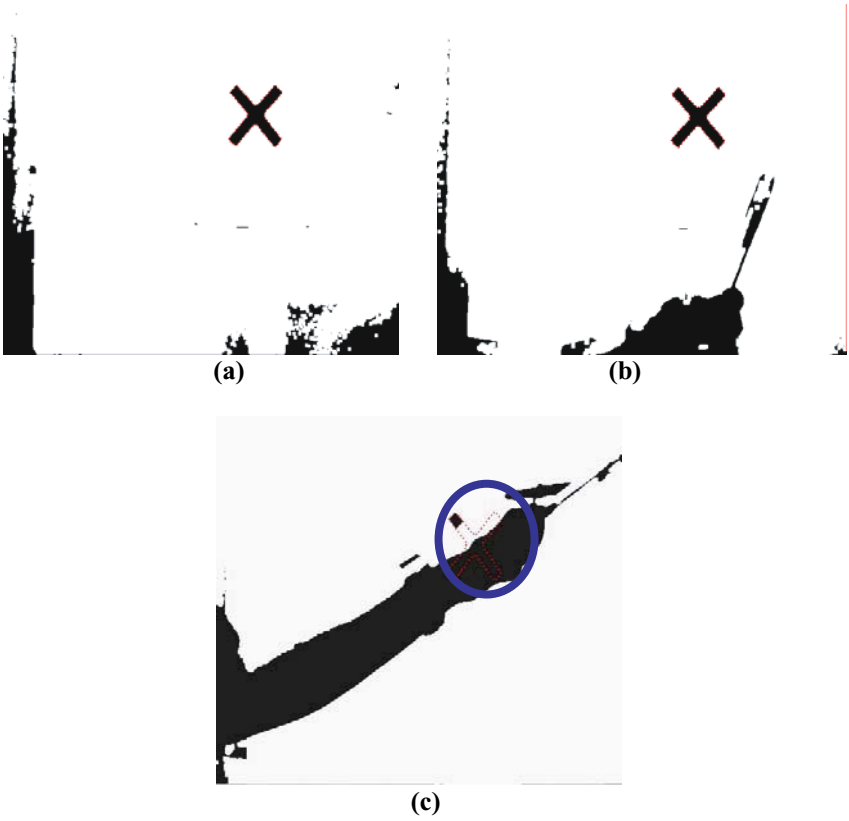
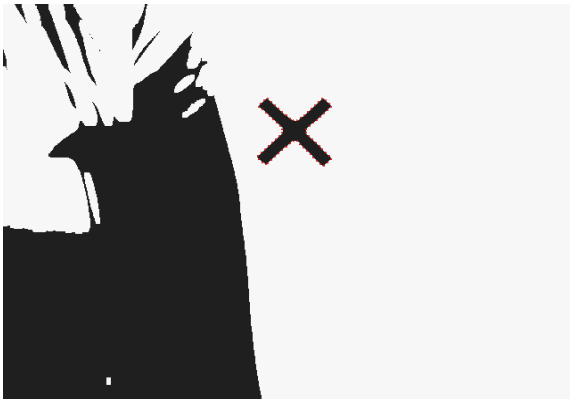


Fig. 7. Sequences # 1 of images controlling a target



(a)

(b)

Fig. 8. Sequences #2 of images controlling a target

But, as shown in Fig. 7.c the system can remember the position of the target and separate the new object from the background.

In the sequence shown in Fig. 8 the object passing over the scene has bigger dimensions and the system is also controlling the target, separating the object from the background.

5 Conclusions

In this paper we present a visual tool to improve the dependability of mobile robots. This is obtained by means of several operations, which incorporate robustness and safety into the system, controlling the perturbations over the robot's scene. This point is very important when the robot is performing

its functions in unpredictable environments, because the conditions are constantly changing. In this way the system is avoiding the ambiguities in the information. And finally the visual tool provides to the robot practical information to help it to take an intelligent decision, including security and surveillance.

On the other hand, this tool is flexible and adaptive. For this reason, the tool can be adapted to the specific functions of the robots.

References

1. Laprice J.C. (1992), Dependability: basic concepts and Terminology, Springer–Verlang.
2. Avizieniz A., Laprice J.C., Randell B. (2003), Fundamental concepts of dependability, UCLA CSD Report no. 010028, LAAS Report no. 01–145, Newcastle University Report no. CS-TR–739.
3. Gonzalez J. (1999), Visión por computador, ITP-Parainfo Chap. 2, pp. 20–37.
4. Weszka L., Rosenfield A. (1978), Threshold evaluation techniques, IEEE Trans. on Systems, Man and Cybernetics, 8(8): 622–629.
5. Salinas C., Pedraza L., Armada M. (2003), Dynamics and Predictive Visual Algorithms for Service Robots, 13th Symposium of Measurement and Control in Robotics, pp. 265–270.
6. Otsu N. (1979), A Threshold Selection Method from Gray-Level Histogram, IEEE Trans. on Systems, Man and Cybernetics, 9(1): 62–66.
7. Wu V., Manmatha R., (1997), Document image clean-up and Binarization, Sup. By National Science Foundation, Library of Congress and Department of Commerce.

Part V

Efficiency and Actuation

Toward Springy Robot Walk Using Strand-Muscle Actuators

Masakazu Suzuki[1] and Azumi Ichikawa[2]

[1] Tokai University, Japan
suzuki@keyaki.cc.u-tokai.ac.jp
[2] Tokai University, Japan

1 Introduction

As robots are requested to perform more and more complex tasks such as rescue activity in more and more practical environment such as rough terrain, it is desired to develope actuators which can flexibly adapt to task environment. The conventional joint mechanism makes the robots structurally complex, then heavy and large, and expensive. In addition it is difficult for them to control joint stiffness in order to handle soft objects or contact human body. Stiffness control by software servomechanism based on the reaction force from the object measured after contact has been vigorously investigated. But the time lag of servomechanism results in undesirable response. On the other hand mechanical stiffness control causes no time lag because the stiffness is determined before the contact with work objects. It is desired to develope small and low cost actuators which can flexibly adapt to task environment, like animate beings. Vertebrates joints easily realize joint angle control and joint stiffness control simultaneously by their antagonistic muscles. A wide variety of artificial muscles are being vigorously investigated [1]. Among them McKibben artificial muscle is a wellknown pneumatic actuator modeled on vertebrates' joints and there are many applications including energy efficient low power joint [2].

We have developed another type of vertebrate-joint-like actuator, Strand-muscle actuator [3] that can adapt to various use and environmental changes. The strand-muscle actuator, having nonlinear elastic characteristics, easily realize angle/stiffness control by antagonistic muscles. In addition the actuator is expected to realize multi-D.O.F complex and flexible motions with simple mechanism. The joint angle/stiffness control has been investigated in [4], and legged walking robots [5] as well as a multi-D.O.F. human-shoulder-like joint, 5 fingered hand have already been developed (Fig. 1).

Fig. 1. 5-fingered hand (*left*), hexapod walking robot, SMAR-II (*center*) and shoulder (*right*) using strand-muscle actuators

The authors are researching an evolutionary behavior learning method, Intelligent Composite Motion Control [6], and applied it to legged robot soccer by a tendon-driven hexapod robot [7]. For more dexterous behaviors, quick actions such as fast walk and rapid turn are desired. The purpose of this research is thus realization of springy walk, i.e., quick and energy-efficient rhythmical walk by a 6-legged Strand-Muscle-Actuator-based Robot, SMAR, developed using strand-muscle actuators.

In the following the strand-muscle actuator is first outlined, where basic actuator characteristics, control scheme, and the experimental result of joint angle/stiffness control are presented. Next SMAR-III is introduced and the experimental result of walking is presented. Finally a method for springy walk by actuator drive pattern optimization is proposed and the prospect of fast and energy efficient walk is given.

2 Strand-Muscle Actuator

A Strand-Muscle Actuator (SMA) has very simple mechanism. It is composed of a motor and a strand-muscle that consists of two or more muscle fibers as shown in Fig. 2. The motor side end is referred to as the actuator's driving end, and the other side is the effected end. At present some kinds of fishing lines are used as suitable muscle fiber material [4].

The drive principle is as follows. Let L be the natural muscle length (Fig. 2(a)). Twisting and straining the fiber strand by the motor, the muscle contracts to ℓ. The contraction ξ is the sum of contraction by the twist $\xi_\phi > 0$ and extension by the tension $\xi_T < 0$, i.e., $\xi = \xi_\phi + \xi_T$. The joint rotation is then realized by antagonistically mounted two actuators (Fig. 3).

With the antagonistic muscles the joint stiffness control is easy as well as joint angle control. In addition to simple structure the muscle itself functions as a speed reducer and a stiffness regulating mechanism. Hence downsizing and weight saving are easily achieved. A multi-D.O.F. joint with high fault tolerance is realized by equipping several redundant actuators in parallel. In

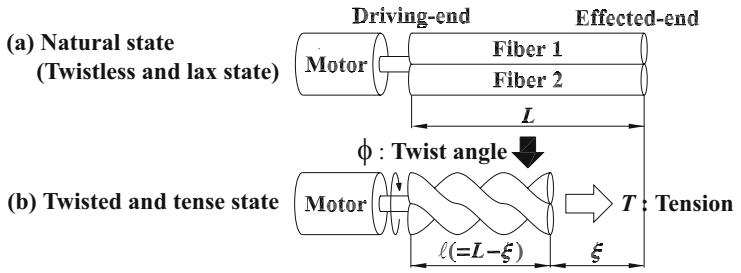


Fig. 2. Strand-muscle actuator with two muscle fibers

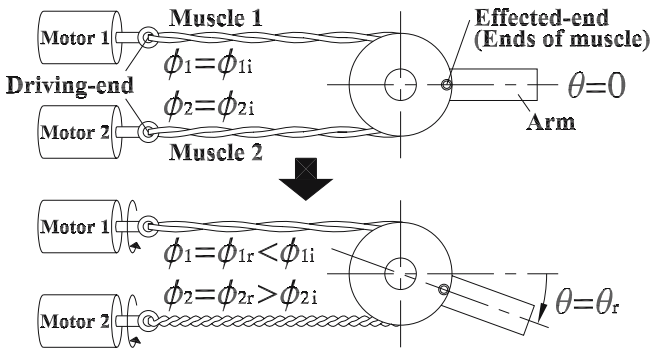


Fig. 3. A robotic joint actuated by antagonistic Strand-muscle-actuators

addition it is possible to select and/or increase muscles according to the actual use because the muscles are easy to equip and replace. This leads to another good feature: the joint characteristics can be easily improved, i.e., variable structure, or moreover, extendable joint mechanism might be possible.

3 Control of Strand-Muscle Actuators

As mentioned in Sect. 2 the SMA can realize flexible and complex motions. On the other hand the mathematical model of the actuator is difficult to establish because of various factors to consider such as fiber bending moment, fiber surface friction and so on. When redundant actuators are installed on a multi-D.O.F. joint, and motions to realize become complex, modeling is more difficult. It is necessary to utilize some additional learning technique, though SMA is basically controlled based on the control equation derived from geometrical analysis with preliminary experiments.

The basic actuator characteristics $\xi(t, T)$ is obtained by motion practice, where t is the actuator drive time from the twistless state under constant muscle tension T. Note that ξ depends on the natural length L and the motor drive voltage V_c, too. The experimental result for the case of $L = 100\,[\mathrm{mm}]$ is

shown in Fig. 4. The initial constraction $\xi(0,T) < 0$ is caused by the preload tention T. The diagram says that twisting the muscle moves the effected end under constant tension, while it increases the tension when the effected end is fixed.

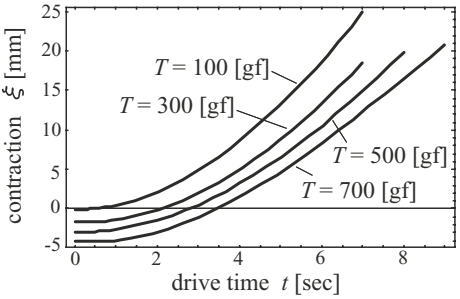


Fig. 4. Experimental result of basic actuator characteristics

Consider a joint in Fig. 3 with two SMAs of the same material/length muscles. In this case joint angle remains $\theta = 0$ when $\xi_1 = \xi_2$, i.e., $\phi_1 = \phi_2$ even if the muscles are twisted and tense (Fig. 3 Upper). When $\phi_1 \neq \phi_2$ contractions ξ_1, ξ_2 are different, and the joint rotates with the angle θ determined by the geometric relation

$$\theta = \frac{\xi_1}{R} - \bar{\theta} = -\frac{\xi_2}{R} + \bar{\theta} \tag{1}$$

where R is the pulley radius, $\pm\bar{\theta}$ are the upper/lower bound of the operation ranges that is realized when muscle 1 or 2 is in natural state.

A SMA-based-joint is driven based on the basic characteristics $\xi(t,T)$. Fig. 5(left) shows the relation between the drive time t and (imaginary) joint

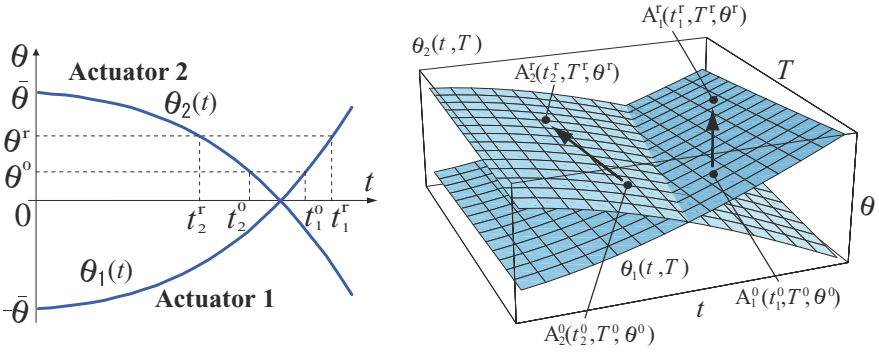


Fig. 5. Actuator state curves for joint angle control under constant tension (*left*) and state sufaces for angle/stifness control (*right*)

angles θ_1, θ_2 when the muscles are separately set so that joint angles are $\theta_1 = -\bar{\theta}, \theta_2 = \bar{\theta}$ without muscle twist under constant tension T. When actuator 1 is driven from the initial state with keeping the tension T, the joint angle θ_1 changes along the curve $\theta_1(t)$. Such is the case also for the curve $\theta_2(t)$. Actually, only states with $\theta = \theta_1(t_1) = \theta_2(t_2)$ are possible since the muscles are antagonistically mounted. Suppose that $\theta = \theta^0$ in initial state. In order to realize the joint angle θ^{r} the actuators are then driven for $t_1^{\mathrm{r}} - t_1^0$ and $t_2^{\mathrm{r}} - t_2^0$, respectively, where t_m^0, t_m^{r} are obtained as $t_m^0 = \theta_m^{-1}(\theta^0), t_m^{\mathrm{r}} = \theta_m^{-1}(\theta^{\mathrm{r}}), m = 1, 2$. ($t_m < 0$ means drive in reverse direction for $|t_m|$).

Extending the curved lines in Fig. 5 (left) in terms of T, the actuator state surfaces in Fig. 5 (right) is obtained. At the points A_1^0, A_2^0 the muscles realize joint angle θ^0 under the tension T^0. In order to realize a desired state $T = T^{\mathrm{r}}, \theta = \theta^{\mathrm{r}}$ represented at $\mathrm{A}_1^{\mathrm{r}}$ and $\mathrm{A}_2^{\mathrm{r}}$, the necessary drive times t_1 and t_2 are obtained from the target joint angle and muscle tension by the joint control equation:

$$t_m = \theta_m^{-1}(\theta^{\mathrm{r}}, T^{\mathrm{r}}) - \theta_m^{-1}(\theta^0, T^0), \ m = 1, 2 \tag{2}$$

Practically however, the geometric equation (1) needs to be compensated according to the experimental control result in order to improve control accuracy. The compensated equation is given as $\theta = \frac{c_1 \xi(t,T)}{R} - c_2\bar{\theta} + c_3$. The control result for $\bar{\theta} = 40\,[\mathrm{deg}]$, $R = 16.5\,[\mathrm{mm}]$ is shown in Fig. 6 (left). The joint angle control with accuracy less than 2 [deg] is possible for a wide range of muscle tension.

The result of stiffness control experiment is given in Fig. 6 (right) to show the joint stiffness characteristics. A moment M is given to the joint which has a specified angle θ under constant tension T to measure the joint stiffness $S = \partial M / \partial \theta$. The result says that the joint stiffness can be controlled by

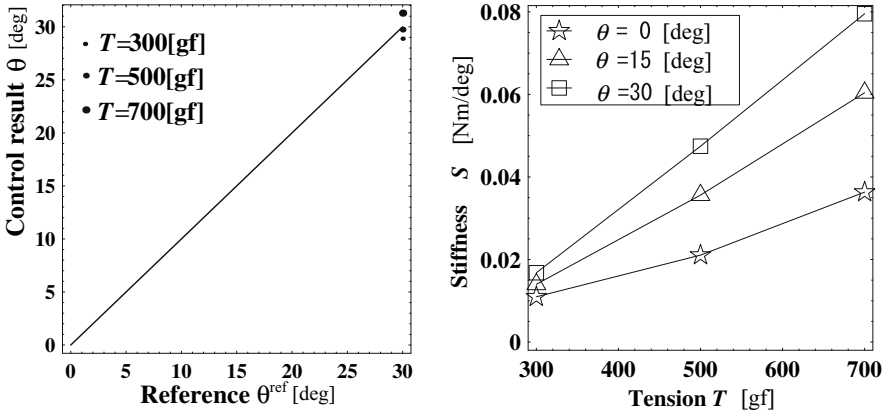


Fig. 6. Experimental result of joint angle control (*left*) and joint stiffness control (*right*)

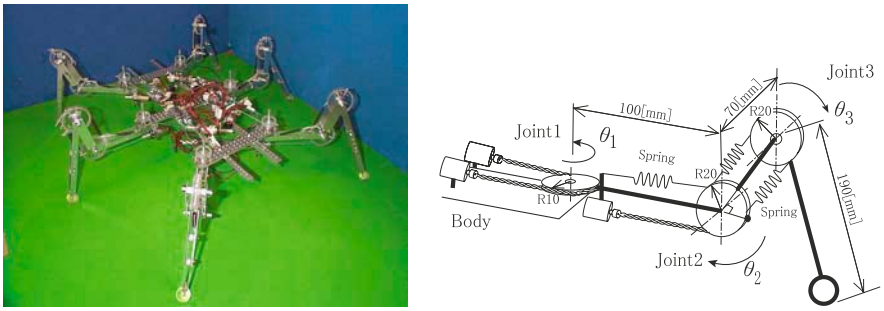


Fig. 7. SMAR-III – Overview and Leg mechanism

muscle tension without joint angle change. The resultant joint angle/stiffness control had enough accuracy, and it was verified to be effective for walking [5].

4 Srand-Muscle-Actuator-based Robot

The SMAR-III (Fig. 7) has 18 joints, 12 of which are driven by 18 SMAs. Each leg has three joints. Joint 1 and 2 are active 1 D.O.F. joints, which are driven by antagonistic and semi-antagonistic SMAs, respectively. Joint 3 is a passive 1 D.O.F. joint which contributes springy walk. Each SMA consists of a DC motor with reduction gear ratio 1/33 and $\phi = 0.5$ [mm] PE-Line (polyester-twine fishing line) as muscle fibers. The size is 445 L×571 W×195 H [mm] in its basic posture, and the weight is 1.47 [kg] (without power supply, computer and cables). Every actuator is driven by a simple on/off control.

Walking is realized according to a predefined on/off actuator drive pattern based on the control equation. Straight tripod walk on a flat terrain has been realized and the captured motion is shown in Fig. 8. The walk velocity is about 75 [mm/sec] that is much faster than 15 [mm/sec] by SMAR-II (Fig. 1 center).

Fig. 8. Tripod gait walking

5 Toward Springy Walk

It is true that the walking by SMAR-III is much faster than conventional, but it is still not so efficient. The simulated motion of θ_1 of the leg 1 (left foreleg) during walking is shown in Fig. 9 (left). Without actuator drive pattern optimization in terms of energy efficiency and walk speed there remains an undesirable vibration after each stride (dotted ovals), i.e., time consuming body swinging without forward move. It makes walk velocity lower and wastes elastic energy.

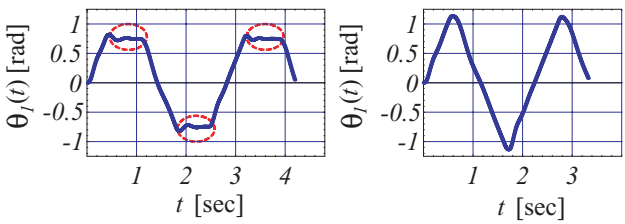


Fig. 9. Motion of joint 1 in nonoptimal and optimal walk

Efficient and rhythmical walking is realized by utilizing the actuators' elastic property and the inertia force by the motion. In other words, the energy efficient walking with a low-duty-ratio intermittent drive is realized by storing the elasic energy obtained from inertia force of each leg by leg-swinging during swing phase and inertia force of the body during support phase. The criterion function for optimal actuator drive is, for example, defines as

$$J(\boldsymbol{a}) = \frac{1}{d_{\mathrm{w}}(t_{\mathrm{w}})} \left(\alpha \int_0^{t_{\mathrm{w}}} \|\boldsymbol{v}(t)\|^2 dt + \beta t_{\mathrm{w}} \right) \tag{3}$$

where $\boldsymbol{a}$ is the parameter vector to specify actuator drive pattern such as walk cycle and on/off switching timing for each actuator, $\boldsymbol{v}(t)$ is motor drive voltage vector, t_{w} is the walk time, $d_{\mathrm{w}}(t_{\mathrm{w}})$ is the walk distance for t_{w}. The criterion is to minimize the energy consumption per unit walk distance (by the 1st term) and to maximize walk velocity (by the 2nd term).

The result of optimizing the drive pattern for joint 1 (Fig. 9 right) says that the walk speed can be nearly doubled with less energy comsumption. The optimization of the joint 1 motion realizes, as it were, rhythmical swinging walk and besides rhythmical hopping walk will be realized by optimizing the joint 2 motion.

6 Concluding Remarks

In this article the Strand-muscle actuator, SMA, and 6-legged SMA-based walking robot, SMAR-III, were introduced, and their mechanisms and control

scheme were outlined. The result of joint angle/stiffness control based on the control equation obtained from preliminary experiments was shown to be effective for SMAR walking. With learning control technique desirable walking was realized without determining many mechanical parameters. It was shown that springy walk based on the actuator drive pattern optimization technique will be possible.

References

1. Proc. 2nd Conf. on Artificial Muscles, Ikeda, Japan, 2004
2. R.Q. van der Linde: Design, analysis, and control of a low power joint for walking robots, by phasic activation of McKibben muscles, IEEE T. Robotics and Automation, Vol. 15, No. 4, pp. 599–604, 1999
3. M. Suzuki, H. Akiba, A. Ishizaka: Strand-muscle robotic joint actuators (in Japanese), Proc. 15th RSJ Annual Conf., pp. 1057–1058, 1997
4. A. Ishizaka, M. Suzuki: On a robotic joint actuated by antagonistic strand muscles (in Japanese), Micromechatronics, Vol. 43, No. 2, pp. 56–61, 1999
5. A. Nakano, M. Suzuki, A. Ishizaka: Development and learning control of a 6 legged robot actuated by Strand-muscle-joints (in Japanese), Proc. 19th RSJ Annual Conf., pp. 361–362, 2001
6. M. Suzuki, A method of robot behavior evolution based on Intelligent Composite Motion Control, Journal of Robotics and Mechatronics, Vol. 12, No. 3, pp. 202–208, 2000
7. M. Suzuki, K.-U. Scholl, R. Dillmann, Complex and dexterous soccer behaviors based on the Intelligent Composite Motion Control, Proc. 4th Int. Conference on Climbing and Walking Robots, Karlsruhe, pp. 443–450, 2001

Actuator Sizes in Bio-Robotic Walking Orthoses

S.C. Gharooni[1], G.S. Virk[1]*, and M.O. Tokhi[2]

[1] University of Leeds, UK
[2] University of Sheffield, UK

Abstract. This paper presents investigations into analysis of energy storage and reductions in size that are possible in designing powered orthoses. A full scale new bipedal humanoid model based on anthropometric data of human is developed with human locomotion pattern to evaluate power and torque profiles at the joints. Current power sources that meet the required specifications are bulky and heavy. These are discussed and the way forward to reduce the size of actuators and sources of power, based on concepts of energy storage and energy conservation is high-lighted in the paper.

1 Introduction

Actuating requirements and actuator sizes pose considerable problems for developing powered orthoses to assist walking in people with disability. Researchers have tried to employ various types of actuator to improve the efficiency of the available orthoses. The actuators are mainly categorised into two branches, biological and mechanical with each category having its own merits. Any actuator used to drive an orthosis should be able to provide the required torque to produce the motion/speed needed; otherwise lack of sufficient power results in dysfunctions in the orthosis. Powering and actuation for bio-robotic walking orthoses are investigated based on existing prototype rehabilitation systems and robotic technologies that have been recently reported. The power requirements and sources for deployment in orthoses have been identified and studied. These include the current technologies and future development of power sources such as fuel cells and nano-combustion engines. The details are published in Virk [8].

*Corresponding author: Prof. G. S. Virk, School of Mechanical Eng, Univ of Leeds, LS2 9JT, UK; Tel: +44 113 343 2156; Fax: +44 113 343 2150; g.s.virk@leeds.ac.uk

From biological actuation, functional electrical stimulation (FES) is an increasingly common technique to restore standing and walking, see Bajd et al. [1]. In this technique paralysed muscles are artificially activated by electrical neuro-muscular stimulation using electrodes either on the skin (surface), through the skin electrodes (percutaneus) or totally implanted ones, see Kobetic et al. [5]. Producing stance phase stability and appropriate swing-phase movement are the two important, often contradictory functions, which challenge FES gait systems. Moreover, two fundamental problems severely restrict the ability of current FES systems to restore gait:

1 Inadequate control of joint torques necessary to produce desired limb trajectories. The non-linear and time varying nature of muscle as an actuator prevents the control of joint torque necessary to produce an acceptable (repeatable and reliable) system output. Moreover, poor movement control results in abnormal gait trajectories with large step-to-step variations.
2 The main problem corresponding to stability maintenance is the rapid muscle fatigue that results from an artificially stimulated muscle contraction. Rapid muscle fatigue limits the standing time and walking distance of the spinal cord injured individual.

To overcome the above restrictions, researchers have explored mechanically powered systems. In these types of systems, a lightweight knee-ankle brace is normally fitted with a DC servo motor and a motor-actuated drum brake coupled to the knee joint with a ball screw so that the device can power or provide controlled damping at the knee. The orthosis is usually modular and designed to slip into a pair of trousers, see Popovic et al. [6]. Such an orthosis system is however bulky because of the large NiCd rechargeable batteries and the heavy motor unit needed to supply the necessary torque; hence it is not practical to use.

2 Requirements

The actuators and power supply will need to meet a range of requirements in order to form part of a successful orthosis. A number of specific requirements (such as peak torque and power) are reported in the engineering specification, see Wright et al. [11]. Moreover, further issues of concern with various technologies, which have been or could be employed in this or similar work include:

- Size and weight
- Efficiency and power to weight ratio
- Specific power and specific energy
- Actuator control and power supply management
- Safety
- Sound and vibration
- Low maintenance.

3 Powering and Actuation

The study on power requirement has revealed that the peak power required at the hip or knee during walking for an average male is 56 W, the mean power is 10 W, the average torque is 72 Nm and the angular speed is 45 degree/sec, see Winter [10]. An average able bodied adult walks for 2.5 hours a day, see Goldsmith et al. [3]. If a 1 kg power source is to run 4 actuators with an efficiency of 60% for one day then it must have specific energy and power greater than 170 Wh/kg and 190 W/kg respectively, see Wright et al. [11].

With advances in battery technologies many types of batteries currently available in the market satisfy these requirements. These include nickel zinc and nickel metal hydride batteries and Li-SO2 and Li-$SOCl_2$ batteries. The power sources that may be utilised include pneumatic and direct fuels using fuel cell or combustion technologies. However, the choice is based on several factors, including type of actuator.

A wide range of actuators are available in the market, however motor technology is leading the market due to its variation, compactness and ease of control and powering. Electric actuators are most common in robotics and can be used for orthoses development, see Virk et al. [7]. Other actuators that can be used include air muscles and cylinders. However, depending on the requirements of specific orthoses a trade off has to be made in the final selection.

Current power sources that meet the required specifications are bulky and heavy. Thus, investigations are carried out to allow reducing the size of actuators and sources of power.

4 Process of Walking in Able Bodied Persons

Human walking is defined as the translation of the centre of body mass through space along a path requiring the least expenditure of energy. As a moving body passes over the supporting leg (stance phase), the other leg swings forward in preparation for the next support phase, see Inman [4]. All the process takes place by controlled motion of three pairs of joints, namely hip, knee and ankle. In normal human walking, progress in the direction of walking is generated by two forces:

1 Hip extension torque $\mathbf{T_h}$ (see Fig. 1) in the stance phase during the swing phase in the contra-lateral leg, causing the trunk to advance.
2 The torque $\mathbf{T_a}$ generated by the plantar flexion of the ankle joint at the beginning of the swing phase.

One of the most profound (and still somewhat controversial) findings of early gait analysis studies was the large contribution of the ankle push-off, which amounts to around 60-70% of the forward propulsive power in natural, normal gait, see Winter [9].

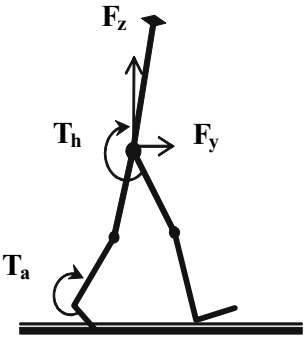


Fig. 1. Joints torques in walking

4.1 Role of the Ankle Joint Activity in Walking

The ankle joint is formed by a highly sophisticated system of bones, muscles, tendons, and ligaments. In quantitative gait analysis, the ankle has been shown to produce substantially more work during gait than any other joint in the lower limb, see Czerniecki et al. [2]. The predominant motion of the ankle joint in walking gait is in the sagittal plane, and the majority of gait analysis techniques developed have focused on that plane of motion.

4.2 Energy Flow at Ankle Joint

The ankle joint has two main functions during walking, firstly it absorbs energy at the beginning of stance phase and secondly it produces push-off or propulsion forces. Propulsion is generated from the reaction between the foot and ground in the late standing phase. Motion and power have some variations in the ankle joint during the standing phase. During the first period the muscles contract and the direction of reaction force exerted by the ground coincides with the direction of the relative motion. The muscular power during that period is defined to be negative. During the second period of the standing phase, the muscle is extended, the reaction force is in the opposite direction to the relative motion, and the power is defined as positive. The ankle joint muscles produce an average of 540 percent more work than they absorb during gait, see Winter [9]. This active generation of power is critical to the production of natural gait. Effective replacement of this power generation is one of the major barriers to total gait replication with an orthotic system.

4.3 Ankle Joint in Orthotic Gait

The ankle joint torque moment cannot be generated in orthosis walking due to the immobilisation of the ankle in the neutral position. Hip extension in orthotic gait only allows the production of torque in an RGO hybrid orthosis

by flexion of the contralateral hip through a mechanical link. Therefore, in paraplegic walking with an orthosis, both the hip extension and propulsive force of the trunk are achieved by using the crutches and arms. By pushing against the crutches, the centre of body rotates over the heel and the toe of the stance leg until the other leg's heel reaches the ground. The rotation of the centre of body mass over the heel moves the trunk forward.

5 Biped Model

A realistic full-scale bipedal humanoid model was designed based on anthropometric data of human with human locomotion pattern to evaluate power and torque profiles at the joints. Standard measurement of human body height and weight has been specified in Winter [10]. The model (Fig. 2), based on simple geometrical shapes, was created within the Visual Nastran (VN) software environment in order to enable its motion control from VN as well as from Matlab/Simulink.

Fig. 2. Humanoid biped model

The control strategy for the humanoid walking cycle was developed, in order to control trunk balance as well as leg swing. The trunk balance is controlled with PD control in closed-loop with the stance hip torque, whereas the leg swing is controlled with open-loop torques defined in the Matlab workspace. The biped humanoid can walk from an initial to a final standing position at constant stride.

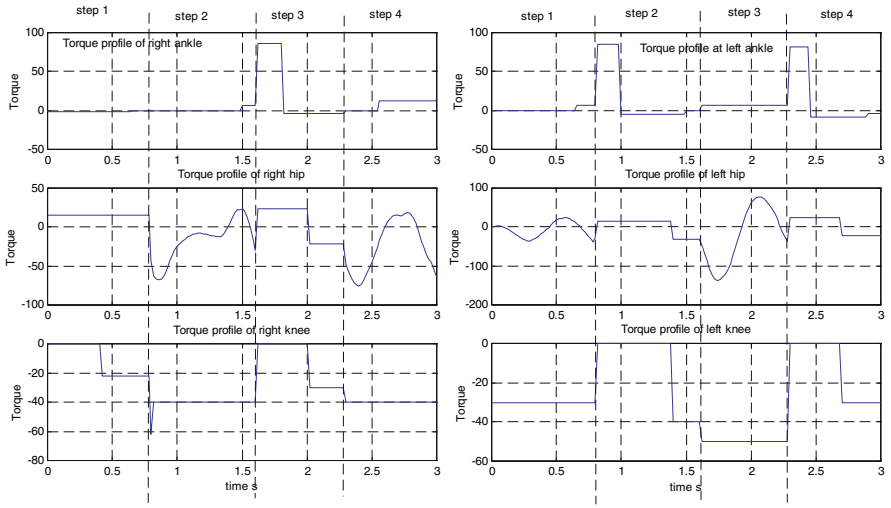


Fig. 3. Torque profile in biped for *right* and *left* leg

6 Torque Profile in Biped Joints

In order to have an accurate estimate of torque required in each joint of synthesized humanoid gait, a model of humanoid biped developed and six step walking were simulated. Simulations (see Fig. 3) reveal the torque profile for each joint and then guide designers to select the motor size required for of the six joints in a biped (hip, knee and ankle for both sides). Note also that large torque is required to keep the torso in an upright position.

At the ankle joint, a very narrow torque is needed at the beginning of stance phase; it lasts for 0.2 sec and has amplitude of 90 Nm.

The size of the motor for each joint should be capable of producing the maximum torque in each joint. For instance, the ankle joint needs a motor size to produce 90 Nm. This is impractical and forces biped designers to consider alternative gait patterns or look for other compact actuators. This research is accordingly considering alternative mechanisms based on energy storage and energy conservation strategies to allow reduction of actuators.

7 Conclusions

An investigation into energy analysis at joints in biped humanoid robots has been carried out. Investigations have accordingly provided essential information on actuator sizes required at the various joints. Among these it has been noted that the largest kicking force is required at the ankle joint for which based on the required power and energy profiles large size actuators will be required at the ankle joints. This, however, will not be practical, specifically

for adoption in paraplegic walking. Accordingly, alternative approaches to allow reduction in size of actuators, need to be sought. A method with such potential currently investigated by the researchers is based on energy storage devices, findings of which will be reported in the future.

Acknowledgments

The work presented was supported by EPSRC under Research Grant GR/R10-981/02.

References

1. Bajd, T., Kralj, A., Stefancic, M. and Lavrac, N. (1999) Use of functional electrical stimulation in the lower extremities of incomplete spinal cord injured patients. In *Artif Organs*, Vol. 23(5), pp. 403–9.
2. Czerniecki, J.M., Gitter, A. and Munro, C. (1991) Joint moment and muscle power output characteristics of below knee amputees during running: the influence of energy storing prosthetic feet. In *J Biomech*, Vol. 24(1), pp. 63–75.
3. Goldsmith, A.A., Dowson, D., Wroblewski, B.M., Siney, P.D., Fleming, P.A., Lane, J.M., Stone, M.H. and Walker, R. (2001) Comparative study of the activity of total hip arthroplasty patients and normal subjects. In *J Arthroplasty*, Vol. 16(5), pp. 613–9.
4. Inman, V.T. (1966) Human locomotion. In *Canadian Medical Association Journal*, Vol. 94, pp. 1047–54.
5. Kobetic, R., Triolo, R.J., Uhlir, J.P., Bieri, C., Wibowo, M., Polando, G., Marsolais, E.B., Davis, J. A., Jr. and Ferguson, K. A. (1999) Implanted functional electrical stimulation system for mobility in paraplegia: a follow-up case report. In *IEEE Trans Rehabil Eng*, Vol. 7(4), pp. 390–8.
6. Popovic, D. and Schwinlich, L. (1987) Hybrid powered orthoses. In *Advances in external control of human ex-tremities IX*, pp. 95–104.
7. Virk, G.S., Bag, S.K., Wright, P.J., Gharooni, S.C., Smith, S.A., Sheth, R., Tylor, R.I., Bradshaw, S., Tokhi, M.O., Jamil, F., Swain, I.D., Chappell, P.H. and Allen, R. (2004) Powering and Actuation for Bio-robotic Walking Orthoses. In *35th Int. Conf. ISR*, Paris, France.
8. Virk, G.S. (2000). *Bio-Robotic Walking Orthoses – Phase 1, Appendix C.* S13/06, EPSRC Research Proposal.
9. Winter, D.A. (1983) Energy generation and absorption at the ankle and knee during fast, natural, and slow cadences. In *Clin Orthop*, Vol. (175), pp. 147–54.
10. Winter, D.A. (1991) *The biomechanics and motor control of human gait: normal, Elderly, and Pathological, 2nd Edition*, Waterloo University Press.
11. Wright, P.J., Virk, G.S., Gharooni, S.C., Smith, S.A., Tylor, R.I., Bradshaw, S., Tokhi, M.O., Jamil, F., Swain I.D., Chappell, P.H. and Allen, R. (2003) Powering and actuation technology for a bio-robotic walking orthosis. In *ISMCR*, pp. 133–38.

The Design and Simulated Performance of an Energy Efficient Hydraulic Legged Robot

Salim Al-Kharusi and David Howard

School of Computing, Science and Engineering, University of Salford, UK
d.howard@salford.ac.uk

1 Introduction

Energy efficiency is a major issue in the design of any vehicle. However, this is particularly the case for legged robots because of the geometric work problem [4] and [1], which occurs because the legs must support the machine's weight as well as providing propulsion. This means that, even when moving horizontally, some leg joint actuators will be doing negative work (braking), while others are doing positive work (propelling). Therefore, to make full use of the energy supply, the negative work done by joint actuators should be stored and used when positive work is required. This means that there should be some kind of storage system that absorbs and gives back energy. A suitable system for an energy efficient legged robot is a hydraulic system because it has the ability to store the energy in the form of pressurized gas, in an accumulator, which can then supply high pressure oil to another part of the system, e.g. hydraulic motors that need this energy.

The hydraulic system proposed here has two advantages over many previous actuation systems (hydraulic, pneumatic or electric) used to drive legged robots. One of these advantages is that the negative work used in braking a leg joint is not wasted energy. The negative work is converted to hydraulic flow, which can be used by another joint in the system. The other advantage is that the net flow from all hydraulic units is stored in an accumulator and can be used when needed, e.g. when accelerating. The research reported here aimed to establish the performance of the proposed hydraulic system.

Although there have been other legged robots that have used hydraulic actuation systems, they have tended to be overly complex and, probably as a direct result of this, not particularly efficient [3]. The Adaptive Suspension Vehicle (ASV) is the most well known hydraulic walker and probably the most expensive. However, the design is very complex, combining pantograph leg mechanisms, an energy storage flywheel, a complicated mechanical power transmission system, and variable displacement hydraulic pumps driving linear actuators – one pump for each actuator [2, 5]. In the proposed design,

the power transmission system is entirely hydraulic and supplied by a single fixed displacement pump. The leg joints are driven directly by variable displacement motor-pump units.

2 The Proposed Hydraulic Design

Figure 1 shows the new hydraulic design, which consists of motor-pump (m-p) units, which drive the leg joints, an accumulator, a fixed displacement pump, a relief valve and a tank. The positive direction of the flows in the system is indicated by the arrows.

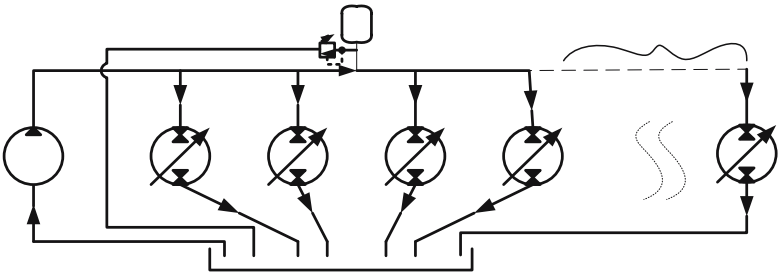


Fig. 1. The proposed hydraulic design

The m-p units are variable displacement axial piston devices, which are used to drive the joints of the legged robot because they have the ability to work as motors or pumps, and their volumetric displacement can be varied. All of the m-p units are connected in parallel between the accumulator and the tank. When the work done by a leg joint is positive, the corresponding m-p unit will consume energy in the form of hydraulic flow taken from the accumulator to the tank, in which case, the unit is working as a motor. On the other hand, when the joint work is negative, the m-p unit will produce energy in the form of hydraulic flow taken from the tank and stored in the accumulator, in which case, the unit is working as a pump.

In the proposed design, the accumulator stores the high pressure oil produced by m-p units doing negative work, which can be used later when there is a demand for positive work. The accumulator pressure is maintained, between lower and upper limits, by a fixed displacement axial piston pump, which is driven by the machine's prime-mover (e.g. Diesel engine). However, if the pressure rises above the upper limit, reaching the maximum allowed value, the relief valve will open to reduce the pressure. The fluid released by the relief valve returns to the tank, which is at atmospheric pressure. When the legged robot is operating on a horizontal surface or moving up an incline, there is no reason for the relief valve to operate. However, when moving down an incline or decelerating, the accumulator pressure may reach its maximum

value, after which energy is converted to heat by the relief valve rather than stored.

3 Simulation Results

A MATLAB simulation of a hexapod walker has been implemented including the proposed hydraulic drive system. Figure 2 indicates the scope of the simulation; the three supporting legs are modelled as they move the machine body parallel to the support surface.

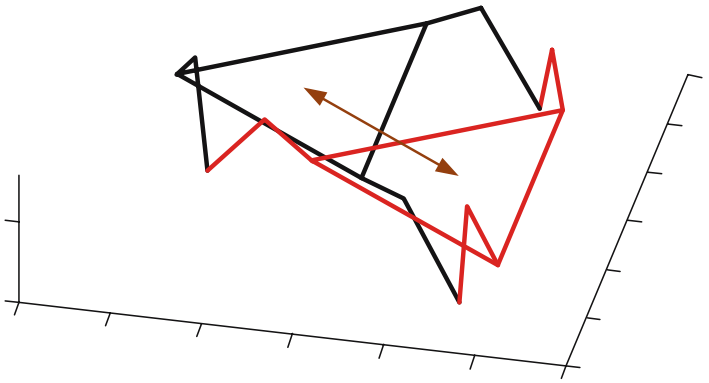


Fig. 2. Simulation of supporting legs for a hexapod robot

The simulation of the motor-pump (m-p) units, which drive the leg joints, includes a detailed model of the hydraulic flow losses due to internal leakage and oil compressibility. In this way, realistic energy consumption data has been obtained and used to assess the efficiency of the system.

The majority of previous legged robot prototypes did not store and return the negative work produced by their actuation systems. Therefore, comparison with a *"no storage"* model is a reasonable way to quantify the potential efficiency of the new hydraulic design as compared with previous designs. In the *"no storage"* model, the actuator efficiencies are assumed to be 100%, so that the comparison with previous designs is more than fair. Mechanical (friction) losses in the proposed design have been neglected as they are likely to be similar to those in previous designs, which are represented by the *"no storage"* model, which also neglects mechanical losses. However, the hydraulic flow losses have been included as these are specific to this type of hydraulic design.

Figure 3 shows some typical simulation results which demonstrate the effect of robot velocity on system efficiency, with the machine moving up a 16° incline. It shows that the efficiency increases as the speed increases. This

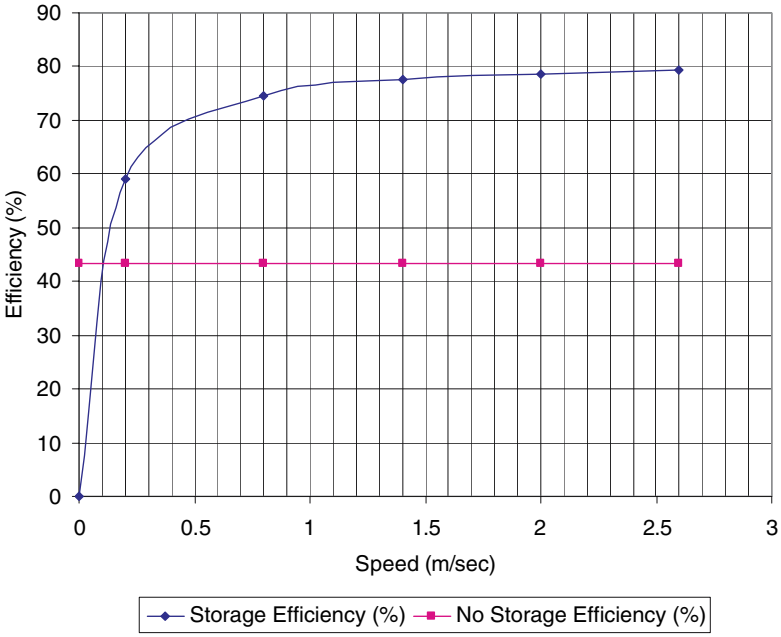


Fig. 3. Efficiency versus speed

is because the hydraulic flow rates increase with speed and, therefore, the flow losses become a smaller proportion of the total flow and the efficiency improves. The efficiency asymptotically approaches a value of around 80%, at which point the compressibility losses dominate and, as these vary in direct proportion to the speed, efficiency no longer varies with speed. The efficiency falls rapidly as the speed approaches zero because of the flow losses caused by internal leakage. Conversely, because the *"no storage"* model assumes 100% efficient actuators, its efficiency is independent of speed.

The *"no storage"* efficiency is around 43% as compared with efficiencies between 70 and 80% for the new hydraulic design (at realistic speeds and power consumptions). This demonstrates the overriding importance of energy storage and return, which has often been neglected in previous legged robot designs.

Figure 4 shows that efficiency increases as the slope increases, and that the trend is similar for the *"storage"* (new hydraulic design) and *"no storage"* models. This increasing efficiency is a result of decreasing geometric work as the slope increases, in other words, the actuators are less likely to do negative work (braking). The improvement in efficiency obtained by changing the actuator gear ratios to suit the slope is surprisingly small. Again, the figure shows a large difference between the *"storage"* and the *"no storage"* models.

Figure 5 shows the effect of the crab angle on the efficiency of the machine. The trends are similar for the *"storage"* (new hydraulic design) and

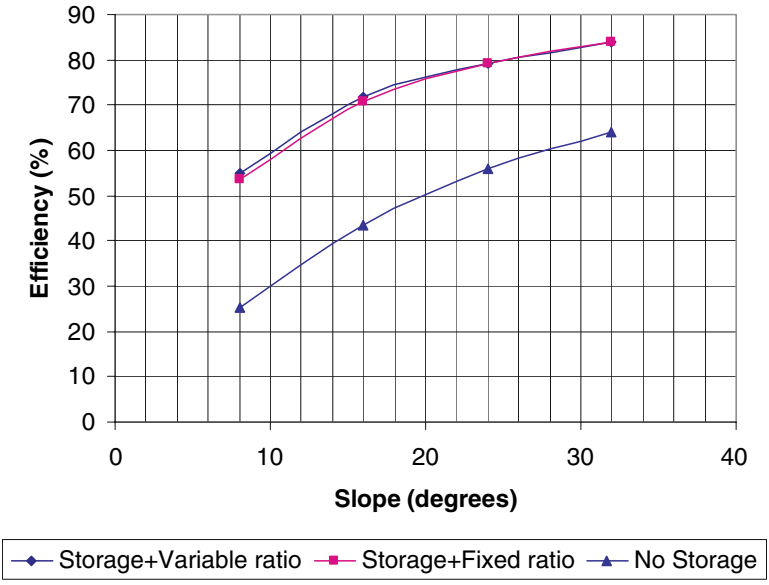


Fig. 4. Efficiency versus slope

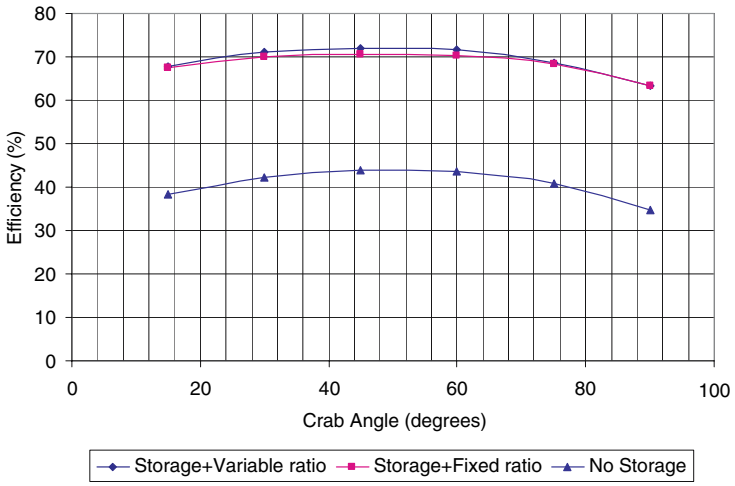


Fig. 5. Efficiency versus crab angle

"no storage" models. The efficiency increases as the angle increases until the angle is between 45 and 60 degrees, then it starts to decrease. The improvement in efficiency obtained by changing the actuator gear ratios to suit the crab angle is surprisingly small. Again, the figure shows a large difference between the *"storage"* and the *"no storage"* models.

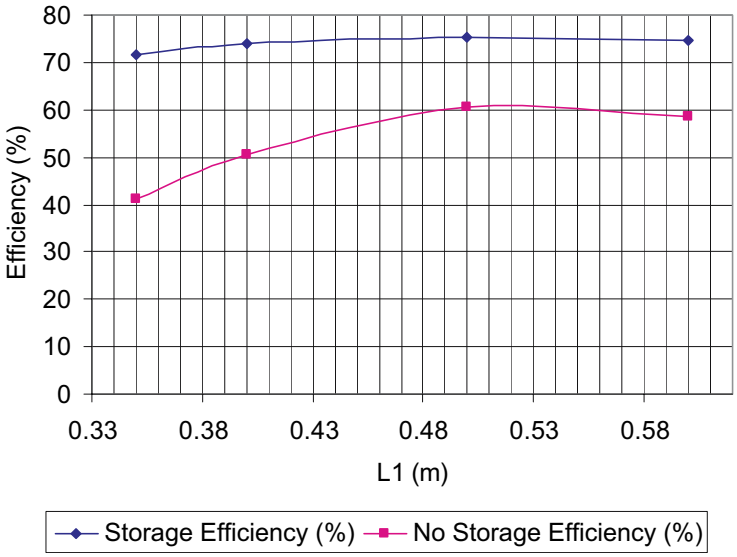


Fig. 6. Efficiency versus leg geometry

Figure 6 shows the effect of leg geometry on efficiency. Because the machine geometry has been normalised with respect to leg length, if the thigh length (L_1) is given, then the shank length is equal to $1-L_1$. Therefore, leg geometry can be represented by just L_1. For each leg design, a different knee actuator gear ratio was found to be necessary, as follows:

L_1	0.35	0.4	0.5	0.6
R_k	95	85	70	50

There are no large differences in efficiency with the *"storage"* model. However, the *"no storage"* model shows efficiency increasing with L_1, until L_1 reaches 0.5, and then decreasing. The conclusion that can be drawn from this is that geometric work is sensitive to leg geometry and, therefore, if braking work is not stored and returned, the efficiency will be sensitive to leg geometry.

4 Conclusions

A new design for an energy efficient hydraulic legged robot has been proposed. An accumulator is used to store and return the negative work done by joint actuators because of the geometric work phenomenon. The majority of previous legged robot prototypes did not store and return the negative work produced by their actuation systems. Therefore, the efficiency of the new design has

been compared with that of a *"no storage"* model, which represents previous designs. The *"no storage"* efficiency is around 43% as compared with efficiencies between 70 and 80% for the new hydraulic design (at realistic speeds and power consumptions). This demonstrates the overriding importance of energy storage and return, which has often been neglected in previous legged robot designs.

References

1. Liu, A., Howard, D. (1999) Leg mechanism designs for CLAWAR machines – a critical review. CLAWAR '99, Portsmouth, September 1999.
2. Pugh, D.R., Ribble, E.A., Vohnout, V.J., Bihari, T.E., Walliser, T.M., Patterson, M.R., Waldron, K.J. (1990) Technical Description of the Adaptive Suspension Vehicle. International Journal of Robotics Research, 9(2), 24–42.
3. Song, S.M., Waldron, K.J. (1989) Machines That Walk: The Adaptive Suspension Vehicle. MIT Press, Cambridge, Mass.
4. Todd, D.J. (1985) Walking Machines, An Introduction to Legged Robots. Kogan Page Ltd.
5. Waldron, K.J., Vohnout, V.J., Pery, A., McGhee, R.B. (1984) Configuration Design of the Adaptive Suspension Vehicle. International Journal of Robotics Research, 3(2), 37–48.

The Modularity of Super Embedded Robotic PC

A. Basile, N. Abbate, C. Guastella, M. Lo Presti, and G. Macina

Automation and Robotics Team, STMicroelectronics, Stradale Primosole 50, 95121 Catania – Italy
adriano.basile@st.com

Abstract. An embedded system has to be considered modular when it is re-usable and implements standards accepted by technical and non-technical viewpoints. Actually, the lack of real standards is the main obstacle for the system manufacturers to create modular objects. The aim of this paper is to present the efforts of the Automation and Robotics Team of STM in realizing a novel embedded prototype, that results modular in own components. In fact, the embedded system, called Super Embedded Robotic PC (SERPC), works on the I^2C bus where the single components are connected or disconnected easily. On this bus could be located a 32-bit microprocessor, a 16-bit microcontroller, an inertial sensor, a CMOS sensor.

1 Introduction

A system is in conformity with the modularity concepts when there is a common framework and the capability to plug or remove components easily. The modularity concept applied to the robotic fields could make the breakthrough to obtain a mass market selling of the robotic products. To introduce the modularity concepts is needed identify the specific requirements of the systems, the design parameters and the design specifications. In this paper the authors would propose a Super Embedded Robotic PC (SERPC), which is an embedded system in a small format where have been integrated many peripherals. SERPC has been conceived by taking into account the requirements of many application fields, starting from the home appliance until to the industrial environment. For example, as regard the navigation purposes is important the environment sensing more than wheel locomotion control. Instead, in a mounting handling machine the motor control is fundamental, because the trajectory on the work-piece must be compensated in real-time. By following these guidelines the peripherals adopted in SERPC are devoted to the actuator control, the visual sensing, the inertial sensing, a wide set of network peripherals to connect with standard devices and, the high and low level computation. Moreover, a series of free digital and analog I/O pins have been considered for further expansions.

SERPC has been designed around one of the most common buses: the I^2C [1]. Its flexibility permits to add peripherals easily and to reach transmission speed of 3.4 Mbits. It works on two wires: one is the serial data line (SDA) and the other one is the serial clock line (SCL), where each device connected to the bus is software addressable by a unique address and by simple master/slave relationships. Therefore this bus is considered a modular connection bus.

The aim of this paper lies in exploring new modular configurations around this bus; the characteristics of the two main processors will be exploited to propose some innovative configurations that fulfill the above mentioned features. Moreover the selected peripherals will be introduced in order to offer the possibility to know SERPC as a fully embedded system.

The present paper will be organized as follows. The next Section shows the main components of the SERPC, starting with the CPUs, which represent the core of our configuration, and concluding with the power boards, which interface the motors. The Section III will observe the modularity concept applied to SERPC system, particular emphasis will give to the application fields. Finally the Section IV concludes the paper summarizing the peculiarity of the presented work.

2 Main Components of SERPC

SERPC is based on the I^2C bus; this has been chosen for its flexibility and its popularity. In fact the I^2C-bus has become a de facto world standard in the last years. The presence of a bus permits to have the basic idea of modularity; on it, most of all components could be plugged or changed following the target specifications. Two components arrange the main part of SERPC: the first one is the STPC [2] chip, it represents a complete 32-bit system-on-chip that could substitute, in some version, a motherboard. The second main component is the ST10 microcontroller [3]. It is a 16-bit micro, already sold in 80 million of pieces.

The STPC exists in four different versions: *Atlas, Consumer II, Elite* and *Vega*. They differ for the embedded capabilities (for further information, please refer to [4]). The STPC Atlas [5] combines a standard 5^{th} generation x86 core with a powerful graphics/video chipset; a series of controllers as the PCI, ISA, USB, EIDE, and the Local Bus. Moreover, it includes also a video input port in CCIR 601/656 mode [6], a TFT interface, and the JTAG and I^2C peripherals. The I^2C interface works on two pins shared with the Direct Data Channel Serial Link (DDC). From the external point of view, the ISA bus, the Local Bus and the PCMCIA interface are multiplexed; this is due to the number of pins available on the package. So the functionalities are selected by the strap options on Reset. Finally, the Video In port stores the video flow from the video pixel decoder or the digital video input.

As regards the ST10, its key features are: the very short instruction time, only 50 ns (@ 40 Mhz); the embedded flash memory; the fast DSP-MAC that

enhances the speed of the math operations; and a set of peripherals as CAN ports, timers, ADCs, PWMs and RTC.

These two components cover completely both the high and low level control of SERPC actions. The low-level control means the managing of the sensors and motors. Meanwhile, the high-level control addresses the trajectory planning, the image elaboration, and others complex elaborations.

On the I^2C bus has been connected also the novel three axial inertial sensor, LIS3L02DS [7], which is able to take the information from the sensing elements and to provide the measured acceleration signals to the I^2C serial interface. Moreover, the STPC could use these bus lines to configure the registers of the VS6502 VGA Color CMOS Image Sensor [8]. It represents a miniature solution to the web-cam, it sends the compressed video stream via USB port. This solution could replace the camera connected to the Video In port of the STPC.

A schematic view of SERPC mainboard could be appreciated in Fig. 1. All the ICs of SERPC are changeable with other components, in order to investigate novel configurations. For example, in further versions the STR7xx [9] processors could replace the STPC, or three ST7MC [10] could replace the ST10. In a more general configuration, other I^2C peripherals could be taken into account: the Global Positioning System (GPS), the ultrasound sensors, the laser scanners, and so on.

The modularity of SERPC lies also on the motor interface stages. Here, several modular boards have been designed to drive various families of motors: DC, Brush Less DC, Stepper motors, and so on. These boards are connected through a unique flat cable to SERPC, where three sockets are present (see Fig. 1). Once the board is connected, a different software library on the ST10 adapts its digital and analog pins to the socket pin-out. In this way more configurations could be adopted. The only restriction regards the third connector that is the only one capable to accept the dual motor board.

These external mini-boards vary on the size of the motor connected to the system and its features. For small-motor applications, the ST10 is directly connected to a DMOS driver as the L6235 [11]. In Fig. 2 is reported a typical layout, where the motor driven is a three phase brushless DC motor. This DMOS driver works with motors up to 50–100 Watts. The L62xx represents a wide family of monolithic motor control ICs, which take full benefit of ST's BCDIIIs diffusion technology. Any member of this family gives an optimized solution to motor control systems with a level of performances and effectiveness low cost. For example, L6208 drivers [12] are used for bipolar stepper motor, and the L6205 drivers are adapt for dual DC motors.

For high power motors a double stage has to be considered (refer to the Fig. 3), the first one is devoted to split the PWM signal on the various low and high sides, these tasks are performed by the interrupt routines of the ST10; the second one starts from the L6386 [13] (high-voltage high and low side driver) and implements the power stage, generally with the IGBT drivers. This high power motor board is divided in two boards when the power engaged

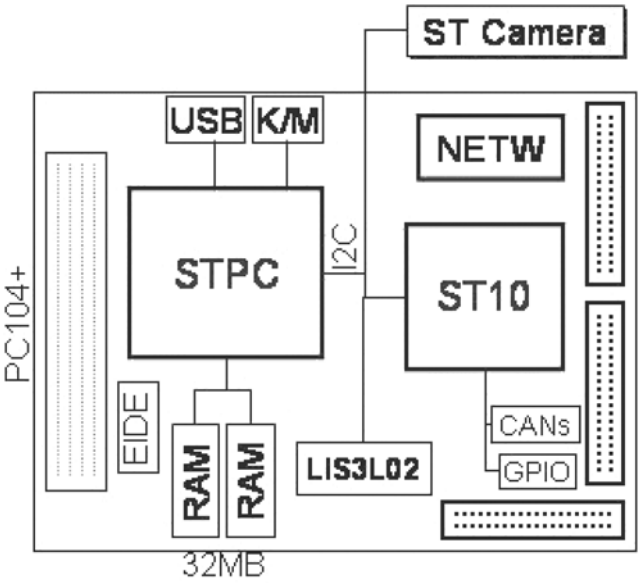


Fig. 1. Schematic view of the SERPC Mainboard

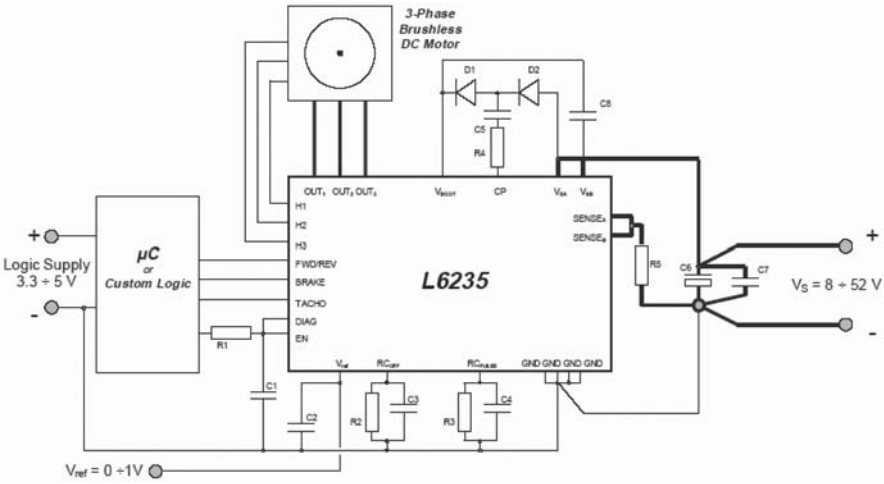


Fig. 2. The connection between the ST10 and the L6235 motor drive

becomes too high. The separate power board composes the power front-end section that allows also supply from AC or DC source. This converter uses a Vertical Intelligent Power Enhanced Regulator (VIPER) [14] that provides with start-up capability, integrated PWM controller and thermal and over-current protection.

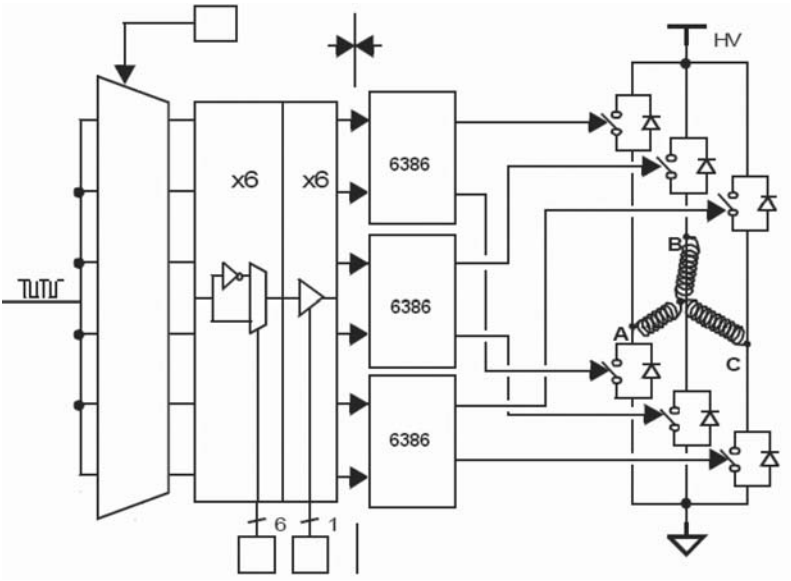


Fig. 3. Schematic view of the High Power Motor Board

3 Overall Performance of SERPC

An idea of the PCB of SERPC could be seen in Fig. 1, where its dimension should be the ETX typical format. Meanwhile, the PC104+ connector guarantees the connectively of SERPC with all the expansion board offered in this format.

The modularity concept of SERPC is applicable also to the PC104+ connector. In fact, this connector is replaceable by all the connector PCI or ISA compatibles. From a technical point of view, the bus modification implies only a rewiring of the connector.

The board mounts 32MBytes of RAM, where the frame buffer occupies up to 4 Mbytes of the physical memory. By using the maximum frame buffer, the graphics resolution reaches the 1280×1024 in 16 Million of colours. The JTAG connection of the STPC is used to program a Programmable Logic Device that guarantee the possibility to create new simple peripherals. The software environment of SERPC is based on Microsoft Windows CE 4.2 .NET, its graphical interface offers an easy way to drive all the on-board peripherals. A further study regards the RT Linux and the QNX operating systems, to implement on these OSs a series of software libraries capable to offer a complete reference design.

It is easy understand that there are many application fields of the SERPC prototype. The simplest one is a mobile roving robot which dispatches mail in a structured environment. It represents a demo system capable to demonstrate the SERPC features and a valid test platform that will stress all the

components at the same time. More complex application could focus on complex navigation algorithm in unstructured environments; on landmark position reckoning; on the odometric sensing validation via GPS; and so on.

4 Conclusions

In this paper, the authors would present a novel modular architecture that is able to respond to the main request of the robotic architecture. It joins the performances of a powerful x86 system to a complete 16 bit microcontroller that is dedicated to drive the motors and manage the low level environment sensing. At these two ICs are added a complete series of peripherals that are able to make the described modular Embedded Robotic system, called SERPC.

The modularity guarantees a high scalability of the whole platform; it will open many new opportunity to different case studies, these ones will give the real feedback to the authors to improve their work. As it is possible to note, in this paper has not been discussed about the many network peripherals that could be connected to the system. This topic is not been faced yet, but its study will be offer the opportunity to interface with all the industrial products, that work already in the factory automation field. One example is given by the two CAN peripherals of the ST10.

Acknowledgement

The authors gratefully acknowledge the Prof. Giovanni Muscato and the Eng. Domenico Longo for their helpful contributions.

References

1. Online page: http://www.semiconductors.philips.com/buses/i2c/
2. Online page: http://www.st.com/stonline/books/ascii/docs/7536.htm
3. Online page: http://www.st.com/stonline/prodpres/dedicate/auto/embedded/-st10.htm
4. Online page: http://www.st.com/stonline/products/selector/107.htm
5. Datasheet of STPC ATLAS – x86 Core PC Compatible System-On-Chip for Terminal, Online Document: http://www.st.com/stonline/books/pdf/docs/7341.pdf
6. R/ITU-R on line: http://www.itu.int
7. Online page: http://www.st.com/stonline/prodpres/dedicate/mems/products/lis3l02.htm
8. Databrief of VGA Color CMOS Image Sensor Module, Online Document: http://www.st.com/stonline/books/pdf/docs/9705.pdf

9. Online page: http://www.st.com/stonline/products/support/micro/arm/-str7.htm
10. Online Document on ST7MC1: http://www.st.com/stonline/books/ascii/docs/-9721.htm
11. Datasheet of DMOS Driver for Three-Phase Brushless DC Motor, Online Document: http://www.st.com/stonline/books/pdf/docs/7618.pdf
12. Datasheet of DMOS Driver for bipolar Stepper motor, Online Document: http://www.st.com/stonline/books/pdf/docs/7514.pdf
13. Data sheet of High-Voltage High and Low Side Driver, Online Document: http://www.st.com/stonline/books/pdf/docs/5653.pdf
14. Online page: http://www.st.com/stonline/bin/sftab.exe?db=rosetta&type=&table=432

Mass Distribution Influence on Power Consumption in Walking Robots

T.A. Guardabrazo and P. Gonzalez de Santos

Industrial Automation Institute – CSIC, Madrid, Spain
t.guardabrazo@iai.csic.es

Abstract. The aim of this paper is to study how the mass distribution of walking robots affects to power consumption and joint torques. First, a theoretical model of the robot's energetics is derived and simulated for appointing the problem. Next, a testbed is used for validating the theoretical model by comparing measurements with simulation results. Finally, the energy performance of the testbed with different mass distributions is evaluated and discussed, and some conclusions are provided.

1 Introduction

During the last two decades, the problem of mass distribution has been widely studied in statically unstable robots as bipeds because it is a key issue for balancing the body and improving the stability of dynamic walking. However, mass distribution is one of the most important factors for optimising power consumption and minimising joint torques not only in bipeds but also in multi-legged robots.

It has been traditionally accepted that a walking robot must be equipped with light legs. It means that electromechanical devices have tended to be as concentrated as possible on the machine's body, increasing its weight. For example, in the walking robot involved in the DYLEMA project [1] all motors are located on the hip so that movement is transmitted to the second and third joints by means of a differential gear system. Although this choice lightens the leg and minimises required torques for the leg in transfer phase, not always does it provide the optimum configuration from the point of view of minimising the power consumption in a full leg cycle. An energetic study that includes both leg configurations and trajectories must be performed for establishing the optimum mass distribution.

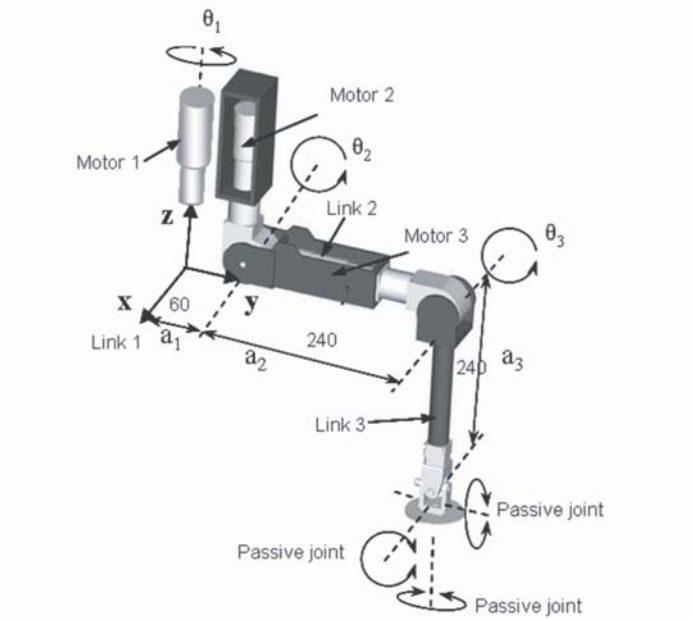


Fig. 1. Leg configuration and dimensions in millimeters

2 Preliminary Approach

In this paper, a leg from the SILO4 walking robot [2] has been used as a testbed for both simulating and providing experimental data. It is a reptile-type leg in which the motor that moves the link i is located on the link $i-1$, as shown in Fig. 1. The first step for solving the mass distribution problem consists in building and validating a power consumption model for the testbed leg. This model, which is widely described in [3], can be summarized in (1) to (4). Let $\theta(t)$ be the angular speed of a joint. Then, the expression for the instant power $P(t)$ and current $I(t)$ in the motor of that joint is:

$$P(t) = RI^2(t) + LI(t)\frac{dI(t)}{dt} + K_b\dot{\theta}(t)I(t) \tag{1}$$

$$I(t) = \frac{\tau_T(t)}{k_M} = \frac{\tau_{L-E}(t) + \tau_f(t)}{k_M} \tag{2}$$

where R, L, K_b, k_M are motor constants, $\tau_{L-E}(t)$ is the dynamic torque that derives from the Lagrange-Euler equation (3):

$$\tau_{L-E}(t) = D(\theta(t)) \cdot \ddot{\theta}(t) + h(\theta(t), \dot{\theta}(t)) + c(\theta(t)) + J^T\overrightarrow{F} \tag{3}$$

and $\tau_f(t)$ is the friction torque that results from considering static and viscous friction with Strybeck effect and asymmetry in direction of rotation ($d = \{+, -\}$), as described in (4):

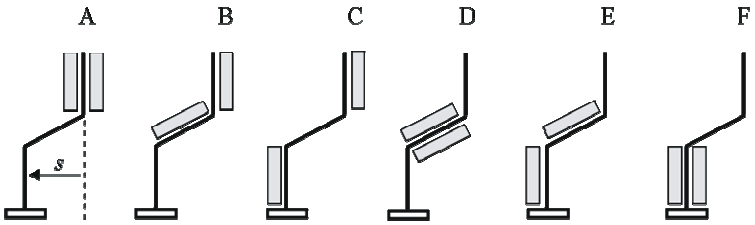


Fig. 2. Possible allocations for motors 2 and 3

$$\tau_f(t) = (\tau_{C_d} + (\tau_{S_d} - \tau_{C_d})\, e^{-|\dot{\theta}(t)|/\dot{\theta}_{S_d}} + c_{V_d}|\dot{\theta}(t)|)\mathrm{sgn}(\dot{\theta}(t)) \tag{4}$$

being *sgn* the sign function. Friction parameters $\tau_{C_d}, \tau_{S_d}, \theta_{S_d}, c_{V_d}$ are previously unknown and must be fitted by comparing with experimental measurements in the validation process.

From this validated power consumption model, the leg's energetic behaviour is now simulated for other possible mass distributions. Due to mechanical reasons, motor 1 is considered to be located always in the body. However, different locations for motors 2 and 3 are tested and evaluated, as shown in Fig. 2. Taking into account that the leg weighs 3.9 kg and each motor weighs 0.5 kg, the mass redistributed is a 25% of the total leg mass.

Different allocations for motors 2 and 3 along the leg imply different mass distributions. It means that torques during the leg cycle will be different depending on the distribution. For example, distribution F will require higher torques in transfer phase and lower torques in support phase if compared with distribution A for a given trajectory. The total energetic balance depends not only on the duty factor β but also on many other factors like the leg trajectory, speed profile, motor weight, gear reduction ratios and motor-gear efficiency. In order to get the greatest difference of energy expenditure between two configurations, distributions A and F are selected to be simulated, since they have the mass concentrated in opposite extremes of the leg. Then, the validated energetic model of the leg is simulated for both mass distributions keeping the leg trajectory unchanged. The duty factor has been set to $\beta = 0.5$, which matches with a hexapod walking with alternating tripods [4]. The study is performed taking into account different distances (parameter s in Fig. 2) from the body to the foot in the support phase for finding out how does it affect to results. The data obtained from simulation with 10 s leg cycle, 34 cm stride length and motor weight of 0.5 kg are shown in Fig. 3.

Figure 3 shows that distribution F, in which the mass is concentrated close to the foot, spends less power during a leg cycle than distribution A, in which motors are near the hip, for any distance between the body and the foot in the support phase. For a reptile configuration, which corresponds to a foot-body distance of 30 cm, the simulated power consumption with distribution F is 6.7% lower than the power spent with distribution A. It means that allocating

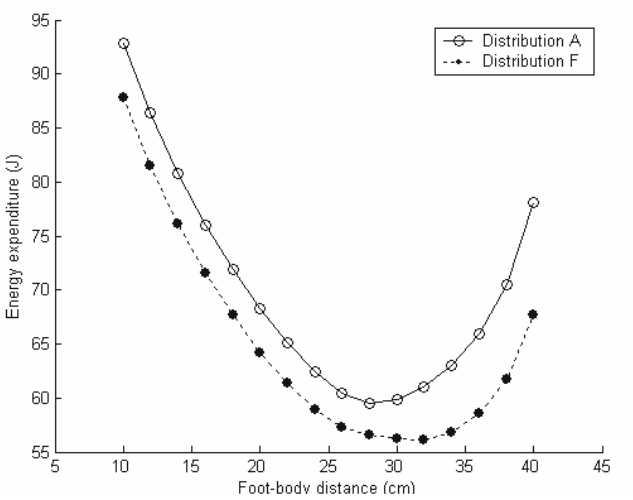


Fig. 3. Energy expenditure for mass distributions A and F vs. foot-body distance

motors near the foot in this machine is a better choice that allocating them next to the hip from the point of view of power economy.

Equations (1) to (2) and results for power consumption imply that required torques at the joints are lower with distribution F than with distribution A, what allows decreasing reduction ratios. Provided that lower reduction ratios imply higher gear efficiencies, power saving is increased in a double manner as shown in Fig. 4. Additionally, lower reduction ratios allow higher joint speeds, what implies that transfer times can be shortened and therefore walking speed can be increased.

3 From Simulation to Experiment

The same trajectory as considered in previous section is now programmed on the leg for comparing simulation results with experimental measurements of power expenditure. In order to adapt the testbed leg to experiments, distributions A and F are approached by loading a mass of 1 kg (which is the mass of two motor-gears) on link 1 and link 3 respectively. The experiment consists of making the leg to walk with both distributions and recording the power signals in order to evaluate the real energy and torque-saving ratio. Although instant power is directly obtained by measuring instant voltage and current, torque is considered to be proportional to current as described in (2).

By integrating the measured instant power signal (plot in Fig. 5) over the full leg cycle, it is obtained that distribution F saves a 7.4% of energy if compared with distribution A, a slightly greater value than the 6.7% obtained from simulation. This difference can be explained in terms of mechanical-

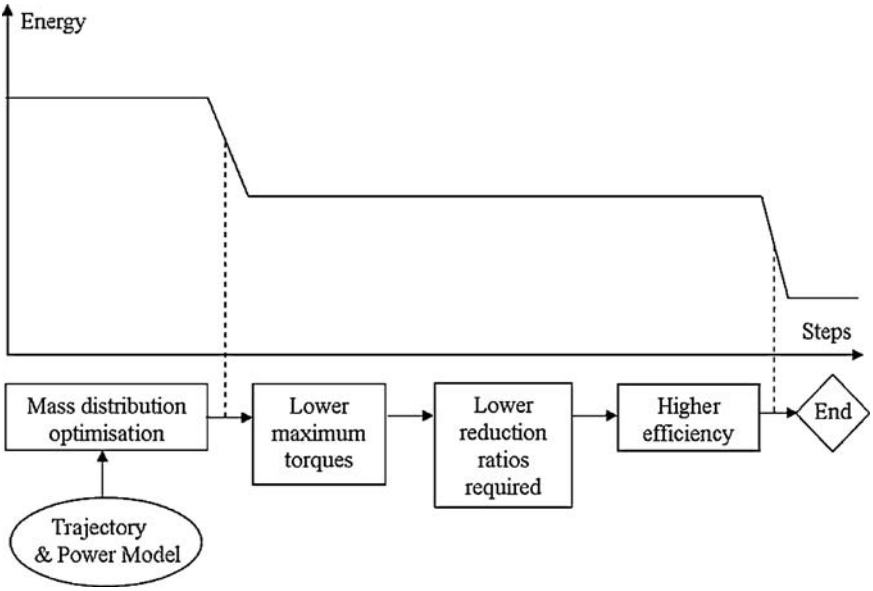


Fig. 4. Procedure for saving power on a walking machine

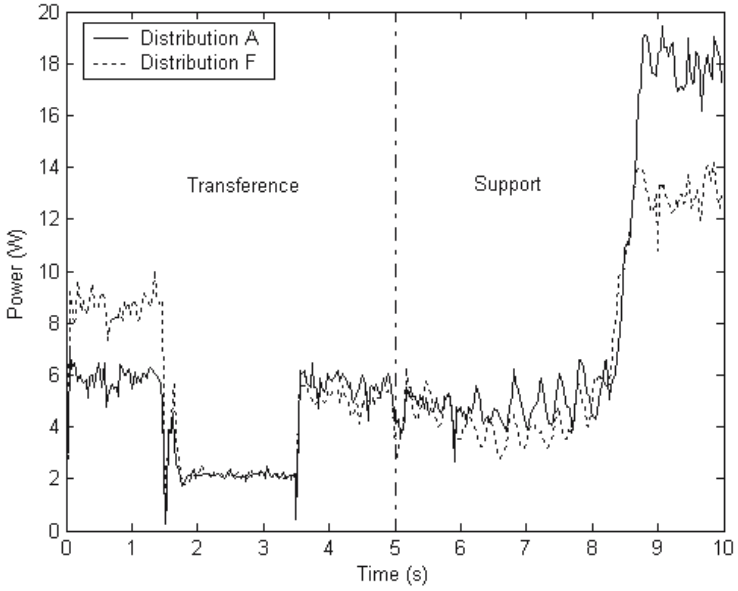


Fig. 5. Total instant power measured on the leg during a full leg cycle

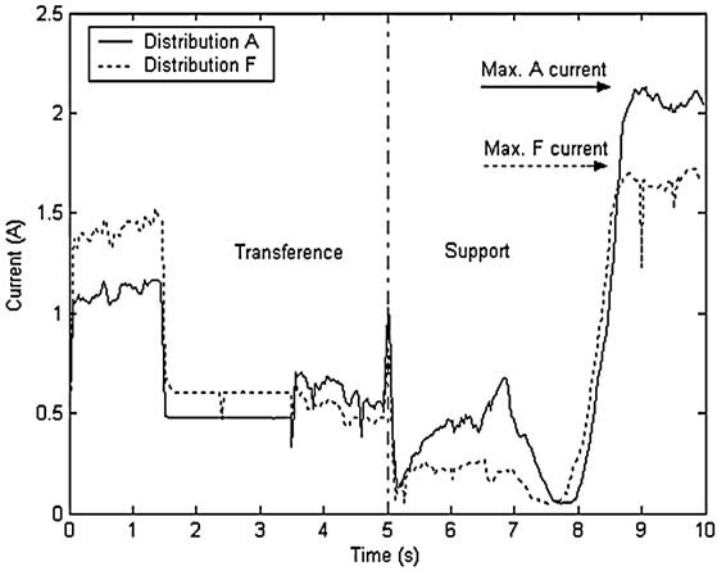


Fig. 6. Instant current measured on motor 2 during a full leg cycle

stress loses caused by link flexion. This effect, which was not considered in simulation, makes that the more the mass is concentrated near the hip, the more supporting torques grow and the more friction increases at the joints.

When separate power signals are inspected, it is observed that motor 2 is the main responsible for this power saving, since it spends most of the total power consumption. Thus, the instant current on this motor is plot in Fig. 6 in order to look for changes in maximum torque, following the procedure described in Fig. 4. By observing Fig. 6 it is found that maximum current in motor 2 with distribution F is around a 20% lower than with distribution A. Using (2) yields that maximum torque is also reduced in the same quantity. Provided that reduction ratio can be decreased at the same ratio as maximum torque decreases, the gear attached to motor 2 can be replaced by a gear with a 20% lower reduction ratio. As mentioned in the previous section, lower-ratio gears are more efficient than those with higher ratios, what means that replacing this gear improves yet more the power performance in accordance with Fig. 4.

Figure 7 shows the manufacturer curve that describes the efficiency of a spiral bevel gear as used in joint 2 for different reduction ratios but same diameter. From this figure, it is deduced that replacing the 20.5:1 gear of joint 2 by a 15:1 gear implies an efficiency gain of 18% in this joint. As a result, the expected total power-saving ratio for the leg can reach a 16% for distribution F with the new gearbox respect to distribution A with the original gearbox (see Table 1).

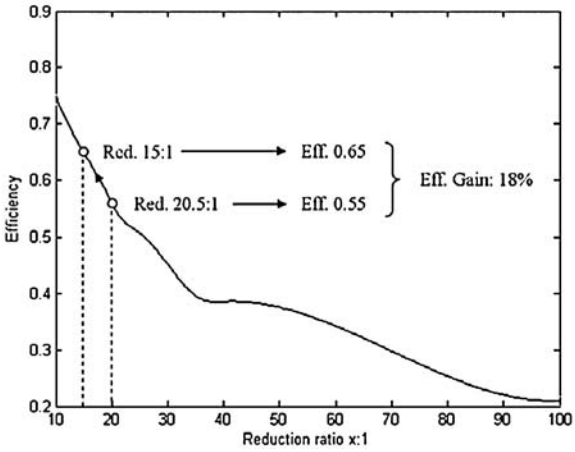


Fig. 7. Manufacturer data: gear efficiency vs. reduction ratio at constant diameter

Table 1. Energy saving as a result of optimizing mass distribution

Energy-Saving Ratio Due to Torque Decreasing	Gear Efficiency Improvement		Total Energy-Saving Ratio
7,4%	Joint 1	0%	16%
	Joint 2	18%	
	Joint 3	0%	

4 Conclusions

This paper has proved the importance of optimising the mass distribution for minimising the power consumption on a walking machine. To accomplish this task it is essential to build an energetic model, perform a validation process and simulate the robot energetics for the leg trajectory and gait parameters that will be considered in walking.

The power saving that derives from selecting one mass distribution or another has been achieved in two steps. In the first step energy saving has been obtained by optimising the mass allocation along the leg, which determines each motor to be located near the body, close to the foot or in an intermediate position.

In the second step, an extra power saving has been obtained by using lower reduction ratios in those joints in which required torques are decreased after optimising the mass distribution. Provided that lower reduction ratios imply higher efficiency, the machine's energetics has been improved yet more. The total energy-saving ratio obtained with this method has reached a 16% with respect to the original robot, which was designed without taking into account this criterion.

Acknowledgements

Part of the work presented on this paper has been supported by MCYT (Spanish Ministry of Science and Technology) under Grant DPI2001-0469-C03-02.

References

1. Gonzalez de Santos, P., Garcia, E., Cobano, J.A., Guardabrazo, T. (2004) Using Walking Robots for Humanitarian De-mining Tasks. In: Proceedings of the 35th International Symposium on Robotics (ISR), Paris, France.
2. Gonzalez de Santos, P., Galvez, J.A., Estremera, J., Garcia, E. (2003) SILO4 – A true walking robot for the comparative study of walking machine techniques. IEEE Robotics and Automation Magazine 10, 4: 23–32
3. Guardabrazo, T., Gonzalez de Santos, P. (2003) A detailed power consumption model for walking robots. In: Muscato, G., Longo, D. (eds) Proceedings of the 6th Conference on Climbing and Walking Robots, Catania, Italy, pp. 235–242.
4. Song, S.M., Waldron, K.J. (1989) Machines that walk: the adaptive suspension vehicle. The MIT Press. Cambridge, Mass.

Design of Dual Actuator for Walking Robots

Teodor Akinfiev, Roemi Fernandez, and Manuel Armada

Industrial Automation Institute, Spanish Council for Scientific Research
Department of Automatic Control. La Poveda 28500 Arganda del Rey, Madrid, Spain
teodor@iai.csic.es

Summary. In this paper a design of special dual drive for walking machines is considered. New drive gives a possibility to use the same motor for realization of two very different kind of movement – while a body of the robot is moving, this drive would work as a conventional drive, and while legs of the robot are moving, it would work as a resonance drive. It is shown that such drives are useful for walking robots in cases when robot must perform technological process simultaneously with a displacement of a robot's body. Results of design, simulations and experiments are presented. It is shown that the use of these drives allows a considerable increasing of machine's effectiveness.

1 Introduction

It is a matter of common knowledge that extremely low speed and high energy expenses characterize walking robots with conventional drives [1]. It is known that the use of the resonance drives allow substantially a gain in speed of the robot at simultaneous lowering of energy expenses 10–50 times [2]. At the same time it is necessary to recognize, that usage of resonance drive in walking robots is not always justified and its usefulness depends on the area of application of the walking robot. If such walking robot is utilized only as a vehicle (including for displacement of the technological robot, which performs some operation while the walking robot rests in a given position [3]), the use of resonance drive is justified [4]. If the walking robot is utilized to perform a technological operation (welding, painting etc.) during robot's body (on which a corresponding equipment, for example, welding head, is fixed) movement [5], the use of resonance drive creates considerable complications. This is so because while using resonance drive, the law of motion of operating mechanism is completely determined by resilient elements and can be corrected by a motor to a small extent. So, using linear springs, the law of motion will be harmonic, and for realization of a manufacturing process a totally different law of motion of the robot's may be required, for example, driving of a body with constant

speed for a realization of welding process. Let's mark, that the requirements of a manufacturing process superimpose limitations on the law of motion of the robot's body, but do not superimpose any limitations on the law of motion of legs of the robot in a phase of their motion. As the phase of the motion of legs is supplementary in relation to a manufacturing process, for increase of productivity it is desirable to move the legs as rapid as possible.

Another aspect, connected with the same problem, is that too rapid moving of robot's body is accompanied by high accelerations. These accelerations create high inertial forces, which can lead to a slippage of legs of the robot or tipping of the robot.

The mentioned above shows that to solve these problems it is desirable to create a special drive, which would possess double properties – while body of the robot is moving, this drive would work as a conventional drive, and while legs of the robot are moving, it would work as a resonance drive. This is a considerably complicate solution because the same drive will be utilized for moving of both a body of the robot (when robot's legs rest on a base) and its legs (when the corresponding leg does not contact with a base). In the present paper the possibilities of design of such kind of dual drive for walking machines are considered; the proposed solution [6] is illustrated by the example of quadruped walking robot.

2 Technical Descriptions of Drives

Let us consider the problem of design of dual drive for quadruped Cartesian walking robot, which is moving upon a horizontal surface.

The quadruped walking robot with a body 5 is shown on Fig. 1. On the robot's body there are fixed all legs of the robot with a possibility of a progressive movement. The robot has two resilient elements. One extreme point of the first resilient element 6 is connected with a leg 1 of the robot, and another extreme point of the first resilient element 6 is connected with a leg 2 of the robot. One extreme point of the second resilient element 7 is connected with a leg 4 of the robot, and another extreme point of the second resilient element 7 is connected with a leg 3 of the robot.

The electric motors 8 are connected to a control system; they use a screw-nut transmission. At high speed of rotation of the screw in a screw-nut transmission it occurs sometimes that vibrations of the screw appear (especially for lengthy screws), leading to increase of force of friction, and, in some cases, even to a fracture of the screw. This being so, a special variant of a screw-nut transmission is offered (Fig. 2). In the proposed variant each of drive motors 8 is fixed on the corresponding leg of the robot and is connected with a nut 10 through pulleys 11, 12 and belt 13. The nut 10 is fixed on a leg of the robot with a possibility of rotation but without a possibility of a progressive movement. The screw 9 is rigidly fixed on a body 5 of the robot parallel to a trajectory of the movement of legs of the robot. In this case the screw does

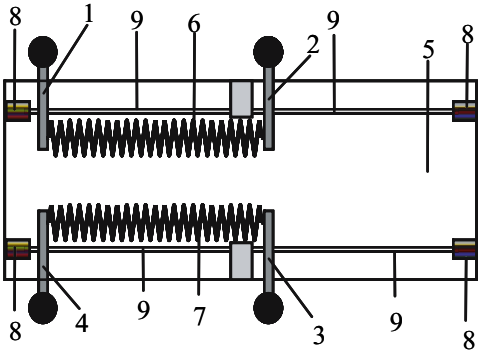


Fig. 1. Kinematic configuration of dual drives

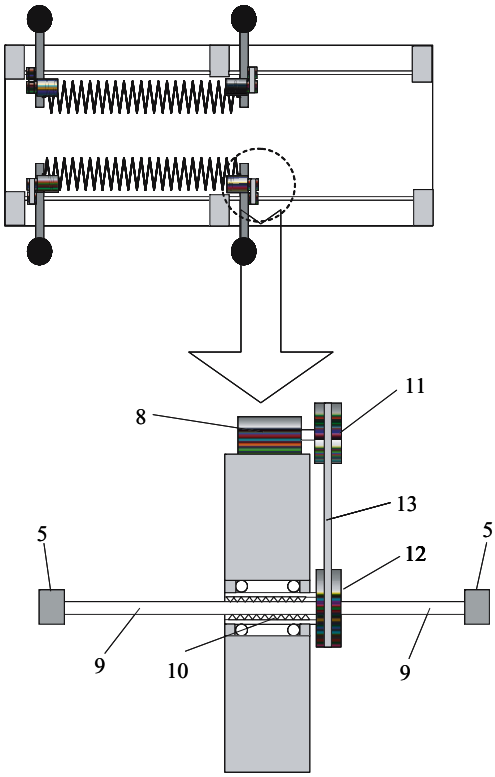


Fig. 2. Transmission pulley – belt – screw – nut

not rotate, which allows a realization of high speed of movement of a leg even at its major move (that corresponds to major length of the screw).

It is interesting to notice that the placement of electric motors on a leg of the robot is additionally advantageous because at resonance movement of a

leg, its mass magnitude is not critical (to have the same velocity it is required only the corresponding increase in rigidity of springs), while for intensive acceleration of a body of the robot it is desirable to have the least possible mass of the body (conventional type of moving).

The presence of resilient elements leads to a necessity of a great force on the part of the drives to be able to hold the legs of the robot in extreme positions, which lead to unnecessary energy expenses. To eliminate this defect, a transmission between drive motor and leg of the robot (for example, screw-nut transmission) can be made as selfbraking.

The alternative variant is to use fixing rods as mechanical latches with the special devices for latches release. The important feature of such fixing rods is that they should retain a leg from relative (i.e. in relation to the other leg of the robot), instead of absolute (i.e. in relation to a body of the robot) displacement.

3 Drive's Operation

In the considered robot each of drive motors provide two essentially different modes of movement. The movement of a body of the robot, when all legs of the robot rest on a base, is carried out in a regime of a conventional drive. At this moving, the springs do not affect the movement of a body of the robot, because the distance between legs does not vary (Fig. 3) and the springs are fixed by the latches. In this case, the law of motion of the body of the robot is determined only by force of motors. This allows (using a control system) a realization of the law of motion, defined in according to the requirements of a manufacturing process, in the same way as when using conventional drives.

The second mode of movement (a movement of any leg of the robot when robot's body does not move) is performed with both spring and corresponding motor effecting upon a leg simultaneously. During this movement the spring helps to get high acceleration of a leg on the first part of a trajectory and heavy braking on the second part of a trajectory, the motor serving only for compensation of friction losses during this movement in the same way as it occurs in resonance drives.

Thus, the designed drive has double properties – at moving of a body of the robot it works in a conventional drive mode, and at moving of each of legs it works in a resonance drive mode.

4 Equations for Drives' Motion

Differential equations, which describe the horizontal movement of legs and body of robot with new drives, can be written in form:

$$m_1\ddot{x}_1 + b\dot{x}_1 + cx_1 + N_1 Sign(\dot{x}_1) = kM_1 \,, \tag{1}$$

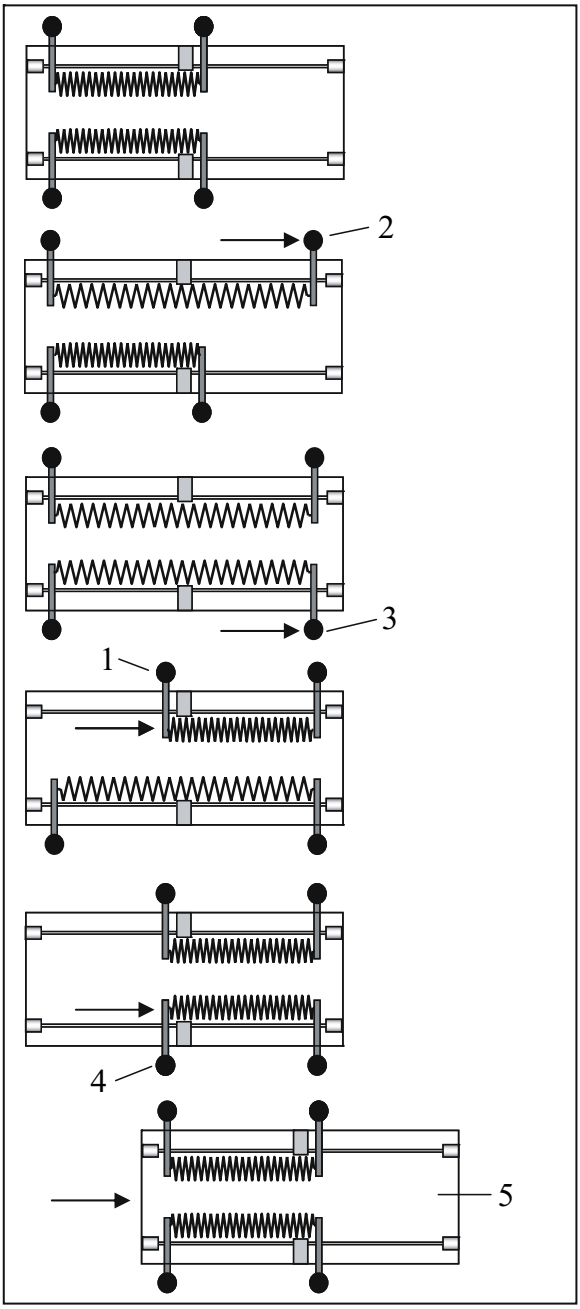


Fig. 3. Operation of robot with dual drives

$$m_2\ddot{x}_2 + 4b\dot{x}_2 + N_2 Sign(\dot{x}_2) = 4kM_2 \ , \tag{2}$$

where m_1 and m_2 – mass of the leg and the body, b – coefficient of viscosity friction, c – rigidity of spring (in common case, $c = c(x_1)$), N_1 and N_2 – forces of dry frictions during the movement of the leg and the body ($N_1 = \mu g m_1, N_2 = \mu g m_2$), k – transmission ratio, M_1 and M_2 – torque of the motors in time of movement of leg or body.

Torque of the motors can be calculated by formula [7]:

$$M_i = \gamma I_i, \quad i = 1 - 2 \ , \tag{3}$$

where γ – is constant parameter of motors, I_i – current in armatures of motors. For calculation of the current the relationship is used:

$$I_i = \frac{U_i - E_i}{R} - \tau \dot{I}_i \tag{4}$$

where U_i – motors terminal voltage, R – armature resistance, $E_i = k\gamma\dot{x}_i$ – the back – EMF, τ – electromagnetic constant.

Motor terminal voltage depends essentially on the type of phase of the robot movement that is carried out at the moment. Thus, when robot's leg is moving (resonance mode of movement), the control system should feed a motor with such voltage, which would allow compensation of energy losses during this moving and would provide leg's coming to the terminal point of a trajectory with a zero velocity. The previously elaborated [8] universal algorithm of adaptive control solves effectively this problem.

When a body of the robot is moving (a conventional drive's mode of movement), a conventional control algorithm [9] is used: at first, a control system provides maximally intensive robot's body acceleration up to the desired speed, then the control system maintains this speed constant, and finally, as the terminal position is approaching, this control system provides the maximally possible deceleration and stopping.

5 Simulation

Simulation was made using MATLAB-SIMULINK to detect the dynamic characteristics of the considered dual drive and to evaluate its technical features. As an example, the quadruped walking robot is considered with the following characteristics: mass of the robot's body is 30 kg, mass of the robot's leg is 2 kg, the maximum displacement of each leg is 0,2 m, rigidity of each spring is 500 N/m. As an example of a technological operation, an operation of welding is considered with the help of a welding head anchored on a body of the robot. Welding should be carried out with the body of the robot moving with velocity 0,02 m/s.

Let us notice that in previous numerous experiments it was shown that for resonance drives the time of displacement of a working element from one

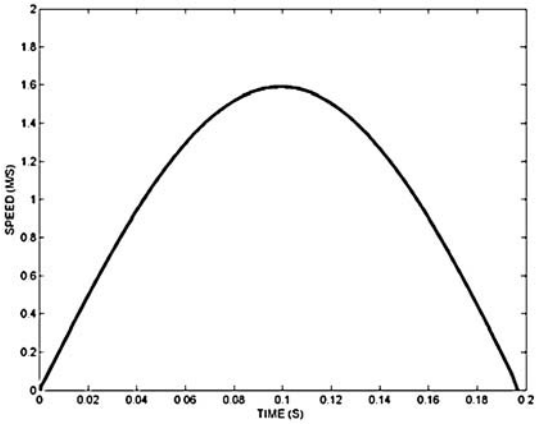


Fig. 4. Movement of a leg: speed vs. time

extreme position to another can be 0,2 s [2]. The further diminution of this time is possible, but creates particular difficulties while technical realization.

The results of simulation are presented on Fig. 4 and Fig. 5. The displacement time of a leg of the robot from one extreme position to another is 0,2 s, while the maximal velocity is 1,59 m/s. The displacement of a body of the robot on 0,2 m takes 10,1 s, and both acceleration time (up to demanded velocity of 0,02 m/s) and braking time make about 0,1 s. Thus, each motion cycle of the robot (displacement of four legs and a body of the robot) is performed in 10,9 s, with the working operation (welding) time being 9,9 s. Consequently, while using of the robot with dual drives, the relationship between the time of the preliminary operation and the execution time of the working operation makes 1:10.

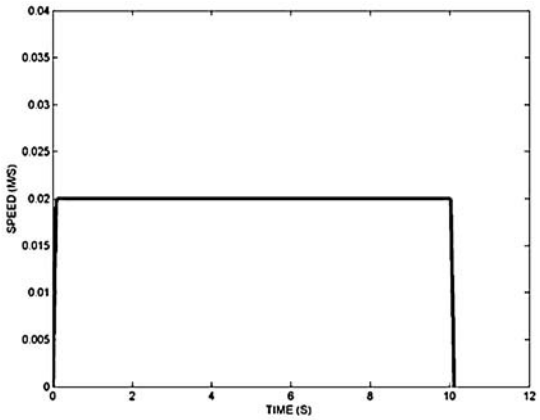


Fig. 5. Movement of a body: speed vs. time

If for execution of the considered welding operation the robot were used with conventional drives, for example, the quadruped robot SILO 4 [1, 10](the speed of displacement of a body of the robot is 0,02 m/s, but the speed of a leg is 0,2 m/s), then for such robot the relationship between the time of the preliminary operation and the execution time of the working operation makes 2:5. This confirms effectiveness of the designed dual drives, which decrease the time of preliminary operation four times.

It is interesting to notice that the use of a dual drive allows not only the increase of displacement speed of robot's legs, but also reduces energy expenses. This is connected with the fact that at resonance movement a motor should not compensate inertial force. This force is compensated by a passive resilient element, with the motor serving only for compensation of friction during movement.

6 Conclusions

Design of a new dual actuator for walking robots is presented. The motor of the new actuator is able to work in two different modes. The first mode is a conventional mode when the drive realizes a movement of robot's body. The second one is a resonance mode, when the drive accomplishes a movement of robot's legs. For each mode a special algorithm of control is proposed. This design is useful for walking machines in cases when robot must perform technological process simultaneously with a displacement of a robot's body. In this case it is possible to increase productivity of robot considerably. The obtained results are confirmed by simulations.

Acknowledgments

The authors would like to acknowledge the financial support from Ministry of Science and Education of Spain: Fellowship F.P.U. and Project "Theory of optimal dual drives for automation and robotics".

References

1. Walking Machine Catalogue. http://www.walking-machines.org/
2. Akinfiev, T. (1996) The Resonance Drives with Adaptive Control. In: 11th ASCE Engineering Mechanics Conference, ASCE Press, New York, USA.
3. Gonzalez de Santos, P., Armada, M., Jimenez, M. (2000) Ship Building with ROWER. IEEE Robotics and Automation Magazine. Vol. 7, No 4.
4. Akinfiev, T., Fernandez, R., Armada, M. (2003) Optimization of Parameters of Resonance Drives for Walking Robots. Proc. 6th International Conference on Climbing and Walking Robots. Professional Engineering Publishing limited, London, UK.

5. Akinfiev, T., Armada, M. (2001) Automatic robotic device for welding of large-sized details. In: 3rd Workshop on European Scientific and Industrial Collaboration WESIC. Published by Drebbel Institute for Mechatronics, Enschede, The Netherlands.
6. Akinfiev, T., Armada, M., Fernandez, R. (2004) Four-legged Robot for Technological Process. Spanish patent application 20041315.
7. Chilikin, M., Sandler, A. (1981) General Course of Electric Drives. Energoizdat, Moscow, Russia.
8. Akinfiev, T. Method of Controlling of Mechanical Resonance Hand. USA, Patent 4958113.
9. Besekerskii, V., Popov, E. (1975) Theory of Automatic Control Systems. Nauka, Moscow, Russia.
10. Gonzalez de Santos, P., Galvez, J., Estremera, J., Garcia, E. (2003) SILO4 – a True Walking Robot for the Comparative Study of Walking Machine Techniques. IEEE Robotics and Automation Magazine, Vol. 10, No. 4.

Part VI

Hopping, Biped and Humanoid Robots

Control of a 3-D Hopping Apparatus

V.B. Larin

The Institute of Mechanics NAS, Nesterov str., 3, 03057, Kiev, Ukraine
larin@uaac.freenet.kiev.ua

Abstract. The model of spatial movement of the hopping device is considered taking into account the inertial properties of its leg. For such model of the hopping device the control system is synthesized. The choice of program of movement and synthesis of system of stabilization were considered taking into account the shock processes connected with touching of the leg with surface. Procedure of synthesis of system of stabilization was based on the method of optimization of periodic systems. The results of numerical modeling of non-stationary movement of the hopping device testify the high efficiency of the synthesized system of stabilization.

1 Introduction

It is considered three-dimensional (3D) model of hopping apparatus (HA) with a inertial leg. This model is similar to the model of HA described in [13] (see Fig. 1). On the Fig. 1 it is designated: 1 – two-axis gyroscope, 2 – compass, 3 – gimbal, 4 – computer, 5 – the hydraulic actuator and position/velocity sensors, 6 – leg. The similar model with a weightless leg has been considered in [7]. However, as against of the model [13] the considered model can turn the torso concerning to the vertical axis. Here as well as in [12] it is supposed, that the effort in the leg is created by a spring with a straight-line characteristic, instead of compressed air as in [13].

At a choice of program of movement and synthesis of system of stabilization of HA it is taken into account the shock processes in the moment when the leg is touching the ground. As well as [5, 8] these shock processes are considered within the framework of Carnot theorem (absolutely not elastic impact). Synthesis of system of stabilization is based on the method of optimization of periodic systems. This approach as it is noted in [3], allows to decrease the loading on the actuators. In the case of the considered model of HA, it is shown, that it is possible the essentially decrease of the loading on the actuators of control system of HA

Spatial character of movement has caused the requirements to detailing of structure of the leg and use, as well as in [7], the general algorithms [2] drawing

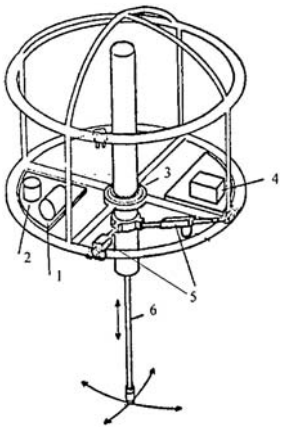


Fig. 1.

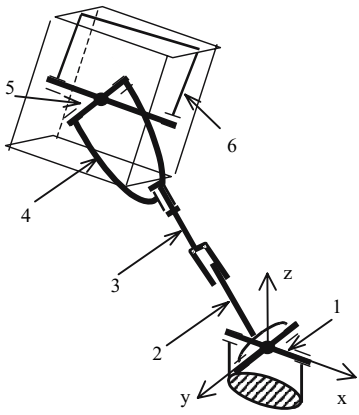


Fig. 2.

up of the equation of movement of robotics systems.. In the paper for deriving equations of motion the method of the Lagrange-Euler will be utilized [2] . It includes a description of kinematic of system by means of matrix representation of Denavit-Hartenberg with consequent usage of Lagrangian formalism. However, the considerable computational difficulties caused by usage of this method are known [2]. For overcoming these computational difficulties the system of analytical computation "Symbolic" of software package MATLAB was used. It has allowed to receive analytical expressions for the Lagrange equations and thus it is essential to raise efficiency of procedure of modeling of movement of HA.

2 The Equations of Movement

On a stance phase of movement as a vector of the generalized coordinates is accepted the following vector

$$q = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6]' \tag{1}$$

Hereinafter the prime means transposition. Expression for kinetic energy of HA has the form

$$\Lambda_0 = \frac{1}{2}\dot{q}' D_0 \dot{q} \tag{2}$$

where the matrix D_0 is a function of the generalized coordinates, of the inertial parameters of the torso and inertial part of the leg (the link "4"). The inertial parameters are the following: $\mathbf{M}$ – mass of the torso, J_x, J_y, J_z its principal moments of inertia, M_n – mass of the leg, J_{nx}, J_{ny}, J_{nz} – central moments of inertia of the leg – displacement of centre of gravity of the leg. For determination of the matrix D_0 the standard procedures [11] were used (details see in [2]).

Let's note, that HA in a phase of flight gets additional two degrees of freedom. In this connection, with the purpose of introduction of the additional generalized coordinates we shall assume, that the point of support of the leg can freely move along the x and y axes. Having added two additional coordinates q_x, q_y to the entered above vector of the generalized coordinates q which are corresponding to moving of a point of support of the leg along the x and y axes accordingly. In other words, on a phase of flight the position of HA will be described by the following vector of the generalized coordinates

$$\bar{q} = [q_x \quad q_y \quad q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6]' \tag{3}$$

Accordingly, the expression for kinetic energy on a phase of flight looks like

$$\Lambda_f = \frac{1}{2}\dot{\bar{q}}' D_f \dot{\bar{q}} \tag{4}$$

Using procedures [2] (details see in [11] we shall receive the expressions for potential energy oh HA and by that we shall determine the functions of Lagrange L_0 and L_f which are corresponding to a stance phase and a phase of flight.

If to designate $u = [u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \quad u_6]'$ as the vector of the generalized forces (controlling actions), operating in the appropriate links of the leg, it is possible to derive the Lagrange equation of which is describing the driving of HA during a stance phase,

$$\frac{d}{dt}\frac{\partial L_0}{\partial \dot{q}} - \frac{\partial L_0}{\partial q} = u \tag{5}$$

Relative to the components of the vector u the following assumptions are accepted $u_1 = u_2 = 0$

$$u_3 = -c(q_3 - q_3^0 - \delta q_p - \delta q_s) \tag{6}$$

The moments u_4, u_5, u_6 have constant values during a stance phase. In (6) c is rigidity of the spring, q_3^0 is the value of coordinate q_3 at the moment of touch of the leg with the ground, δq_p is the program value of the deformation of the spring, δq_s is the deformation of the spring used as control action at the solution of problems of stabilization.

If to introduce the vector of the generalized forces

$$\bar{u} = [u_x \quad u_y \quad u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \quad u_6]' \tag{7}$$

that the Lagrange equation on a phase of flight can be noted as follows

$$\frac{d}{dt}\frac{\partial L_f}{\partial \dot{\bar{q}}} - \frac{\partial L_f}{\partial \bar{q}} = \bar{u} \tag{8}$$

In (8) the vectors $\bar{q}, \bar{u}$ are determined by the expressions (3), (7). We shall note, that on a phase of flight the first six component of the vector $\bar{u}$ are equal to zero ($u_x = u_y = u_1 = u_2 = u_3 = u_4 = 0$) the others ($u_5, u_6$) are constants during a phase of flight.

3 Change of Phases of Movement HA

Except of the Lagrange equations (5), (8) for the mathematical description of the model of HA it is necessary to formalize the processes of transition between phases of a jump (transition from a stance phase in a phase of flight and on the contrary). We shall designate as t_0 – the moment of the beginning of a stance phase (the moment of touch of the leg and the ground), t_f – the moment of the termination of a stance phase, t_u – the moment of the termination of a phase of flight. As to each jump there corresponds the frame which origin of coordinates coincides with the point of support of the leg on the corresponding stance phase the process of transition from a basic phase in a phase of flight is described by the following expressions:

$$\bar{q}(t_f) = [0 \quad 0 \quad q(t_f)']', \dot{\bar{q}}(t_f) = [0 \quad 0 \quad \dot{q}(t_f)']' \tag{9}$$

The vectors $q, \bar{q}$ have been determined above.

Under the assumption, transition from a phase of flight in a stance phase (touch of a leg with the ground) is accompanied by absolutely not elastic impact. Thus, the phase coordinates are changing continuously, i.e. values of the vector q (correspond to the moment $t_u + 0$) and the vector $\bar{q}$ (correspond to the moment $t_u - 0$) are connected by the relation:

$$q(t_u + 0) = \bar{q}(3 : 8) \tag{10}$$

In (10) the designations of MATLAB are used, namely $\bar{q}(3 : 8)$ means a vector containing 3 to 8 components of a vector $\bar{q}$. The generalized velocities

at the impact moment have interruption which is described by the Carnot theorem. Namely (details see, for example [5]

$$\dot{q}(t_u + 0) = f(f' D_f f)^{-1} D_f \dot{\dot{q}}(t_u - 0) \tag{11}$$

In (11) the matrix (8×8) D_f, determines kinetic energy of HA on a phase of flight (expression (4)); the matrix $(6 \times 8) f = [\mathrm{I\ O}]$, here O is 6×2 a matrix with zero elements, I – the identity matrix. Further an identity matrix of the corresponding size we shall designate as I.

4 A Choice of a Program Trajectory

As well as in [5, 7, 8] a problem of synthesis of control system of HA we shall solve in two stages. On the first, we shall choose a program trajectory and corresponding control actions, on the second – we synthesize system of stabilization of this program movement. Both these stages are in detail described in [5, 8, 9] with reference to the similar models of HA, therefore we shall be limited only to a summary of corresponding procedures with reference to the consider model of HA. There are assumed: the total time of a jump $\tau = t_u - t_0$, duration of a stance phase $\tau_0 = t_f - t_0$, the average speed of movement of HA along the x axis: $v_x = \frac{\ell}{\tau}$, where ℓ is the moving of HA along the x axis during one jump.

As control actions on a stance phase the generalized forces (moments) u_4, u_5, u_6 and compression of the spring δq_p (see (6)) are used. On a phase of flight, the moments u_5, u_6 are used. All these control actions have constant values during a corresponding phase of movement. The problem of a choice of the program trajectory of movement of HA along the x axis is formulated as a two-point boundary value problem with the unseparated boundary conditions for the object which is described by the Lagrange equations (5), (8) and the expressions (9)–(11):

$$q(t_0 + 0) = q(t_u + 0), \dot{q}(t_0 + 0) = \dot{q}(t_u + 0) \tag{12}$$

$$q_x(t_u) = \ell, q_y(t_u) = 0$$

Besides it is necessary to specify the value of $q_3(t_0 + 0)$ (length of a leg in the beginning of a jump) which ensure the desirable value of z_0 – heights of the center of gravity of torso of HA.

If in this problem as goal to choose the quadratic functional of the mentioned above the control actions, that, as shown in [5, 9] for its solution it is possible to use the method of quazilinearization. In turn, each step of this method can be realized by the standard procedure of solution of the problem of quadratic programming, namely procedure *qp.m* of the MATLAB package [10].

5 Synthesis of System of Stabilization

For stabilization of a program of motion it is necessary to obtain an equations, which describe a dynamics of small deviations of generalized co-ordinates and velocities from their program values (i.e. to derive the equation in variations) and to determine a control actions, which allow to stabilize a program of motion. Let's note, that owing to periodicity of the program trajectory, it is enough to receive the equations in variations only for one step, for example for step with number i. Further, as well as in [3, 5, 7, 8, 12] a problem of synthesis of system of stabilization we shall consider in the discrete statement. In this connection, we shall consider on a program trajectory three vectors:

$$p^p_{(j)} = \left[q' \dot{q}'\right]', (j = 0, 1, 2) \tag{13}$$

which are corresponding, depending on an index j, to the following phases of movement of HA:

- at $j = 0$ it is corresponding to the moment of touch a leg and the ground (the values of components of the vector $p^p_{(0)}$ are determined by components of the vectors $q(t_f), \dot{q}(t_f)$);
- at $j = 1$ – to the moment of the beginning of a phase of flight (the values of components of the vector $p^p_{(1)}$ are determined by the vectors $q(t_f), \dot{q}(t_f)$);
- at $j = 2$ – to the moment $t_u + 0$. We shall note, that $p^p_{(2)} = p^p_{(0)}$.

As control actions which carry out stabilization of a program trajectory, the additives to the program values of the following parameters are accepted. On a stance phase:

Changes the compression of the spring δq_s (see (6)).

Changes of the moments u_4, u_5, u_6 which we shall designate as $\delta u_4, \delta u_5, \delta u_6$

On a phase of flight – additives to the moments u_5, u_6 which we shall designate as $\delta u_5, \delta u_6$.

We shall designate as $p_i(j)$ the vectors, structure of which is similar to (13), but values of the component correspond to the actual movement of HA on a step with number i. In other words, variations $\Delta p_i(j)$ of the values of a phase vector concerning the program trajectory are determined as follows:

$$\Delta p_i(j) = p_i(j) - p^p(j) \tag{14}$$

Let's note, that

$$p_i(1) = \Phi_1(p_i(0), u_4 + \delta u_4, u_5 + \delta u_5, u_6 + \delta u_6, \delta q_p + \delta q_s) \tag{15}$$

$$p_i(2) = \Phi_2(p_i(1), u_5 + \delta u_5, u_6 + \delta u_6) \tag{16}$$

The function $\Phi_1(\cdot, \cdot, \cdot, \cdot)$ in (15) determines the connection between $p_i(0)$ and $p_i(1)$. This connection is established by result of integration of the Lagrange (5). Similarly, the function $\Phi_2(\cdot, \cdot, \cdot, \cdot)$, determines connection between

the vectors $p_i(1)$ and $p_i(2)$. This function is determined by result of integration of the Lagrange equation (8) and the expressions (9)–(11).

We linearize the expressions (15), (16) concerning a program trajectory. We shall receive the equations in the variations, which are determining (in linear approximation) the dynamics of the change of the vectors $\Delta p_i(j)$, determined by (14):

$$\begin{aligned} \Delta p_i(1) &= \Psi_0 \Delta p_i(0) + \Gamma_0 w(0), \\ \Delta p_i(2) &= \Psi_1 \Delta p_i(1) + \Gamma_1 w(1), \\ w(0) &= [\delta u_4 \delta u_5 \delta u_6 \delta q_s]', \\ w(1) &= [\delta u_5 \delta u_6]' \end{aligned} \tag{17}$$

As the matrices $\Psi_0, \Gamma_0, \Psi_1, \Gamma_1$ do not depend on an index i and $\Delta p_{i+1}(0) = \Delta p_i$ (2) it is possible in (18) to omit the index i. We having introduced through numbering for the sequences $\Delta p(k)$ and $w(k), k = 0, 1, 2, 3, \ldots$. After this the problem of synthesis of a regulator of HA for stabilizing the program movement can be reduced to the problem of synthesis of regulator which is stabilizing the following periodic (with the period equal two) system:

$$\begin{gathered} \Delta p_i(k+1) = \Psi_k \Delta p(k) + \Gamma_k w(k), k = 0, 1, 2, \ldots, \\ \Psi_{k+2} = \Psi_k, \Gamma_{k+2} = \Gamma_k \end{gathered} \tag{18}$$

The solution of this problem can be reduced to construction of the solution of the discrete periodic Riccati equation [1, 4, 6]. Really, if to accept the quadratic criterion of quality:

$$\Omega = \sum_{k=0}^{\infty} \left(\left((\Delta p(k))' Q_k \Delta p(k) + (w(k))' R_k w(k) \right) \right) \tag{19}$$

where $Q_k = Q_k' \geq 0, R_k = R_k' \geq 0, Q_{k+2} = Q_k, R_{k+2} = R_k$. The regulator $w(k) = K(k)\Delta p(k)$, which is minimizing the criterion (19) on the class asymptotic stable systems (18) ($\lim\limits_{k \to \infty} \Delta p(k) = 0$), is determined by the following matrices:

$$\begin{aligned} K(0) &= -\left(\Gamma_0' S_1 \Gamma_0 + R_0\right)^{-1} \Gamma_0 S_1 \Psi_0, \\ K(1) &= -\left(\Gamma_1' S_0 \Gamma_1 + R_1\right)^{-1} \Gamma_1 S_0 \Psi_1, \\ K(k+2) &= K(k), \\ k &= 0, 1, \ldots \end{aligned} \tag{20}$$

In (20) the symmetric matrices S_0, S_1, are the stabilizing solution of the periodic difference Riccati equation:

$$S_k = \Psi_k' (S_{k+1} - S_{k+1}\Gamma_k (R_k + \Gamma_k' S_{k+1} \Gamma_k)^{-1} \Gamma_k' S_{k+1}) \Psi_k + Q_k \tag{21}$$

The numerical algorithms of solution of the equation (21) (construction of the periodic sequence of matrices S_k) see, for example [4, 10].

6 Example

As the initial data we choose the parameters of model [13]. Values of the inertial parameters appearing in (2) are the following:

$$M = 16.09\,\mathrm{kg},\ M_n = 0.91\,\mathrm{kg}\ r_n = 0,\ J_x = J_y = J_z = 0.709\,\mathrm{kgm}^2\ ,$$

$J_{nx} = J_{ny} = 0.111\,\mathrm{kgm}^2$, $J_{nz} = 0.0111\,\mathrm{kgm}^2$. The value of the coefficient c in (6) is chosen in such a manner that "eigen frequency" $\omega = \sqrt{\frac{c}{M+M_n}} = 5\pi$. The height of the center of gravity of the torso at the beginning of a jump (z_0) is equal to 0.58 m. Duration of a jump is equal to 0.38 s, duration of a stance phase 0.1 s.

Using described in Sect. 6, the algorithms of choice of a program trajectory, have been received the following values of the vectors of phase coordinates and velocities corresponding to the moment $t_0 + 0$ for a case $\ell = 0$ (horizontal speed, $v_x = 0$):

$$\begin{aligned} q &= [0 \quad 0 \quad 0.5800 \quad 0 \quad 0 \quad 0]', \\ \dot{q} &= [0 \quad 0 \quad 1.3720 \quad 0 \quad 0 \quad 0]' \end{aligned} \tag{22}$$

The following control actions are corresponding to this program of movement: on a stance phase $u_4 = u_5 = u_6 = 0, \delta q_p = 0.1271$ m; on a phase of flight $u_5 = u_6 = 0$.

For a case when HA is driving along the x axis when in (12) $\ell = 0.63\mathrm{m}, (\mathrm{v_x} = \frac{0.63}{0.38} = 1.7\,\mathrm{m/s})$, the following values for the vectors of phase coordinates and velocities corresponding to the moment $t_0 + 0$ are received:

$$\begin{aligned} q &= [0 \quad 0.1346 \quad 0.5850 \quad 0 \quad -0.1346 \quad 0]', \\ \dot{q} &= [0 \quad 2.5466 \quad 1.5874 \quad 0 \quad -3.0958 \quad 0]' \end{aligned} \tag{23}$$

The following control actions are corresponding to this program of movement: on a stance phase $u_4 = u_6 = 0, u_5 = 2.7812$ N · m $\delta q_p = 0.124$ m; on a phase of flight $u_6 = 0, u_5 = -7.7881$ N · m. For these programs of movement the regulators have been synthesized according to the procedure described in Sect. 7. The following values of the matrices appearing in (19) have been accepted. The matrices R_1, R_2 – are zero matrices, the matrices $Q_1 = Q_2 = Q$, where the matrix Q has the following structure: $Q = \mathrm{diag}\{100I, I\}$, where I is the identity matrix, size of which is equal to 6×6.

For estimation of quality of the synthesized regulators the following numerical experiments have been carried out.

1. Starting of HA. Process of transition from the program of movement determined by (22) ($v_x = 0$) toward the movement corresponding to (23) ($v_x = 1.7$ m/s). was modeled. The system of stabilization of HA was synthesized with reference to the program of movement determined by (23). Results of experiment are shown on the Fig. 3. On this figure are accordingly designated as "□", "+" the values of horizontal velocity of the center of gravity of

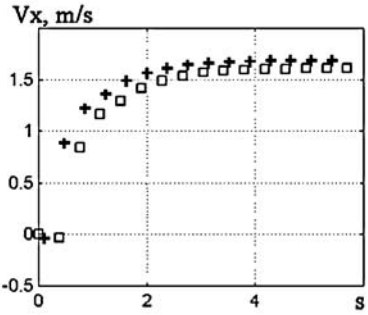


Fig. 3.

torso of HA in the beginning of a stance phase (after the impact) and at the end of a stance phase.

From these results it is possible to draw the conclusion, that attainment of HA the program mode of movement occurs for 2–3 seconds.

Other two experiments are connected to change of orientation of torso of HA and change of the direction of movement of HA.

2. Change of orientation of torso of HA. In this experiment the system of stabilization was synthesized with reference to the program of movement determined by (22). It was supposed, that HA is moving according to the program set by (22), however, at the initial moment the torso of HA is revolved on the angle equal to –0.5 radian ($q_4 = -0.5$). Process of restoration of the orientation of torso of HA is shown on the Fig. 4. From this figure it is possible to draw a conclusion, that orientation of torso of HA is restored practically during 0.5 s.

3. Change of a direction of movement of HA. It is supposed, that HA which moved along the x axis according to the program determined by (23) should change the direction of movement. Namely, since some moment HA should move along the axis which is making angle 0.5 radian with the x axis. The system of stabilization of HA is synthesized with reference to the program of

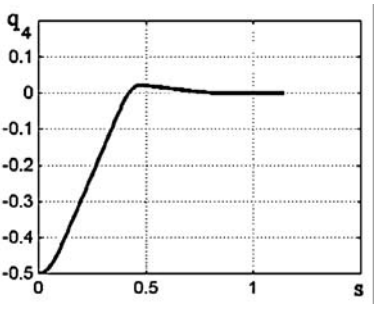


Fig. 4.

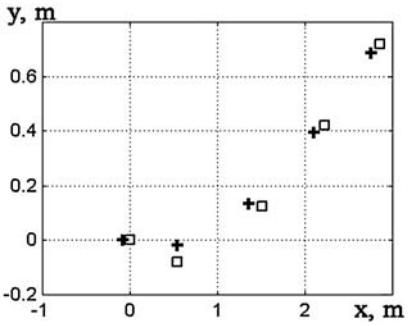


Fig. 5.

movement which is determined by (23). Results of this experiment are shown on the Fig. 5. Here by signs "□" designate the points of support of the leg on the corresponding jump, by signs "+" designate the projections to the xy plane of the center of gravity of torso of HA in the beginning of a stance phase of the corresponding jump. The shown data speak that the transitive process connected with change of the direction of movement have duration no more than 3–4 jumps.

Let's analyze efficiency of synthesized control system of HA from the point of view of loading of actuators. In this connection we compare the opportunities of the actuators of HA [13] and, received in the described above numerical experiments the values of the corresponding generalized forces. We shall be limited to comparison of these values with reference to the generalized forces (moments) which are corresponding to the generalized coordinates q_5, q_6.

According to [13] the actuators of turn of the leg (on the Fig. 1 they are designated by the number 5) can develop the moments up to 90 N·m. In the first and third experiments the maximal values of the corresponding moments did not surpass 15 N·m. Thus, it is possible to speak about the opportunity of essential decrease the loading on the actuators if the synthesis of control system of HA is doing according to the algorithms basing on the procedures, described in Sects. 6, 7.

7 Conclusion

The model of spatial movement of the hopping device is considered taking into account the inertial properties of its leg. For such model of the hopping device the control system is synthesized. The choice of program of movement and synthesis of system of stabilization were considered taking into account the shock processes connected with touching of the leg with surface. Procedure of synthesis of system of stabilization was based on the method of optimization of periodic systems. The results of numerical modeling of non-stationary

movement of the hopping device testify the high efficiency of the synthesized system of stabilization.

References

1. Aliev, F.A., Larin, V.B. (1998) Optimization of linear control systems: analytical methods and computational algorithms. Amsterdam: Gordon and Breach Science Publishers, p. 261
2. Fu, K.S., Gonzalez, R.C., Lee, C.S.G. (1987) Robotics: Control, Sensing, Vision, and Intelligence, McGraw – Hill Book Company, New York, p. 595
3. Larin, V.B. (1989) Control of a Walking Apparatus. Soviet Journal of Computer and Systems Sciences, vol. 27, no. 1, pp. 1–8
4. Larin, V.B. (1998) The optimization of periodic systems with a singular weighting matrix which determines a quadratic form of control actions. Problemi upravleniya i informatiki, no. 2, pp. 33–46 (in Russian, will be translated in J. Autom. and Inform. Sci.)
5. Larin, V.B. (2000) Control of Statically Unstable Legged Vehicles. Int. Appl. Mechanics, vol. 36, no. 6, pp. 729–754
6. Larin, V.B. (2002): Algorithm of the solution of the discrete periodic Riccati equation. Problemi upravleniya i informatiki, no. 1, pp. 77–83 (in Russian, will be translated in J. Autom. and Inform. Sci.)
7. Larin, V.B. (2003,a)) Problem of control by spatial motion of the hopping vehicle. Problemi upravlenija i informatiki, no 5, pp. 21–36. (in Russian, will be translated in J. Automat. Inform. Sci.)
8. Larin, V.B. (2003,b) Toward the Problem of Constructing the Model of a Walking Apparatus. Int. Appl. Mech., vol. 39, no. 4, pp. 729–754
9. Larin, V.B. (2003,c) Non-stationary modes of movement of the hopping device. Problemi upravlenija i informatiki, no. 1, pp. 143–154 (in Russian, will be translated in J. Automat. Inform. Sci.)
10. Larin, V.B. (2003,d) About synthesis of a robust regulator for periodic controlled system. Problemi upravlenija i informatiki, no. 6, pp. 33–46. (in Russian, will be translated in J. Automat. Inform. Sci.)
11. Larin, V.B. (2004) Spatial model of the one-legged hopping device. Int. Applied Mechanics, vol. 40, no. 5, pp. 726–736
12. Larin, V.B.V.M. (2002) A Note on a Model Hopping Machine, Int. Appl. Mech., vol. 38, no. 10, pp. 1272–1279
13. Raibert, M.H., Brown, H.B., Chepponis, M. (1984) Experiments in balance with 3D one-legged hopping machine, Int. J. of Robotic Research, vol. 3, no. 2, pp. 75–95

Learning of the Dynamic Walk of an Under-Actuated Bipedal Robot: Improvement of the Robustness by Using CMAC Neural Networks

Christophe Sabourin, Olivier Bruneau, and Jean-Guy Fontaine

Laboratoire Vision et Robotique, Ecole Nationale Supérieure d'Ingénieurs de Bourges, 10 Boulevard Lahitolle 18020 Bourges Cedex France
christophe.sabourin@ensi-bourges.fr

Abstract. In this paper, we propose a new control strategy based on the use of the neural network CMAC in order to control the under-actuated robot RABBIT. This control strategy is very easy to implement on-line and robust. The first result of the experimental validation is presented.

1 Introduction

The design and the control of bipedal robots are one of the more challenging topics in the field of robotics and were treated by a great number of researchers and engineers since many years. The potential applications of this research are very important in the medium and long term. Indeed this can lead initially to a better comprehension of the human locomotion, what can be very helpful for the design of more efficient orthosis. Secondly, the bipedal humanoid robots are intended to replace the human for interventions in hostile environments or to help him in the daily tasks. In addition to the problems related to autonomy and decision of such humanoid robots, the essential locomotion task is still today a big challenge. If it is true that some prototypes were constructed and prove the feasibility of such robots, of which most remarkable is undoubtedly today the robots Asimo [1] and HRP-2P [2], the performances of these are still far from equalizing the human from the point of view of the dynamic locomotion process. The design of new control laws making it possible to ensure real-time control for real dynamic walking in unknown environments is thus today fundamental.

Most of the time, the control of bipedal walking robots is carried out with methods based on tracking of temporal reference trajectories generally associated with a control of the Zero Moment Point (ZMP) [3]. The torque applied at each joint is thus computed by using the difference between the

desired and real trajectories. However, this kind of approaches involves some difficulties with regard to the autonomy of the robot and are today limited on the one hand with regard to the complexity of calculations required to generate full dynamical motions and on the other hand with regard to the structure of robots (unusable for robots without feet).

In this paper, we present a new control strategy based on the use of neural network CMAC within the framework of the control of an under-actuated robot: RABBIT [4, 5]. The main characteristic of RABBIT is that it has no foot. Consequently, its walking gait is completely dynamic and the contacting zone between foot and ground is reduced to a point of contact. The neural network which is used is the Cerebellar Model Controller Articulation (CMAC) [6, 7]. This approach was imagined by Albus starting from the studies on the human cerebellum. Despite its biological relevance, its main interest is the reduction of the training and computing times in comparison to other neural networks [8]. This is of course a considerable advantage for real-time control. Because of these characteristics, the CMAC is thus a neural network well adapted for the control of complex systems with a lot of inputs and outputs and has already been the subject of some researches in the field of the control of biped robots [9–11]. However, in our case, the trajectories memorized by the CMAC, by using supervised learning, are functions of the geometrical pattern of the robot and independent of time, what allows to increase the robustness of the control strategy.

This paper is organized as follows. Section 2 briefly presents the characteristics of the under-actuated robot and our simulation platform. In Sect. 3, the control strategies are described. In Sect. 4, the first results of the experimentation are shown. Conclusions and further developments are finally given in Sect. 5.

2 Model of the Bipedal Robot: RABBIT

The robot RABBIT is an experimentation of seven french laboratories (IRCCYN Nantes, LAG Grenoble, LMS Poitiers, LVR Bourges, LGIPM Metz, LRV Versailles, LIRMM Montpellier) concerning the control of a biped robot for walking and running within the framework of a CNRS project ROBEA [12]. This robot is composed of two legs and a trunk and has no foot as shown on Fig. 1. The joints are located at the hip and at the knee. This robot has four actuators: one for each knee, one for each hip. The characteristics (masses and lengths of the limbs) of RABBIT are summarized in Table 1. Motions are included in the sagittal plane by using a radial bar link fixed at a central column that allows to guide the direction of progression of the robot around a circle. This robot represents the minimal system able to generate walking and running gaits. Since the contact between the robot and the ground is just one point (passive dof), the robot is under-actuated during the single support phase: there are only two actuators (at the knee and at the hip of the

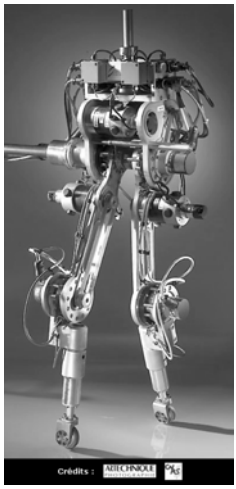


Fig. 1. Prototype RABBIT

Table 1. Masses and Lengths of the Limbs of the Robot

Limb	Weight (Kg)	Length (m)
Trunk	12	0.20
Thigh	6.8	0.4
Shin	3.2	0.4

contacting leg) to control three parameters (vertical and horizontal position of the platform and pitch angle) explicitly. The numerical model of the robot previously described was designed with the software ADAMS[a]. This software, from the modeling of the mechanical system (masses and geometry of the segments) is able to simulate the dynamic behavior of this system and namely to calculate the absolute motions of the platform and the relative motions of the limbs when torques are applied on the joints by virtual actuators.

3 Control Strategy

This control strategy is based on two stages. The first one uses a set of pragmatic rules making it possible to stabilize the pitch angle of the trunk and to generate the leg movements [13, 14]. This simple control strategy, in the absence of disturbances, allows us to produce the dynamic walking of the bipedal robot without reference trajectories and with an average velocity included in $[0\,\mathrm{m/s}; 1\,\mathrm{m/s}]$. The interest of this method resides on the fact that on the one hand the intrinsic dynamics of the system are exploited by using a succession

[a]ADAMS is a product of MSC software

of active and passive phases and on the other hand that the control strategy is very easy to implement on-line. But the use of passive phases implies that this control strategy is very sensitive to the external disturbances that could occur during the progression of the robot. Consequently, we propose to use a neural network allowing to increase the robustness of our control strategy. In fact, in the first stage, a set of articular trajectories of the swing leg are learned by the CMAC. And in the second stage, we use these neural networks to generate the trajectories.

3.1 Learning During a Walking Gait

Figure 2 shows the method used during this training phase. During the learning phase, we use pragmatic rules to control the robot. Furthermore, the trajectories of the swing leg (in terms of joint positions and velocities) are learned with several "single-input/single-output" CMAC neural networks. Indeed, two CMAC are necessary to memorize the joint angles q_{i1} and q_{i2} and two other CMAC for angular velocities $\dot{q}_{i1}$ and $\dot{q}_{i2}$. q_{i1} and q_{i2} are respectively the measured angles at the hip and the knee of the leg i; $\dot{q}_{i1}$ and $\dot{q}_{i2}$ are respectively the measured angular velocities at the hip and the knee of the leg i (see Fig. 2).

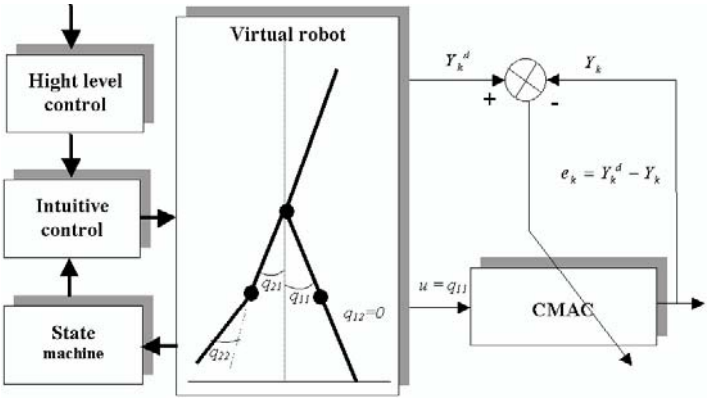


Fig. 2. Principle of the control strategy used during the training of one CMAC neural network ($u = q_{11}$)

When leg 1 is in support ($q_{12} = 0$), the angle q_{11} is applied to the input of each CMAC ($u = q_{11}$) and when leg 2 is in support ($q_{22} = 0$), this is the angle q_{21} which is applied to the input of each CMAC ($u = q_{21}$). Consequently, the trajectories learned by the neural networks are not function of time but function of the geometrical pattern of the robot. Furthermore, we consider that the trajectories of each leg in swing phase are identicals. This makes it possible on the one hand to divide by two the number of CMAC and on

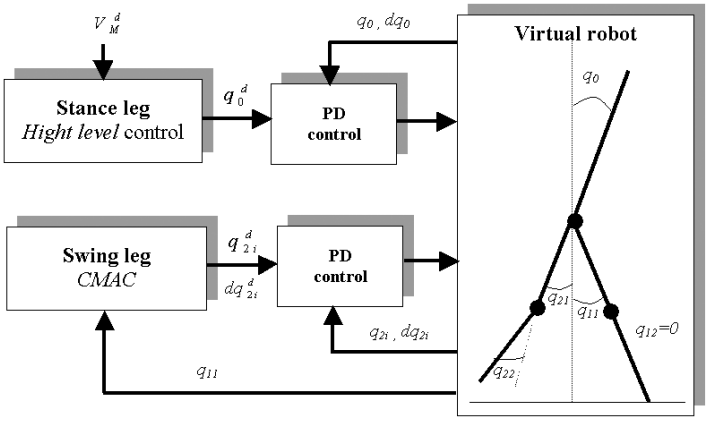


Fig. 3. Principle of the control strategy based on the use of the CMAC neural networks, when the leg 1 is in the stance phase

the other hand to reduce the training time. The weights of each CMAC k ($k = 1, \ldots, 4$) are updated by using the error between the desired output of the CMAC Y_k^d and the computed output of the CMAC Y_k.

3.2 Walking Gait Generation with the CMAC

Figure 3 describes the control strategy used after the phase of training. Firstly, the CMAC networks are used to generate the joint trajectories of the swing leg starting from the geometrical configuration of the stance leg. A PD control makes it possible to ensure the tracking of these trajectories (equations 1 and 2). q_{i1}^d and $\dot{q}_{i1}^d$ are respectively the reference trajectories (position and velocity) of the swing leg generated by the CMAC.

$$T_{hip}^{sw} = K_{hip}^p(q_{i1}^d - q_{i1}) + K_{hip}^v(\dot{q}_{i1}^d - \dot{q}_{i1}) \tag{1}$$

$$T_{knee}^{sw} = K_{knee}^p(q_{i2}^d - q_{i2}) + K_{knee}^v(\dot{q}_{i2}^d - \dot{q}_{i2}) \tag{2}$$

Secondly, the knee of the stance leg is locked (equation 3) and the torque applied to the hip allows to control the pitch angle of the trunk (equation 4). q_0 and $\dot{q}_0$ are respectively the measured absolute angle and angular velocity of the trunk. q_0^d is the desired pitch angle.

$$T_{knee}^{st} = -K_{knee}^p q_{i2} - K_{knee}^v \dot{q}_{i2} \tag{3}$$

$$T_{hip}^{st} = K_{trunk}^p(q_0^d - q_0) - K_{trunk}^v \dot{q}_0 \tag{4}$$

4 Experimentation

The first experiments consisted in validating the approach used in simulation. The training of the CMAC was carried out in simulation for an average desired

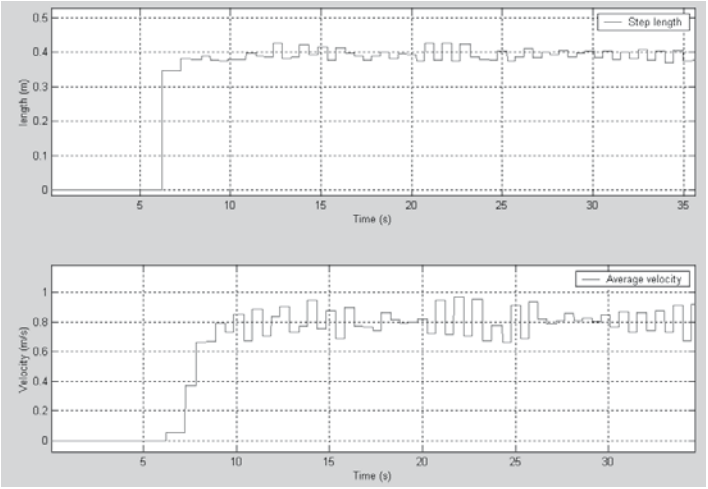


Fig. 4. Step length and average velocity of the real robot RABBIT during approximately 50 steps

velocity of 0.6 m/s and a step length of 0.39m. When the training is finished, these neural networks are used to control the real robot RABBIT.

After an initialization phase of the robot, it is possible to initiate the walking by pushing the robot forwards. This force is necessary to break static balance in which the robot is initially located in position of stop. After some steps, the average velocity converges towards a value of 0.8 m/s. The steps carried out seem natural and relatively regular. To stop the robot, it is necessary to apply a force on the trunk of the robot in the opposed direction of the progression. This force causes a loss of kinetic energy and the robot is then immobilized in phase of double support.

The Fig. 4 shows the evolution of the average velocity and the step length during approximately 50 steps. It should be noted that there is a difference between the average velocity of the left leg and the right leg. This average velocity V_M is calculated with equation (5) where L_{step} is the distance between the two feet at the moment of double impact and t_{step} is the duration of the step (from takeoff to landing of the same leg).

$$V_M = \frac{L_{step}}{t_{step}} \tag{5}$$

In fact, the bar measuring only 1.50 m, its length is insufficient to consider that the robot advances in the sagittal plane. Consequently, the movements of the inner leg are faster than the movements of the outer leg.

Figure 5 shows the evolution of the angles and angular velocities at the hip and the knee during 5 steps. It should be noted that the movements of the legs are really dynamic.

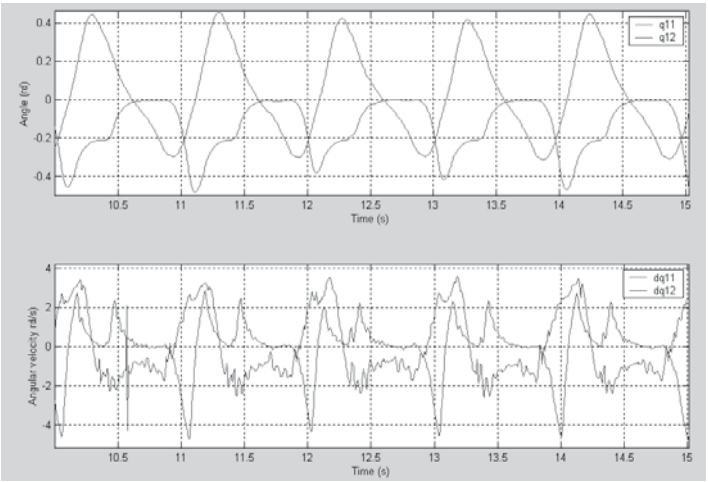


Fig. 5. Measured angles and angular velocities at the hip and the knee during 5 steps of the real robot

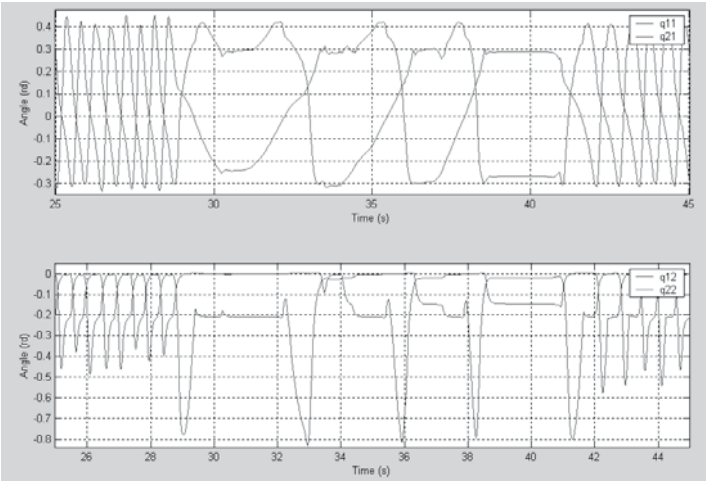


Fig. 6. Reversed walking of the real robot RABBIT

Furthermore, this control strategy is very robust since the movements of the legs are reversible. Figure 6 shows the evolution of the articular angles when the robot is pushed in the rear by a man. We can observe that the articular movements are reversed. Consequently, the robot does not fall but can walk in rear. Then the robot is able to walk fowards again. This is very interesting because the control strategy allows us to increase the robustness with regard to the perturbation forces.

5 Conclusion

In this paper, we proposed a new robust control strategy. This one is based on two stages. The first one consists to carry out the learning by using the neural network CMAC in simulation. The second one consists to use the CMAC to control the virtual or the real robot. Furthermore, the movements of the leg are reversible. Consequently, the walking of the robot is very robust. Future work will focus on the generation of the fast walking gait and the running gait.

References

1. Y. Sakagami et al. The intelligent ASIMO: system overview and integration. *Proc. IEEE Conf. on Robotics and Automation* (2002) 2478–2483.
2. F. Kanehiro et al. The first humanoid robot that has the same size as a human and that can lie down et get up. *Proc. IEEE Conf. on Robotics and Automation* (2003) 1633–1639.
3. M. Vukobratovic, B. Bocovac, D. Surla, D. Stokic. Biped locomotion. *Scientific fundamentals of robotics* **vol 7** Spinger-Verlag, (1990).
4. C. Chevallereau, P. Sardain. Design and actuation of a 4 axes biped robot for walking and running. *Proc. IEEE Conf. on Robotics and Automation* (2000) 3365–3370.
5. *http://robot-rabbit.lag.ensieg.inpg.fr/*
6. J.S. Albus. A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *Trans. ASME* (1975) 220–227.
7. J.S. Albus. Data storage in the cerebellar model articulation controller. *Journal of Dynamic Systems, Measurement and Control* (1975) 228–233.
8. W.T. Miller, F.H. Glanz, and L.G. Kraft. CMAC: an associative neural network alternative to backpropagation. *IEEE Proceedings* **(78)** (1990) 1561–1567.
9. A.L. Kun, W.T. Miller. Adaptive Dynamic Balance of A Biped Robot Using Neural Networks. *Proc. IEEE Conf. on Robotics and Automation* (1996) 240–245.
10. A. Brenbrahim, J. Franklin. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems* **22** (1997) 283–302.
11. J. Hu, J. Pratt, G. Pratt. Stable adaptative control of a bipedal walking robot with CMAC neural networks. *Proc. IEEE Conf. on Robotics and Automation* (1999) 1050–1056.
12. *http://www.laas.fr/robea/*
13. C. Sabourin, O. Bruneau, J.-G. Fontaine. Pragmatic rules for real-time control of the dynamic walking of an under-actuated biped robot *Proc. IEEE Conf. on Robotics and Automation* (2004) 4216–4221.
14. C. Sabourin, O. Bruneau, J.-G. Fontaine. Start, stop and transition of velocities on an underactuated bipedal robot without reference trajectories. *International Journal of Humanoid Robotics* Vol. 1, No. 2 (June 2004) 349–374.

Dynamic Stabilization of an Under-Actuated Robot Using Inertia of the Transfer Leg

A. David, O. Bruneau, and J.G. Fontaine

Laboratoire Vision et Robotique, Bourges, France
anthony.david@ensi-bourges.fr

Abstract. This paper presents the first theoretical tools to control an under-actuated robot by taking into account its intrinsic dynamics and by directly modifying the dynamic effects generated by the robot if the measured velocity of the robot is not the desired one.

1 Introduction

When a human walks quickly, he actively uses its own dynamic effects to ensure his stability and his propulsion. Actually some works use these dynamic effects to generate fast walking gaits [1, 2] and use pragmatic rules based on a qualitative study of the human walk. Our global aim is to produce a more adaptive and universal approach based on a study of the dynamic equations.

In this paper, we develop the first theoretical tools necessary to control an under-actuated robot by taking into account its intrinsic dynamics and by directly modifying the dynamic effects generated by itself to stabilize it. We show the approach's viability by the correction of the moving velocity's robot with the inertial effects generated by the transfer leg during the single support phase.

Our results are validate on a numeric simulator of the robot RABBIT (see Fig. 1) in two dimensions. The parameters used to describe the configuration of RABBIT are given in Fig. 2. G_p, the trunk's centre of gravity, is chosen as the reference point.

The organization of this paper is as follows: in part two, we present a fundamental notion: the dynamic stability, and we define a criterion to represent it. Then, in part three, we calculate the analytic expression of this criterion. In part four, we describe the strategy, based on this criterion, to dynamically stabilize the system, and, in part five, we interpret the results obtained.

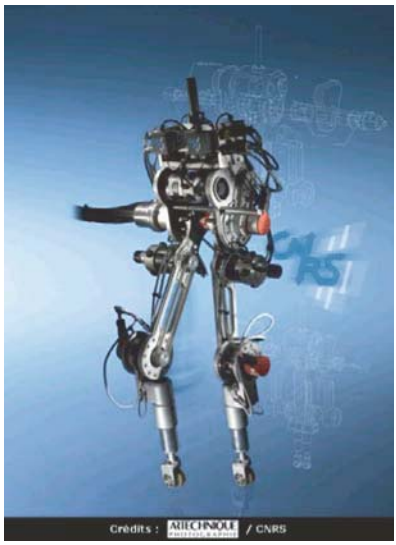


Fig. 1. RABBIT

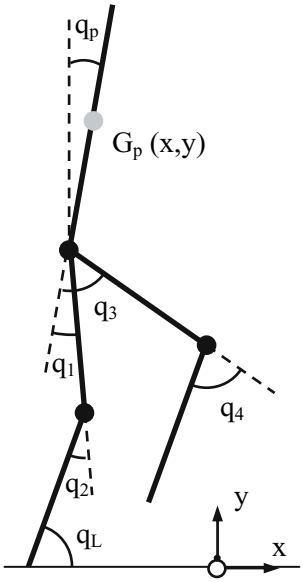


Fig. 2. Model of RABBIT

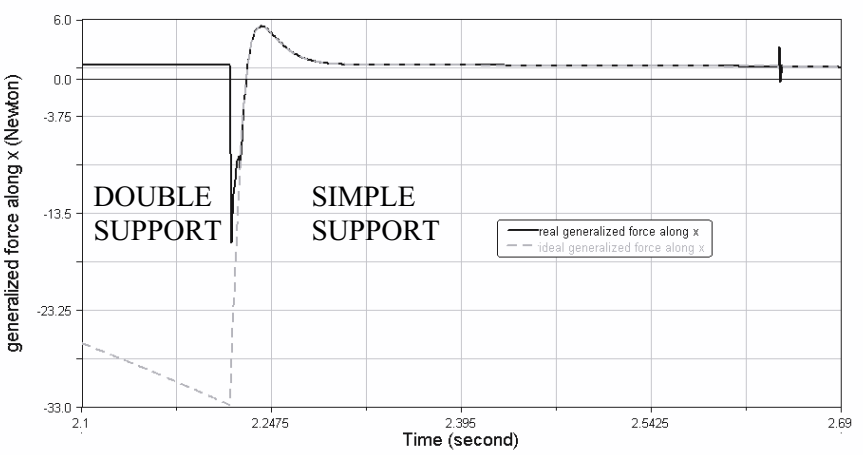


Fig. 3. Evolution of the x-component of real and ideal generalized forces

2 Dynamic Stability

One aspect of the walking robots' control having a dynamic behaviour is the dynamic stability's characterization by a suitable criterion. Vukobratovic [3] proposed the ZMP (Zero Moment Point). Even if it takes into account the dynamic effects generated by the system, this criterion measures only a distance between the actual configuration and the limit of the static stability. Furthermore, it does not measure, but detects, the static unstability. Goswami [4] proposed the FRI (Foot Rotation Indicator). As the ZMP, it measures a static stability, but also the static unstability.

In order to exceed the limitations of these criterions, we propose a new criterion to characterize the dynamic stability and unstability. This new criterion is derived of the dynamic stability criterion of VGS (Virtual Generalized Stabilizer) defined by Foret [5]. Contrary to the VGS, this new criterion quantifies exactly, without giving priority to some dynamical effects, the acceleration to bring to the system to ensure its desired speed.

We start with the dynamic walking definition. During rapid dynamic walking, the system's real position (p_p), velocity (ν_p) and acceleration (a_p) are composed of a controlled part, coming from joint position (q), velocity ($\dot{q}$) and acceleration ($\ddot{q}$) via the leg's jacobian matrix ($J, \dot{J}$) and a part coming from inertial effects. This last part is composed of two parts: a "desired and no directly controlled" part (p_d, ν_d, a_d) and a "no desired and no controlled" part (p_e, ν_e, a_e). This "no desired and no controlled" part is responsible for the system's dynamic stability loss. The system's real position, velocity and acceleration, at the reference point, is written as follows:

$$\begin{aligned} p_p &= f(q) \ + p_d + p_e = p_s + p_e \\ \nu_p &= J\dot{q} \ + \nu_d + \nu_e = \nu_s + \nu_e \\ a_p &= \dot{J}\dot{q} + J\ddot{q} + a_d + a_e = a_s + a_e \end{aligned} \tag{1}$$

where p_s, ν_s and a_s are respectively the desired position, velocity and acceleration for a desired stable gait. A system will be dynamically stable if the "no desired and no controlled" part (p_e, v_e, a_e) is equal to zero. We can consider this part as the error in position (p_e), velocity (v_e) and acceleration (a_e).

We have defined the system's dynamic stability in terms of positions, velocities and accelerations, i.e. in terms of trajectories. Now, we define the system's dynamic stability in terms of generalized forces.

Let $\overline{F}_r$ be the external resulting generalized forces applied to the robot at the reference point and generating the real trajectory defined by p_p, ν_p and a_p. Let $\overline{F}_i$ be the external generalized forces that we must apply to the robot at the reference point to generate the desired trajectory defined by p_s, ν_s and a_s. **The system will be considered as dynamically stable if $\overline{F}_i$, named ideal generalized forces, are equal to $\overline{F}_r$, named real generalized forces**. And, the more important the difference between the two sets of generalized forces is, the more dynamically unstable the robot is. This criterion gives us relevant informations about the global robot stability and about the stability of the robot in one direction or another.

3 Analytical Expression of the Dynamical Stability Criterion

3.1 Dynamic and Control Expression

The lagrangian motion equations, applied at the reference point, are written in two subsystems:

$$\overline{\overline{M}}_p \overline{a}_p + \overline{\overline{H}}_k \overline{a}_k + \overline{C}_p + \overline{G}_p = \overline{\overline{D_{1p}}}^T \overline{F_{c1}} + \overline{\overline{D_{2p}}}^T \overline{F_{c2}} \tag{2}$$

$$\overline{\overline{H}}_k^T \overline{a}_p + \overline{\overline{M}}_k \overline{a}_k + \overline{C}_k + \overline{G}_k = \overline{\overline{D_{1k}}}^T \overline{F_{c1}} + \overline{\overline{D_{2k}}}^T \overline{F_{c2}} + \overline{\Gamma} \tag{3}$$

where $\overline{a}_p = [\ddot{x}, \ddot{y}, \ddot{q}_p]^T$ is the trunk's accelerations vector, $\overline{a}_k = [\ddot{q}_1, \ddot{q}_2, \ddot{q}_3, \ddot{q}_4]^T$ the joint accelerations vector, $\overline{\overline{F_{cj}}}$ $(j = 1, 2)$ the contacting forces applied to the feet and $\overline{\Gamma}$ the vector of joint torques. The index "p" refers to the trunk and the index "k" to the legs. The subsystem (2), coming from the first three motion equations, is related to the trunk. This subsystem is said passive. The subsystem (3), coming from the last four motion equations, is related to the legs. This subsystem is said active.

The chosen joint control is a computed torque method using non linear decoupling of the dynamics:

$$\overline{\Gamma} = \lfloor \overline{\overline{M}}_k - \overline{\overline{H}}_k^T \overline{\overline{M}}_p^{-1} \overline{\overline{H}}_k \rfloor \overline{g}_k + \lfloor \overline{C}_k - \overline{\overline{H}}_k^T \overline{\overline{M}}_p^{-1} \overline{C}_p \rfloor + \lfloor \overline{G}_k - \overline{\overline{H}}_k^T \overline{\overline{M}}_p^{-1} \overline{G}_p \rfloor \quad (4)$$

where:

$$\overline{g}_k = \overline{a}_k^d + \overline{\overline{K}}_v \left(\overline{v}_k^d - \overline{v}_k \right) + \overline{\overline{K}}_p \left(\overline{p}_k^d - \overline{p}_k \right) \quad (5)$$

is the control vector. $\overline{a}_k^d, \overline{\nu}_k^d$ and $\overline{p}_k^d$ are respectively the desired joint acceleration, velocity and position vector, and $\overline{\nu}_k$ and $\overline{p}_k$ are respectively the measured joint velocity and position vector.

3.2 Trunk's Dynamics and Real Generalized Forces Expression

The trunk's dynamics and the real generalized forces are calculated in isolating $\overline{a}_k$ in (5) and in injecting it in (4). Then on the one hand we group the terms depending uniquely on the trunk and on the other hand the terms depending uniquely on the legs and the terms coupled depending on the trunk and the legs. For that, we separate the following terms:

$$\overline{\overline{M}}_p = {}^p\overline{\overline{M}}_p + {}^k\overline{\overline{M}}_p,\ \overline{C}_p = {}^p\overline{C}_p + {}^k\overline{C}_p \text{ with } {}^p\overline{C}_p = \overline{0}, \overline{G}_p = {}^p\overline{G}_p + {}^k\overline{G}_p \quad (6)$$

The first part comes from the trunk effects on itself and the second part comes from the legs' effects on the trunk. We obtain the following equations system:

$$\begin{aligned} {}^p\overline{\overline{M}}_p \overline{a}_p + {}^p\overline{C}_p + {}^p\overline{G}_p = & -\overline{\overline{H}}_k \overline{\overline{M}}_k^{-1} \left[-\overline{C}_k - \overline{G}_k + \overline{\Gamma} \right] - {}^k\overline{C}_p - {}^k\overline{G}_p \\ & - \left[{}^k\overline{\overline{M}}_p - \overline{\overline{H}}_k \overline{\overline{M}}_k^{-1} \overline{\overline{H}}_k^T \right] \overline{a}_p \\ & + \left[\sum_{i=1}^{2} \overline{\overline{D_{ip}}}^T \overline{F_{ci}} - \overline{\overline{H}}_k \overline{\overline{M}}_k^{-1} \left(\overline{\overline{D_{1k}}}^T \overline{F_{c1}} + \overline{\overline{D_{2k}}}^T \overline{F_{c2}} \right) \right] \end{aligned} \quad (7)$$

The left term of (7) represents dynamics of the trunk whereas the right term is the real generalized forces $\overline{F}_r$.

3.3 Ideal Generalized Forces

To calculate the ideal generalized forces $\overline{F_i}$, we use the trunk's dynamics obtained in (7), where we replace the real values $\ddot{x}, \dot{x}, x, \ddot{y}, \dot{y}, y, \ddot{q}_p, \dot{q}_p$ and q_p by their desired expression $\ddot{x}^d, \dot{x}^d, x^d, \ddot{y}^d, \dot{y}^d, y^d, \ddot{q}_p^d, \dot{q}_p^d$ and q_p^d, defining the desired trajectory. The desired trajectory being defined in acceleration terms as well as in velocity and position terms, we replace:

$$\overline{a}_p^d = \left[\ddot{x}^d \ddot{y}^d \ddot{q}_p^d \right]^T \quad \text{by} \quad \overline{a}_p^d = \begin{bmatrix} \ddot{x}^d + K_\nu \left(\dot{x}^d - \dot{x} \right) + K_p \left(x^d - x \right) \\ \ddot{y}^d + K_\nu \left(\dot{y}^d - \dot{y} \right) + K_p \left(y^d - y \right) \\ \ddot{q}_p^d + K_\nu \left(\dot{q}_p^d - \dot{q}_p \right) + K_p \left(q_p^d - q_p \right) \end{bmatrix} \quad (8)$$

what gives us the following ideal generalized forces expression:

$$\overline{F_i} = {}^p\overline{\overline{M}}_p \overline{a}_p^d + {}^p\overline{C}_p^d + {}^p\overline{G}_p^d \quad (9)$$

4 Dynamic Stabilization Strategy

With the dynamic stability criterion's definition in terms of generalized forces: "*the system will be considered as dynamically stable along one direction if the component of* $\overline{F}_i$ *are equal to the component of* $\overline{F}_r$ *along this direction*", we develop a control strategy in order to ensure the robot's dynamic stability along the moving direction ($\vec{x}$). We validate this strategy on the beginning of a walking gait.

4.1 Double Support Phase

During the double support phase, we have an over-actuated robot: we have four motors to control directly the three dof (degrees of freedom) of the trunk. In fact, we must also ensure the chain closing constraint, which adds an equation to be satisfied. We use inverse kinematics in order to calculate the desired joint positions, velocities and accelerations. Each leg is treated independently and inverse kinematics are written from the trunk to the tips of the legs.

4.2 Single Support Phase

During the single support phase, we have an under-actuated robot: we only have the two motors of the stance leg to control directly the three dof of the trunk. We use these motors to directly control the trajectory along the gravity direction ($\vec{y}$) and around the rotation axis of the trunk ($\vec{z}$) via inverse kinematics, written from the tips of the stance leg to the trunk.

We ensure the no-contact between the transfer foot and the ground with the knee joint of the transfer leg via inverse kinematics, written from the knee to the tips of the transfer leg. On the other hand, the transfer foot trajectory along the moving direction ($\vec{x}$) is not controlled. However, we will see the dynamic stabilization strategy involves naturally the moving of the transfer foot along this direction.

With the hip joint of the transfer leg, we ensure the dynamic stability along the moving direction ($\vec{x}$). The idea is to modify, via the torque applied to the hip joint, the inertial effects generated by the transfer leg in order to $\overrightarrow{F}_r$ converge towards $\overrightarrow{F}_i$ along this direction. For that, $\overrightarrow{F}'_i$ component along this direction is calculated by using (9). Then, we calculate the hip joint component control vector. To do this, we replace $\overline{\Gamma}$ in (7) by its analytical expression obtain by (4). This is allows us to directly express $\overline{F}_r$ as a function of the control vector $\overline{g}_k$:

$$\begin{aligned}\overline{F_r} = &\left[\overline{\overline{D_{1p}}}^T\overline{F_{c1}} + \overline{\overline{D_{2p}}}^T\overline{F_{c2}} - \overline{\overline{H}}_k\overline{\overline{M_k^{-1}}}\left(\overline{\overline{D_{1k}}}^T\overline{F_{c1}} + \overline{\overline{D_{2k}}}^T\overline{F_{c2}}\right)\right]\\ &- \left[{}^k\overline{\overline{M}}_p - \overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\overline{\overline{H_k^T}}\right]\overline{a}_p \; - \left[{}^k\overline{C}_p - \overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\overline{\overline{H}}_k^T\overline{\overline{M}}_p^{-1}\overline{C}_p\right]\\ &- \left[{}^k\overline{G}_p - \overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\overline{\overline{H}}_k^T\overline{\overline{M}}_p^{-1}\overline{G}_p\right]\\ &- \overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\left[\overline{\overline{M}}_k - \overline{\overline{H}}_k^T\overline{\overline{M}}_p^{-1}\overline{\overline{H}}_k\right]\overline{g}_k \end{aligned} \tag{10}$$

After that, we use the equation along the moving direction ($\overline{x}$) of (10), where we replace $(\overrightarrow{F}_r)_x$ by the desired value $(\overrightarrow{F_i})_x$ obtained by (9), which gives us:

$$\begin{aligned}\left(\overline{F_i}\right)_x = &\left(\left[\overline{\overline{D_{1p}}}^T\overline{F_{c1}} + \overline{\overline{D_{2p}}}^T\overline{F_{c2}} - \overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\left(\overline{\overline{D_{1k}}}^T\overline{F_{c1}} + \overline{\overline{D_{2k}}}^T\overline{F_{c2}}\right)\right]\right)_x\\ &- \left(\left[{}^k\overline{\overline{M}}_p - \overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\overline{\overline{H}}_k^T\right]\overline{a}_p\right)_x - \left(\left[{}^k\overline{C}_p - \overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\overline{\overline{H}}_k^T\overline{\overline{M}}_p^{-1}\overline{C}_p\right]\right)_x\\ &- \left(\left[{}^k\overline{G}_p - \overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\overline{\overline{H}}_k^T\overline{\overline{M}}_p^{-1}\overline{G}_p\right]\right)_x\\ &- \left(\overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\left[\overline{\overline{M}}_k - \overline{\overline{H}}_k^T\overline{\overline{M}}_p^{-1}\overline{\overline{H}}_k\right]\right)_{x,1} g_1\\ &- \left(\overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\left[\overline{\overline{M}}_k - \overline{\overline{H}}_k^T\overline{\overline{M}}_p^{-1}\overline{\overline{H}}_k\right]\right)_{x,2} g_2\\ &- \left(\overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\left[\overline{\overline{M}}_k - \overline{\overline{H}}_k^T\overline{\overline{M}}_p^{-1}\overline{\overline{H}}_k\right]\right)_{x,3} g_3\\ &- \left(\overline{\overline{H}}_k\overline{\overline{M}}_k^{-1}\left[\overline{\overline{M}}_k - \overline{\overline{H}}_k^T\overline{\overline{M}}_p^{-1}\overline{\overline{H}}_k\right]\right)_{x,4} g_4 \end{aligned} \tag{11}$$

where the components vector control g_1 (stance hip), g_2 (stance leg) and g_4 (transfer knee) had been calculated before with inverse kinematics. The equation (11) allows us to calculate the control vector's component g_3 (transfer hip) to ensure the dynamic stability along the moving direction.

5 Simulation Results

In order to validate this strategy, we simulate the beginning of a walking gait. The system starts in double support phase with a velocity equal to $0,0\,\mathrm{m/s}$ and increases its velocity until $0,2\,\mathrm{m/s}$. To define the transition velocity we use a polynomial interpolation of the fifth order.

We see, Fig. 3, the dynamic stability of the system in double support phase is not ensured. Indeed, inverse kinematics does not take into account

Fig. 4. First Step

the dynamic effects. Nevertheless, as these effects is few important, the desired trajectory is approximately followed. In simple support phase, the system becomes dynamically stable. The desired trajectory is perfectly followed. We see, Fig. 4, the strategy involves naturally the moving of the transfer foot along the moving direction.

6 Conclusions and Future Work

In this paper, we proposed a new approach to control an under-actuated robot by taking into account its intrinsic dynamics and by directly modifying the dynamic effects generated by the robot if the measured velocity of the robot is not the desired one. Firstly, we developed and validated the first theoretical tools necessary for this control. After the description of a dynamic walking gait and the definition of the dynamic stability, we proposed a new criterion characterizing this stability. Finally, we showed the approach 's viability by the correction of the robot moving speed with the inertial effects generated by the swing leg during the single support phase.

Our future work, in a first time, will consist in stabilizing the three degree of freedom of the trunk with the two legs during the single support phase as well as during the double support phase.

References

1. Pratt, J.E., Pratt, G.A. (1998) Intuitive Control of a Planar Bipedal Walking Robot. In: IEEE Proc. of Int. Conf. on Robotics and Automation
2. Sabourin, C., Bruneau, O., Fontaine, J.G. (2004) Pragmatic rules for real-time control of the dynamic walking of an under-actuated biped robot. In: IEEE Proc. of Int. Conf. on Robotics and Automation
3. Vukobratovic, M., Borovac, B., Surla, D., Stokic, D. (1990) Biped locomotion. In: Scientific fundamentals of robotics (ed. Springer-Verlag), Vol. 7, 1990.
4. Goswami, A., Kallem, V. (2004) Rate of change of angular momentum and balance maintenance of biped robots. In: IEEE Proc. of Int. Conf. on Robotics and Automation
5. Foret, J., Bruneau, O., Fontaine, J.G. (2003) Unified Approach for m-Stability Analysis and Control of Legged Robots. In: IEEE Proc. of Int. Conf. on Intelligent Robots and Systems

Kinematic and Dynamic Analyses of a Pantograph-Leg for a Biped Walking Machine

E. Ottaviano, M. Ceccarelli, and C. Tavolieri

LARM: Laboratory of Robotics and Mechatronics – University of Cassino, via Di Biasio 43, 03043 Cassino (Fr), Italy
ottaviano/ceccarelli@unicas.it

Abstract. In this paper Kinematic and Dynamic analyses are presented for a 1-DOF (Degree-of-Freedom) pantograph-leg. The main objectives that have been considered to achieve a walking operation are modularity, compactness, light weight and reduced number of DOFs. A preliminary version of a low-cost biped machine, which is capable of a straight walking with only one actuator, has been built at LARM: Laboratory of Robotics and Mechatronics in Cassino. Experimental tests have been carried out to validate the proposed model and to test the practical feasibility of the leg design.

1 Introduction

Several walking machines and robots have been conceived, designed and built in the last 30 years with the aim to open new fields of application [1, 2]. Compared to wheeled vehicles, walking machines show the advantage that they can act in highly unstructured terrain. Legged robots cross obstacles more easily; they depend less on the surface conditions and quality and, in general, exhibit better adaptability [3]. In the past, at LARM biped walking robots EP-WaR and EP-WaR II with pneumatic actuation have been designed and built with a suitable pantograph mechanism for the legs [4–6]. Pantograph mechanisms are frequently applied to robot arms and legs with 1 or 2 DOFs. They have suitable static and dynamic characteristics because they are closed-loop mechanisms and driven by actuators fixed on frame links, [4–6].

In this paper numerical and experimental analyses are presented for a 1-DOF leg mechanism for walking machines. A suitable model of the leg has been analyzed and simulations of Kinematics and Dynamics have been carried out. A preliminary version of the low-cost biped machine with only one actuator has been built in order to validate the numerical results.

2 Design of a 1-DOF Pantograph-Leg with a Fully-Rotative Actuating Mechanism

Basic considerations for a leg design can be outlined as follows: the leg should generate an approximately straight-line trajectory for the foot with respect to the body [7–11]; the leg should have an easy mechanical design; if it is specifically required it should posses the minimum number of DOFs to ensure the motion capability. Among many different structures, we have considered a leg mechanism that is based on a Chebyshev mechanical design [12, 13]. The proposed leg mechanism is schematically shown in Fig. 1. In particular, its mechanical design is based on the use of a Chebyshev four-bar-linkage, a five-bar linkage, and a pantograph mechanism. For such mechanism, the leg motion can be performed by using 1 DOF. The leg has been designed by considering compactness, modularity, light weight, reduced number of DOF as basic objectives to achieve the walking operation. Furthermore, the mechanical design has been conceived to built a low cost prototype, as shown in Fig. 1. Numerical and experimental results in [15] show that good kinematic features can be obtained when points C and P in Fig. 1 are not coincident. In particular, it has been shown in [14, 15] that better features can be obtained if the transmission angles γ_1 and γ_2 have suitable large values.

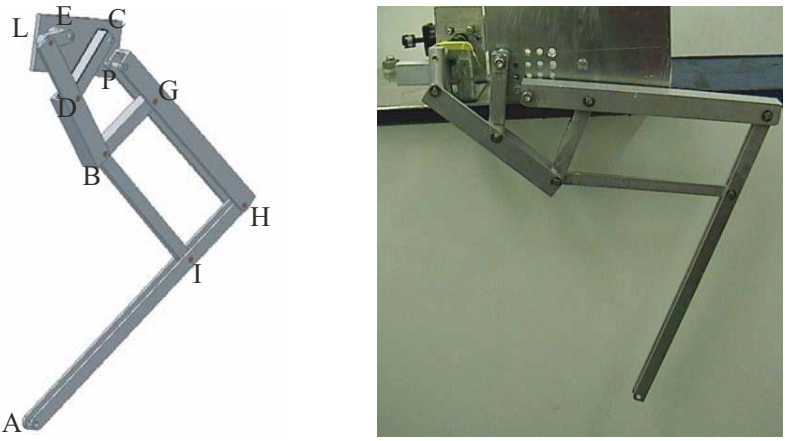


Fig. 1. The 1-DOF pantograph-leg: (**a**) a scheme; (**b**) the built prototype at LARM

3 Kinematic and Dynamic Analyses

Kinematic and Dynamic analyses have been carried out in order to evaluate and simulate the performance and operation of the leg system.

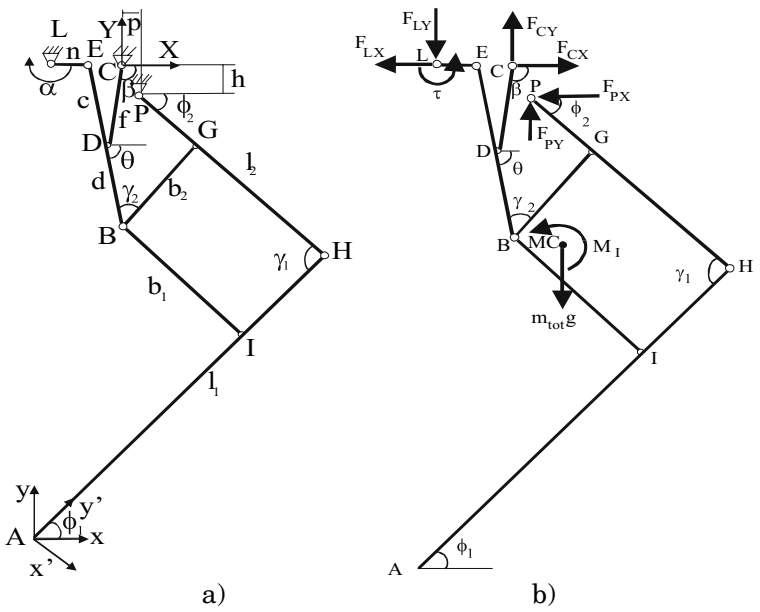


Fig. 2. Schemes for the 1-DOF leg for: (**a**) kinematic analysis; (**b**) dynamic analysis

In particular, the mechanical design is based on the Chebyshev design [12, 13]. Three reference systems have been considered, as shown in Fig. 2. The position of point B with respect to CXY frame can be evaluated as a function of the input crank angle α as

$$X_B = -a + n\,\cos\alpha + (c+f)\,\cos\theta; \quad Y_B = -n\,\mathrm{sen}\,\alpha - (c+f)\,\mathrm{sen}\,\theta \tag{1}$$

where

$$\theta = 2\tan^{-1}\frac{\mathrm{sen}\,\alpha - (\mathrm{sen}^2\alpha + B^2 - D^2)^{1/2}}{B+D} \tag{2}$$

Coefficients B, C and D can be derived by the closure equation of the mechanism. By considering the five-bar linkage CDBGP in Fig. 2, one can write the closure loop equation to obtain φ_1 and φ_2 as

$$\varphi_1 = 2\tan^{-1}\frac{1-\sqrt{1+{k_1}^2-{k_2}^2}}{k_1-k_2}; \quad \varphi_2 = 2\tan^{-1}\frac{1-\sqrt{1+{k_3}^2-{k_4}^2}}{k_3-k_4} \tag{3}$$

where

$$\begin{aligned} k_1 &= \frac{x_B-p}{y_B-h}; k_2 = \frac{{b_1}^2+(y_B-p)^2-(l_2-b_2)^2+(x_B-h)^2}{2b_1\,(y_B-h)} \\ k_3 &= \frac{p-x_B}{y_B-h}; k_4 = \frac{-{b_1}^2+(y_B-p)^2+(l_2-b_2)^2+(x_B-h)^2}{2\,(l_2-b_2)\,(y_B-h)} \end{aligned} \tag{4}$$

Consequently, from Fig. 2a, transmission angles γ_1 and γ_2 can be evaluated as $\gamma_1 = \varphi_1 + \varphi_2$ and $\gamma_2 = \pi - \theta - \varphi_1$. The position of A can be given as

$$\begin{aligned} X_A &= X_B + b_2 \cos\varphi_2 - (l_1 - b_1)\cos\,\varphi_1 \\ Y_A &= Y_B - b_2\,\mathrm{sen}\,\varphi_2 - (l_1 - b_1)\,\mathrm{sen}\varphi_1 \end{aligned} \tag{5}$$

The position of C with respect Axy reference frame can be written as

$$x_C = l_1\cos\varphi_1 - l_2\cos\varphi_2 - p; \quad y_C = l_1\,\mathrm{sen}\,\varphi_1 + l_2\,\mathrm{sen}\,\varphi_2 - h \tag{6}$$

Acceleration of point A, with respect to Axy frame can be obtained as

$$\begin{aligned} a_{AX} &= a_{BX} - b_2\left(\ddot{\varphi}_2\,\mathrm{sen}\,\varphi_2 + \dot{\varphi}_2^2\cos\,\varphi_2\right) \\ &\quad - (l_1 - b_1)\left(\ddot{\varphi}_1\,\mathrm{sen}\,\varphi_1 + \dot{\varphi}_1^2\cos\,\varphi_1\right) \\ a_{AY} &= a_{BY} - b_2\left(\ddot{\varphi}_2\cos\,\varphi_2 - \dot{\varphi}_2^2\,\sin\,\varphi_2\right) \\ &\quad - (l_1 - b_1)\left(\ddot{\varphi}_1\cos\,\varphi_1 + \dot{\varphi}_1^2\,\sin\,\varphi_1\right) \end{aligned} \tag{7}$$

The walking performances can be also evaluated from dynamic viewpoint by using the Newton-Euler method to obtain the equations to derive the actuating torque $\boldsymbol{\tau}$. In the following analysis 3 assumptions have been made [14]: 1) concentrated masses; 2) no friction forces; 3) no elastic deformations. Expressions of the velocity and acceleration are evaluated by using (1) to (7) for the center of masses for each link. A suitable model for the dynamic analysis is shown in Fig. 2b.

The force acting on the i-th link of the leg can be evaluated in the form

$$\sum_{i=1}^{7}(F_{exi} + F_{Ixi}) = 0; \quad \sum_{i=1}^{7}(F_{eyi} + F_{Iyi}) = 0; \quad \sum_{i=1}^{7}(M_{ei} + M_{Ii}) = 0 \tag{8}$$

in which $\mathbf{F}_e$ and $\mathbf{F}_i$ represent respectively the acting and inertia forces and $\mathbf{M}_e$ and $\mathbf{M}_i$ represent respectively acting and inertia moments.

The equation's system in (8) can be rewritten in the form

$$A\mathbf{X} = \mathbf{F} \tag{9}$$

where A is a (21×21) matrix, $\mathbf{X}$ is the vector containing 21 unknown forces and torques. $\mathbf{F}$ is the vector of 21 known forces and torques.

4 Numerical Evaluation of the Mechanical Characteristics

Kinematics and Dynamics have been considered for numerical simulations with data of Table 1. In particular, results have been obtained without considering the leg's interaction with the ground. It has been shown in [14, 15] that the best motion characteristics for the leg can be obtained when p and h in Fig. 2a are equal to 20 mm and 30 mm.

Table 1. Design parameters in mm for the model of Fig. 2a

n = 25	d = 62.5	c = f = 62.5	a = 50
$l_1 = 300$	$l_2 = 200$	$b_1 = 75$	$b_2 = 150$

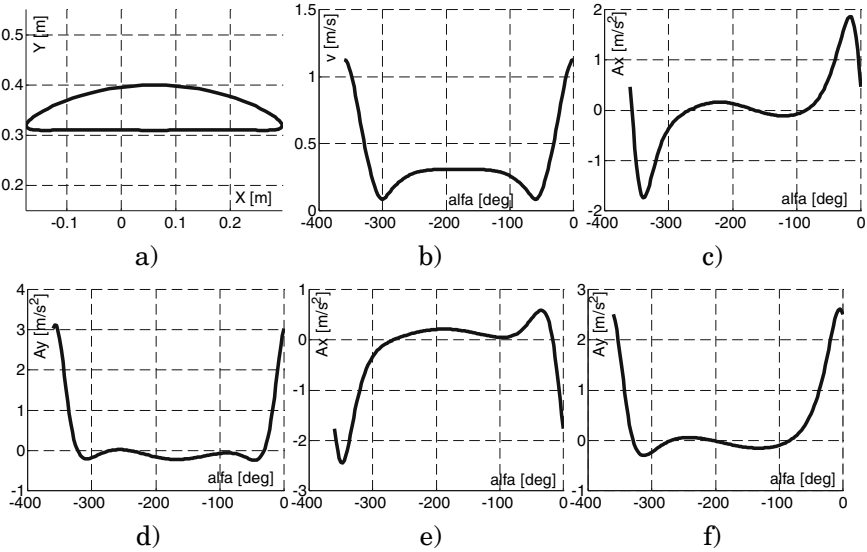


Fig. 3. Simulation for the walking characteristics of the 1-DOF leg ($p = 20\,\text{mm}$, $h = -30\,\text{mm}$): (**a**) Point C trajectory; (**b**) Point A velocity; (**c**) Acceleration a_{AX}; (**d**) Acceleration a_{AY}; (**e**) Acceleration a_{Ax}; (**f**) Acceleration a_{Ay}

Figure 3 shows results of the numerical simulation that has been carried out with $p = 20$ and $h = -30$ and the angular velocity ω of the input crank is chosen as a constant value and equal to 2.3 rad/s. Figure 4 shows transmission angles and the actuating torque. Simulations have been also carried out by removing the hypothesis of input constant velocity. Indeed, the angular velocity ω can be considered as

$$\omega = -\cos(\nu) \tag{10}$$

5 A Prototype of A Low-Cost Biped Machine

A prototype of low-cost biped walking machine has been built as based on the proposed 1-DOF leg at LARM in Cassino. It is shown in Fig. 5. Experimental tests have been carried out with the biped machine, it has been tested by considering two operation modes, [14]: "walker", which has been obtained by

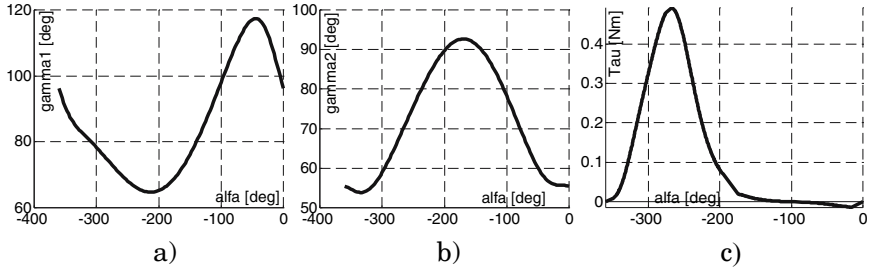


Fig. 4. Simulation for the 1-DOF leg ($p = 20\,\text{mm}$ and $h = 30\,\text{mm}$): (**a**) Transmission angle γ_1; (**b**) Transmission angle γ_2; (**c**) actuating torque $\boldsymbol{\tau}$

a counter-clockwise motion of the input crank; and "ostrich", which has been obtained by a clockwise motion of the input crank. In particular, in this paper a comparison between numerical and experimental results has been presented by referring to the operation mode "ostrich".

6 Experimental Tests

A scheme of the measuring system that has been settled up, is shown in Fig. 5b. It is composed of commercial measuring sensors and LabView software [16] with NI 6024E Acquisition Card [17]. Referring to Fig. 5 one biaxial accelerometer Ac, [18], has been installed at point A. It gives the possibility to measure and monitor the acceleration at the extremity of the leg. Numerical

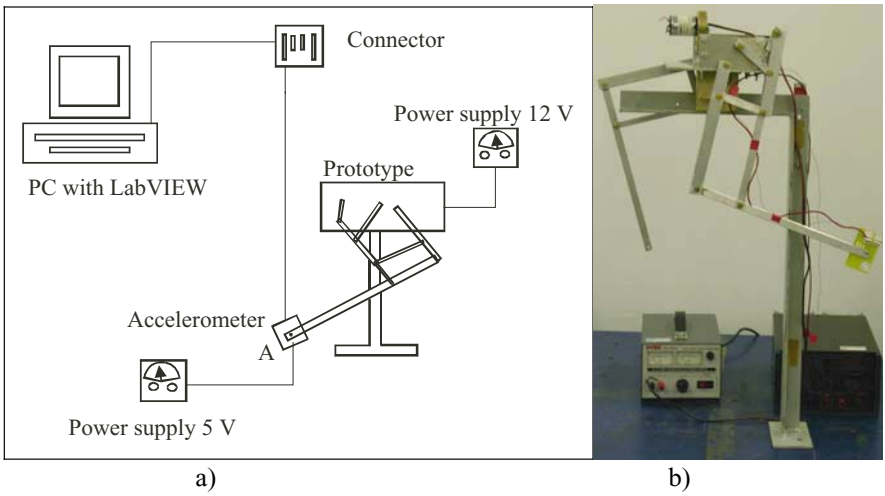


Fig. 5. The experimental test bed at LARM: (**a**) a scheme; (**b**) the sensored prototype of low-cost biped machine

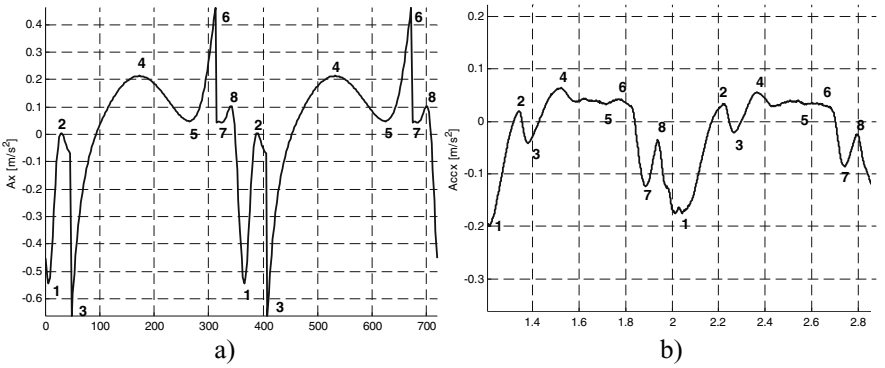


Fig. 6. A comparison between numerical and experimental results for the prototype of Fig. 5 for the operation mode "ostrich": (**a**) simulated acceleration along x′-axis; (**b**) measured acceleration along x′-axis

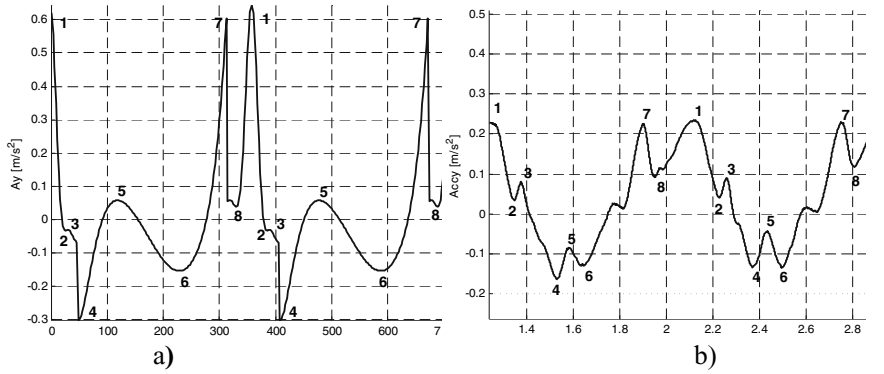


Fig. 7. A comparison between numerical and experimental results for the prototype of Fig. 5 for the operation mode "ostrich": (**a**) simulated acceleration along y′-axis; (**b**) measured acceleration along y′-axis

results are shown in Figs. 6 and 7 in which the acceleration components a_{Ax} and a_{Ay} are expressed in the AX_mY_m reference frame, as shown in Fig. 2a.

7 Conclusion

In this paper a 1-DOF pantograph-leg is presented. Kinematics and Dynamics have been solved to obtain suitable simulation of its operation. A low-cost biped machine has been built at LARM in Cassino by using the proposed 1-DOF leg. First experimental results has been presented.

References

1. Rosheim, M.E. (1994) Robot Evaluation, Wiley, New York
2. Chernousko, F.L. (1990) On the Mechanics of a Climbing Robot, Jnl. of Mechatronics Systems Engineering, Vol. 1, pp. 219–224
3. Morecki, A., Waldron, K.J. (1997) Human and Machine Locomotion, Springer, New York
4. LARM web page: http://webuser.unicas.it/weblarm/larmindex.htm
5. Figliolini, G., Ceccarelli, M. (1996) A Mechanical Design of an Electropneumatic Anthropomorphic Walking Robot, 11th CISM-IFToMM Symp. on Theory and Practice of Robots and Manip., Springer-Verlag, Wien, pp. 189–196
6. Figliolini, G., Ceccarelli, M. (2003) Novel binary pneumatic actuation for EP-WAR3 biped robot. 6th Int. Conf. on Climbing and Walking Robots, CLAWAR2003, Eds. Muscato and Longo, Professional Engineering Publishing, London (UK), pp. 853–860
7. Berns, K. Web Page: www.fzi.de/ids/WMC/walking_machines_katalog/walking_machines_katalog.html
8. Hirose, S., Kunieda, O. (1991) Generalized Standard Foot Trajectory for a Quadruped Walking Vehicle, The Int. Jnl. of Rob. Research, Vol.10, n.1, pp. 3–12
9. Yoneda, K. (2001) Design of Non-Bio-Mimetic Walker with Fewer Actuators. CLAWAR 2001, Karlsruhe, pp. 115–126
10. Song, S.M., Lee, J.K., Waldron, K.J. (1987) Motion Study of Two and Three Dimensional Pantograph Mechanisms, Mech. and Mach. Theory
11. Morecki, A. (1995) Biomechanical Modeling of Human Walking, 9th World Congr. on the Theory of Mach. and Mech., Milan, Vol.3, pp. 2400–2403
12. Artobolevsky, II. (1986) Mechanisms in Modern Engineering Design, Vol. 1, Mir Publ., Moscow, 3rd Ed.
13. Lanni, C., Ceccarelli, M., Ottaviano, E., Figliolini, G. (1999) Actuating Mechanisms for Pantograph Legs: Structures and Characteristics. 10th World Congr. on the Theory of Mach. and Mech., Oulu, Vol. 3, pp. 1196–1201
14. Tavolieri, C. (2004) Design, Simulation and Construction of a Biped Walking Robot. Master Thesis, LARM, University Cassino, Cassino
15. Ottaviano, E., Lanni, C., Ceccarelli, M. (2004) Numerical and Experimental Analysis of a Pantograph-Leg with a Fully-Rotative Actuating Mechanism. 11th World Congr. in Mech. and Mach. Science, IFToMM 03, Tianjin, paper Rb31
16. National Instruments, VI Reference Manual, 1993
17. National Instruments, DAQ: 6023E/6024E/6025E User Manual, 2000
18. Analog Devices, Analog Accelerometer ADXL202E, 2000

Humanoid Robot Kinematics Modeling Using Lie Groups

J.M. Pardos and C. Balaguer

Robotics Lab, Department of Systems Engineering and Automation, University Carlos III of Madrid, C/ Butarque 15, Leganés, 28911, Madrid – Spain
jpardos@endesa.es
balaguer@ing.uc3m.es

Abstract. This paper presents new analytical methods, using techniques from the theory of **Lie Groups** and tools from **Computational Geometry**, to develop algorithms for kinematics modeling and path-planning of humanoid robots. We introduce deterministic solutions, geometrically meaningful and numerically stable, based on the **PoE (Product of Exponentials formula)**, which allows us to develop algorithms to solve the humanoid mechanical problem, even for real-time applications. Besides, to solve the path-planning problem, we introduce a close-form solution based on the **FMM (Fast Marching Methods)** used to study interface motion. The first development of this paper is a new approach for the analysis of the humanoid model; this is **the humanoid SKD (Sagital Kinematics Division)**. The second development is the construction of the **new FM3 (Fast Marching Method Modified) algorithm** for the humanoid collision-free WBT (Whole Body Trajectory) calculation. In addition, the third main goal of this work is to build a general and analytical solution for the humanoid kinematics problem (direct and inverse); this is the **new humanoid OSG (One Step to Goal) algorithm**. The works are presented along with computed examples using both simulated and experimental results with the humanoid robot **RH0** at the **University Carlos III of Madrid**.

1 Introduction

The Humanoid mechanic modeling presents a formidable computational challenge, especially for real-time applications, due to the high number of degrees of freedom, complex kinematics models, and balance constraints [7–10]. As the complexity increases, the need for more elegant formulations of the equations of motion becomes a paramount issue. We introduce new closed-form geometric algorithms based on the **FMM** [1] and the **PoE** [2], which allows us to develop elegant solutions to solve the path-planning and kinematics and modeling of humanoid robots.

The development of general-purpose motion generation tools has been based on the use of virtual humanoid robot platforms [7–9]. In a similar way,

our results are illustrated in studies with the 21 DoF **RH0 humanoid robot** (**University Carlos III of Madrid**), using a graphical simulation environment based on the H-Anim (Humanoid Animation) definition, and VRML (Virtual Reality Modeling Language).

The first objective of this paper is to introduce the new approach called **humanoid SKD** (**Sagital Kinematics Division**) for the analysis of the humanoid model. The SKD considers the humanoid to be divided into two separated manipulators, these are, the left and right half-humanoid robots.

The second goal of this paper is to path planning the collision-free WBT (Whole Body Trajectory) for the humanoid, by developing the **new FM3** (**Fast Marching Method Modified**) **algorithm**.

The third aim of this work is to build the **new OSG** (**One Step to Goal**) **algorithm**, for generating motions through the solution of the humanoid inverse kinematics, by imposing balance constraints of the ZMP (Zero Moment Point) trajectory upon the SKD model. The problem is solved in an analytical and complete way with five steps (Orientate, Tilt, Elevate, Lean and Balance), using techniques from the Lie groups [2, 4, 5].

2 Background

The Fast Marching Methods [1] are numerical techniques for analyzing interface motion. They rely on a Eulerian perspective for the view of moving boundaries. They can be applied to find out optimal path planning problems, through solving the Eikonal equation, extracting among all possible solutions, the one that corresponds to the first arrival of information from the initial disturbance.

The homogeneous representation of a rigid motion "g", belongs to the Lie group SE(3) (Especial Euclidean group) [2, 3]. The Lie algebra of SE(3), denoted se(3), can be identified with the matrices $\xi^\wedge$ (twists) A "twist" can be interpreted geometrically using the theory of screws [6]. Charles's theorem proved that any rigid body motion could be produced by a translation along a line followed by a rotation about that line, this is, a screw motion, and the infinitesimal version of a screw motion is a twist.

$$g = e^{\xi^\wedge \theta} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} e^{\omega^\wedge \theta} & (I - e^{\omega^\wedge \theta}) \cdot (\omega \times v) + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix} \; \omega \neq 0$$

$$e^{\omega^\wedge \theta} = I + \omega^\wedge \sin\theta + \omega^{\wedge 2}(1 - \cos\theta);\ \textit{Rodrigues' formula}$$

It is possible to generalize the forward kinematics map for an arbitrary open-chain manipulator by the product of exponentials.

$$g_{st}(\theta) = e^{\xi_1{}^\wedge \theta_1} \cdot e^{\xi_2{}^\wedge \theta_2} \cdots e^{\xi_n{}^\wedge \theta_n} \cdot g_{st}(0)$$

One payoff for the product of exponentials formalism is that it provides an elegant formulation of a set of canonical problems, the Paden-Kahan subproblems [2, 4, 5]. Reducing it into appropriate subproblems can solve the full inverse humanoid kinematics problem.

3 Humanoid Model – RH0 Robot

Because the RH0 has 21 degrees of freedom, there might be many solutions for any motion, and we search for some kind of simplification to solve the problem. Consequently, we introduce the idea of considering the Humanoid to be divided by its sagital plane into two kinematical different mechanisms; this is what we call **Humanoid SKD (Sagital Kinematics Division)** (e.g. Fig. 1). The idea behind the SKD is that we can analyze and control the complete humanoid body, considering separately the left and right body halves, as a resemblance of the two hemispherical locomotion control made by the human brain. This is a divide and conquers approach, which allows us to solve the humanoid problem in a much easier way by solving the problem for two synchronized manipulators.

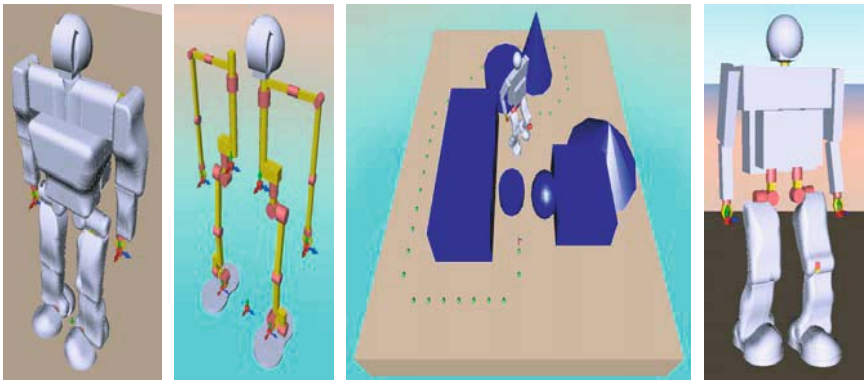
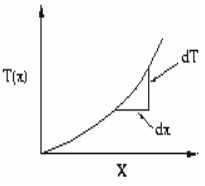


Fig. 1. RH0 Robot – humanoid SKD (Sagital Kinematics Division) approach – FM3 (Fast Marching Method Modified) algorithm for path planning – and humanoid OSG (One Step to Goal) motion algorithm



$$dx = F(dT)$$

$$1 = F\,dT/dx$$

$$|\nabla T|F = 1$$

Optimally Efficient Path Around Obstacles

Fig. 2. One dimension Eikonal Equation example Fast Marching Method, and Efficient Path Planning

4 Whole Body Trajectory (WBT) and Footstep Path Planning

We develop a **new FM3 (Fast Marching Method Modified) Algorithm**, for generating collision-free trajectory motions for the entire humanoid body. The FM3 overcomes the problem of other slow numeric methods [1] (because of the quadratic approach), through the building of a lineal scheme for approximating the Eikonal equation, which is directly resoluble. The FM3 Arrival Function "T" increases in sense of information propagation from the origin to the objective points, for all points inside the configuration space. Graphically, the arrival function in R^3 is the development of a plane interface in motion (kind of accumulative pyramidal fronts). The lineal approached of the FM3 generates a non-differentiable but continuous "T" function, and therefore the explicit construction of the shortest path can come through back propagation following the maximum negative gradient. This method always allows us to find out a cuasi-optimal humanoid WBT (Whole Body Trajectory) path, whatever the nature of the obstacles. An efficient scheme for R^3 is presented hereafter, being "h" the integration step ant "t" the time [1].

$$\max(D_{ij}^{-x}T, D_{ij}^{+x}T, D_{ij}^{-y}T, D_{ij}^{+y}T, D_{ij}^{-z}T, D_{ij}^{+z}T, 0) = \frac{1}{F_{ij}}$$

$$D_{ij}^{\pm\alpha}T \equiv Abs\left(\frac{T(\alpha \pm h, y) - T(\alpha, y)}{h}\right)$$

$$T(x, y, z) = \min \int_{A}^{(x,y,z)} F(x, y, z)d\tau$$

To get **the footstep path planning**, we use the WBT as the main input information, and besides the desired step height, weight and length.

5 Humanoid Kinematics Motion

The idea behind the **new humanoid OSG (One Step to Goal) algorithm** is to develop a general-purpose solution for the humanoid kinematics problem of moving it one-step towards a given goal. The algorithm uses the concept of humanoid SKD subject to the following constraints: keep the balance of the humanoid ZMP (Zero Moment Point) and impose the same position and orientation for the common parts (pelvis, thoracic, cervical) of the left and right half-humanoids.

*A*ny OSG movement is generated into five phases. The first one is to **Orientate** the humanoid body towards the goal. The second is to **Tilt** the ZMP on a stable position over the pillar foot. The third is to **Elevate** the flying foot. The fourth is to **Lean** the flying. The fifth is to **Balance** the humanoid body to leave the ZMP in the center of the convex hull.

Each of the OSG phases considers alternatively the humanoid legs as open chain manipulator with 12 DoF. The first six DoF are NON-PHYSICAL and

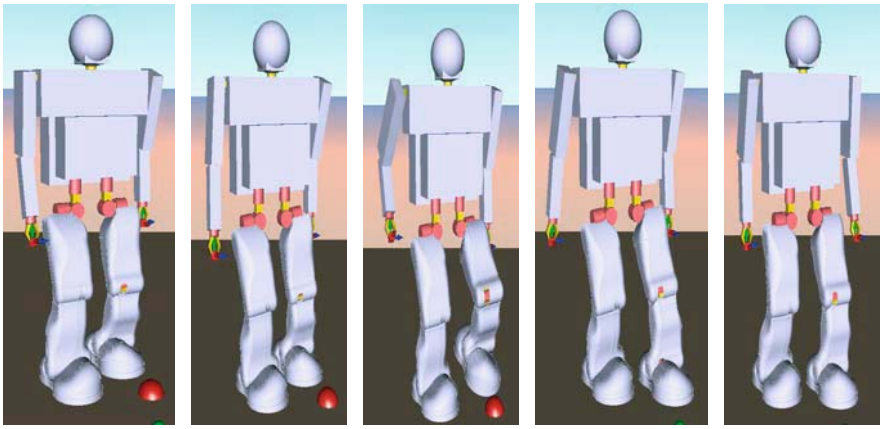


Fig. 3. The five phases for any humanoid OSG (One Step to Goal) Algorithm: 1-Orientate, 2-Tilt, 3-Elevate, 4-Lean and 5-Balance

correspond to the position $(\theta_1, \theta_2, \theta_3)$ and orientation $(\theta_4, \theta_5, \theta_6)$ of the foot. The other DoF are PHYSICAL: θ_7 for the hindfoot, θ_8 for the ankle, θ_9 for the knee, θ_{10} for the hip on the x-axis, θ_{11} for the hip on the z-axis and θ_{12} for the hip on the y-axis. Let "S" be a frame attached to the base system, and "T" be a frame attached to the humanoid ZMP. The reference configuration corresponds to $\theta = 0$, and $g_{\mathrm{st}}(0)$ represents the rigid body transformation between "T" and "S" at the reference configuration. The PoE for the manipulator leg forward kinematics is $g_{\mathrm{st}}(\theta)$.

$$g_{st}(\theta) = e^{\xi_1^{\wedge}\theta_1} \cdot e^{\xi_2^{\wedge}\theta_2} \cdots e^{\xi_{12}^{\wedge}\theta_{12}} \cdot g_{st}(0) \wedge \xi_i = \begin{bmatrix} \upsilon \\ \omega \end{bmatrix} = \begin{bmatrix} -\omega_i \times q_i \\ \omega_i \end{bmatrix}$$

As an example, hereafter we detail the inverse kinematics calculation for a leg. This consists on finding the joint angles, this is, the six physical DoF $(\theta_7 \ldots \theta_{12})$, given the non-physical DoF $(\theta_1 \ldots \theta_6)$ from the footstep planning, which achieve the ZMP humanoid desired configuration $g_{\mathrm{st}}(\theta)$ from the WBT planning. We define four points: "p" is a common point for the axis of the last three DoF, "q" is a common point for the axis of the two first physical DoF, "t" is a point on the axis of the last DoF and "s" is a point not on the axis of the last DoF. Now, it is straightforward to solve the problem in a closed-form way, as follows.

We obtain θ_9 using the third Paden-Kahan subproblem.

$$\left\| e^{-\xi_6{}^{\wedge}\theta_6} \cdots e^{-\xi_1^{\wedge}\theta_1} \cdot g_{st}(\theta) \cdot g_{st}(0)^{-1} \cdot p - q \right\| = \left\| e^{\xi_7{}^{\wedge}\theta_7} \cdots e^{\xi_{12}{}^{\wedge}\theta_{12}} \cdot p - q \right\|$$
$$\longrightarrow \delta = \left\| e^{\xi_9{}^{\wedge}\theta_9} \cdot p - q \right\| \xrightarrow{P-K-3} \theta_9 \quad \textit{Double}$$

We obtain θ_7 and θ_8 using the second Paden-Kahan subproblem.

$$e^{-\xi_6{}^\wedge\theta_6}\cdots e^{-\xi_1{}^\wedge\theta_1}\cdot g_{st}(\theta)\cdot g_{st}(0)^{-1}\cdot p = e^{\xi_7{}^\wedge\theta_7}\cdots e^{\xi_{12}^\wedge\theta_{12}}\cdot p$$
$$\longrightarrow q' = e^{\xi_7{}^\wedge\theta_7}\cdot e^{\xi_8{}^\wedge\theta_8}\cdot p' \xrightarrow{P-K-2} \theta_7,\theta_8\ Double$$

We obtain θ_{10} and θ_{11} using the second Paden-Kahan subproblem.

$$e^{-\xi_9{}^\wedge\theta_9}\cdots e^{-\xi_1{}^\wedge\theta_1}\cdot g_{st}(\theta)\cdot g_{st}(0)^{-1}\cdot t = e^{\xi_{10}{}^\wedge\theta_{10}}\cdot e^{\xi_{11}{}^\wedge\theta_{11}}\cdot e^{\xi_{12}{}^\wedge\theta_{12}}\cdot t$$
$$\longrightarrow q'' = e^{\xi_{10}{}^\wedge\theta_{10}}\cdot e^{\xi_{11}{}^\wedge\theta_{11}}\cdot p'' \xrightarrow{P-K-2} \theta_{10},\theta_{11}\ Double$$

Finally, we obtain $\theta 12$ using the first Paden-Kahan subproblem.

$$e^{-\xi_{11}{}^\wedge\theta_{11}}\cdots e^{-\xi_1{}^\wedge\theta_1}\cdot g_{st}(\theta)\cdot g_{st}(0)^{-1}\cdot s = e^{\xi_{12}{}^\wedge\theta_{12}}\cdot s$$
$$\longrightarrow q''' = e^{\xi_{12}{}^\wedge\theta_{12}}\cdot p''' \xrightarrow{P-K-1} \quad \theta_{12}$$

Besides, we obtain not only one solution for the kinematics problem, but also all possible solutions whether they exists, this is, for the above example of the six DoF Humanoid's leg, we get the eight possible solutions. The same simple procedure is applied to solve the problem for the other leg and a quite similar formula is applicable for the humanoid arms.

Afterwards, the building of a complex motion for the humanoid consists on applying successive goals from the WBT to the OSG algorithm.

6 Conclusions

We think that abstraction saves time, and because of that, this paper presents a slightly more abstract but deterministic formulation, for humanoid robot kinematics modeling, using the theory of **Lie groups**. We present the new approach **SKD (Sagital Kinematics Division)** for a humanoid model, **the new OSG (One Step to Goal) algorithm**, for generating motions and **the new FM3 (Fast Marching Method Modified) algorithm** to path planning collision-free humanoid WBT (Whole Body Trajectory). All these algorithms are analytical techniques suitable (and improvable) for fast calculations and efficient humanoid real-time applications.

References

1. Sethian, J.A. (1999) Level Set Mehods and Fast Marching Methods, Evolving Interfaces in Computational Geometry ... Cambridge University Press
2. Murray, R.M., Li, Z., Sastry, S.S. (1993) A Mathematical Introduction to Robot Manipulation. Boca Raton, FL: CRC Press
3. Park, F.C., Bobrow, J.E., Ploen, S.R. (1995) A Lie group formulation of robot dynamics. Int. J. Robotics Research. Vol. 14, No. 6, pp. 609–618
4. Paden, B. (1986) Kinematics and Control Robot Manipulators. PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley

5. Kahan, W. (1983) Lectures on computational aspects of geometry. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
6. Ball, R. (1900) The Theory of Screws. Cambridge University Press, Cambridge
7. Kanehiro, F., Fujiwara, H., et al. (2002) Open Architecture Humanoid Robotics Platform. Proceedings of the 2002 IEEE International Conference on Robotics & Automation Washington, DC May 2002
8. Kuffner, J.J., Nishiwaki, K., Kagami, S., Inaba, M., Inoue, H. (2003) Motion Planning for Humanoid Robots. In Proc. 11th Int'l Symp. of Robotics Research (ISRR 2003)
9. Inoue, H., Tachi, S., Tnie, K., Yokoi, K., Huirai, S., et al., University of Tokyo, MEL, ETL, Honda R&D Co. Ltd., Matsuita Ltd, Kawasaki Ltd., Fanuc Ltd., (2000) HRP: Humanoid Robotics Project of MITI. In: Proc. First IEEE-RAS International Conference on Humanoid Robots (HUMANOID2000), Sep. 2000
10. Vukobratovic, M., Juricic, D. (1969) Contribution to the Synthesis of Biped Gait. IEEE Tran. On Bio-Medical Engineering, Vol. 16, No. 1, pp. 1–6

GA Optimisation of the PD Coefficients for the LMBC of a Planar Biped

D. Harvey, G.S. Virk, and D. Azzi

DH, GSV – School of Mechanical Engineering, University of Leeds, Leeds, LS2 9JT, U.K.
DA – University of Portsmouth, Faculty of Technology, Portsmouth, Hants. PO1 3DJ
d.r.harvey@leeds.ac.uk
g.s.virk@leeds.ac.uk
djamel.azzi@port.ac.uk

Abstract. This paper describes the use of a multi-objective Genetic Algorithm (GA) for optimising the PD coefficients used in the Linear Model Based Controller (LMBC) for a planar 5-link biped system during the Double Support (DS) phase. Three parameters are used to evaluate the performance of the gains including the mean square error (mse) of the linear and non-linear states and the number of linear models used. Linearisation of the non-linear equations and the use of a GA to optimise the PD coefficients and hence reduce the number of linear models is also discussed.

1 Introduction

Linear models are an approximation of the system in the area about an equilibrium point and are useful for analysis and developing controllers for the system. A typical method of tuning controllers based on linear models, is to use Ziegler-Nichols, however this method is normally applied to tuning a single control loop in isolation. Tuning multiple controllers in parallel requires a different approach such as Gradient Descent (GD) or Genetic Algorithms (GA). The GD method tends to search within a limited area of the search space and is dependent on the derivatives of the functions being available which is not always the case. A GA does not require the derivatives and also searches the whole solution space. In this paper a GA will be used to optimise the gains of the three PD controllers used in the LMBC for a 3 dof planar biped during the DS phase to minimize the mse between the reference and system states.

John Holland [1] first introduced genetic algorithms in the early 70's, but now their use has extended to many fields. In the area of robotics, GAs have been used to generate and optimise robot trajectories of a planar redundant

robot [2]; the inverse kinematic solution of a robot end effector in three dimensions was solved by Khwaja [3] using a multi-objective GA. They point out that the inverse kinematics problem may have many solutions and that the GA is able to cope with this as it searches the complete solution space. In the field of humanoid robots, a multi-objective GA was used to tune 30 gains of 15 PI controllers for a biped to ensure tracking performance and smoothness [4]. A simulation was used to test the individuals of the GA population before application on the physical system. The authors point out the difficulty of using traditional methods of tuning PID controllers for this type of problem.

2 Non-Linear System and Linear Models

The non-linear equations representing the DS phase of a 5-link biped system were developed using Lagrangian dynamics. The system has 3 dof; two knee joints (α_1 and α_2) and one for the trunk (α_3), each is controlled using an independent PD controller (Fig. 1). Typically the system would be controlled directly, however it is intended to compare the performance of a NN model [5] of the DS phase with the system and hence a LMBC is introduced to control both. The linear models are obtained by linearising the system equations using a Taylor series expansion, where the Jacobian matrices are derived numerically due to the complexity of the system.

The control of both the linear models and the system is performed in parallel as shown in Fig. 2.

Initially a linear model is introduced and the errors between the reference and linear model ($X_{lin}(i)$) states are used to calculate the control forces for the system (τ_i) and for the linear model ($\Delta\tau$). A new linear model is introduced when the sum of the differences between the linear and non-linear

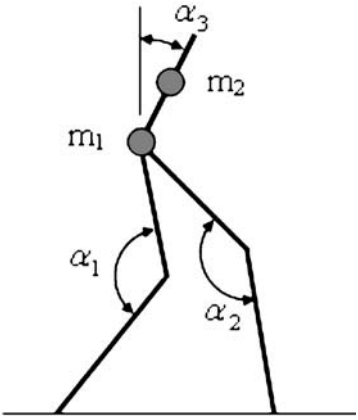


Fig. 1. DS Phase Configuration

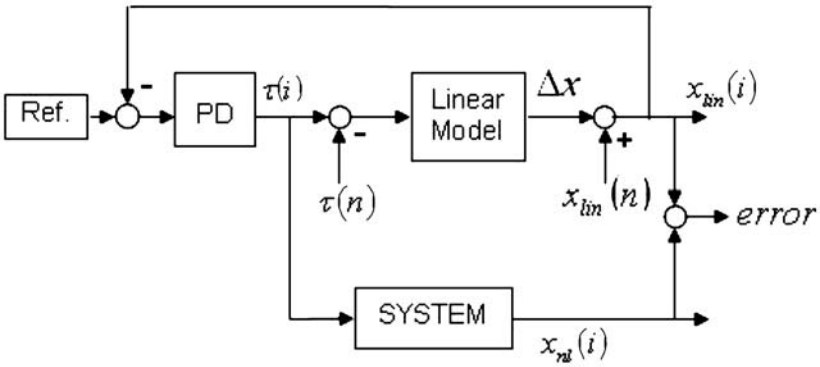


Fig. 2. LMBC of DS Phase

states exceeds some specified threshold value, and is described by (1).

$$state\ sum\ error\ (SSE) = \sum abs\,(X_{nl} - X_{lin}) \tag{1}$$

This process continues until the end of the DS phase.

3 Genetic Algorithms

The method of implementing the GA follows the standard approach using breeding, crossover and mutation to evolve the population. An initial population of 36 individuals is randomly generated, where an individual consists of 7 floating point value genes, three for the P and D gains and one for the desired error. The Pi genes were selected in the range 0 to 4095, whilst the Di genes were chosen to be between 0 and 255. The error gene was not allowed to exceed 0.5. To allow crossover and mutation, the genes were converted to binary numbers with the fractional parts using an 8 bit binary string and the integer parts using 12 and 8 bits for the P_i and D_i respectively. This approach allowed the coefficients to be tuned more finely than if integers were used. Three fitness functions are used to evaluate the performance of the individuals; the number of models, fewer being better, the mse between the linear and reference states and the mse between the non-linear and reference states; all of these values had to be minimised. Instead of using a weighted cumulative sum of the individual fitness criteria to select individuals, each of the fitness functions selected one third of the new population. This approach meant that individuals who performed well in more than one criteria were more likely to be selected more often using the roulette method. Individuals are randomly picked from the new population and bred using two-point crossover with a probability of 60%. The chance of mutation occurs with a probability of 0.1%. The new individuals are simulated and their fitness evaluated, this

process was repeated for one hundred generations before all of the individuals became identical.

4 Results

Once the GA gains (GA) had been obtained, the response of the system could be compared with that of the hand tuned gains (HT), the gains are shown in Table 1. The variation in the controller gains for the two knee joints is due to the different motions of the legs which aim to produce enough momentum for the next single support (SS) phase. The coefficients P1 and D1 are used for the rear leg which is required to straighten whilst the front leg using coefficients P2 and D2 bends forward. The rear leg acts against gravity and hence requires a larger controlling force. This means that the gain coefficients for each will alternate as they become either the forward or rear leg. It is clear GA optimization has changed the coefficients especially for the D terms. The change is most significant for the trunk controller, whose motion affects the action of both legs. The increased coefficients ensure greater stability of the trunk and reduces it's influence on the knee joints.

Table 1. PD Coefficients

	Rear Leg		Front Leg		Trunk	
	P_1	D_1	P_2	D_2	P_3	D_3
HT	3505	25	2705	19	2800	65
GA	2802.7	59.227	2782	10.379	3833.5	209.17

To compare the performance of both the HT and GA tuned controllers in terms of the number of linear models needed, the mse between the linear models and reference states and also between the system and reference states were obtained. The results are shown in Table 2 and Table 3 respectively. GA tuning provided little improvement and in some cases a reduction in the following of the references for the linear model.

Table 2. Linear States mse

	Rear Leg		Front Leg		Trunk	
	α_1	$\dot{\alpha}_1$	α_2	$\dot{\alpha}_2$	α_3	$\dot{\alpha}_3$
HT	2.082e-5	3.739e-2	5.6528e-5	2.6641e-2	1.5218e-5	5.389e-3
GA	2.092e-5	3.302e-2	4.913e-5	2.972e-2	4.662e-6	1.211e-3

Table 3. Non-linear States mse

	Rear Leg		Front Leg		Trunk	
	α_1	$\dot{\alpha}_1$	α_2	α_1	$\dot{\alpha}_1$	α_2
HT	3.745e-5	4.862e-2	9.709e-5	3.7384e-2	1.3221e-5	5.899e-3
GA	3.023e-5	4.221e-2	4.564e-5	3.4471e-2	4.4708e-6	9.6515e-4

Reference following of the non-linear states was improved in all cases using the GA tuned gains.

The behaviour of the control forces derived from the linear models are compared with those obtained from the non-linear model in Fig. 4 using the GA tuned coefficients. There are some variations, but the LMBC forces do follow the trend. The behaviour of the linear and non-linear models using the LMBC are compared in Figs. 5 and 6. The position states especially the knee joints are indistinguishable for the two systems, whilst the trunk position has a small variation, which is only visible due to the magnified scale.

When linearising a system there is always a trade off between the accuracy and the number of linear models required to approximation it. Using the hand tuned PD coefficients, a good balance between the number of linear models required and the mse was found with a threshold value of 0.22 and 20 models. This was achieved by starting the threshold value at zero and calculating the

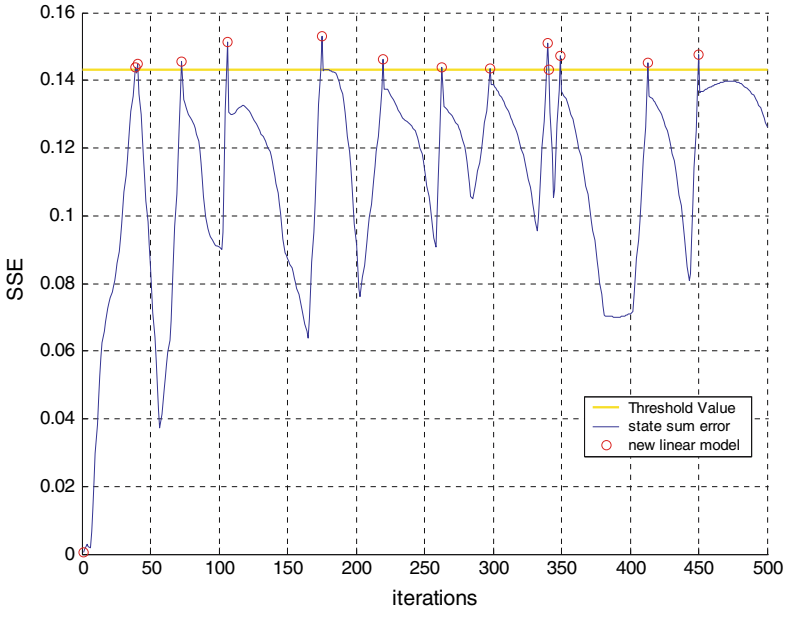


Fig. 3. SSE Behaviour of the GA Optimised Gains

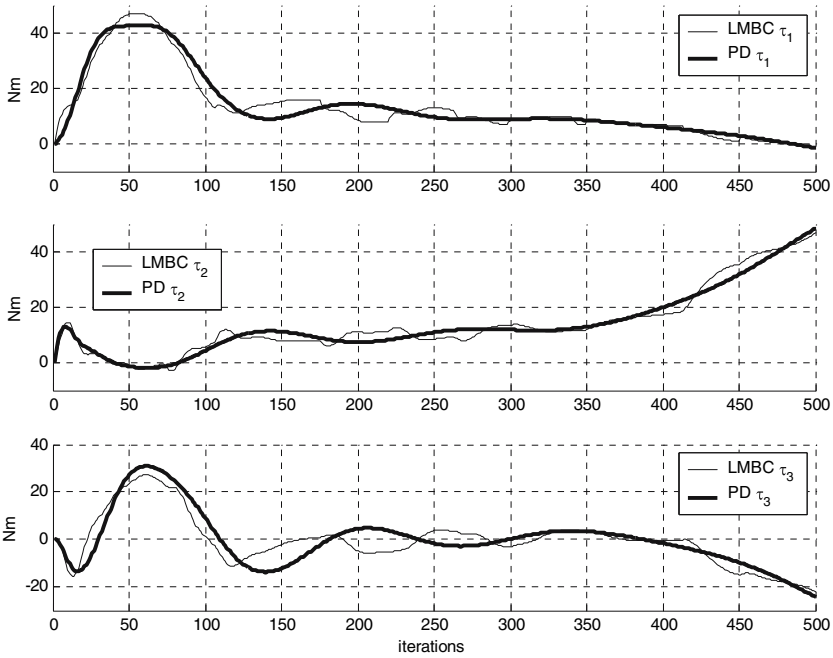


Fig. 4. Control Force Comparison

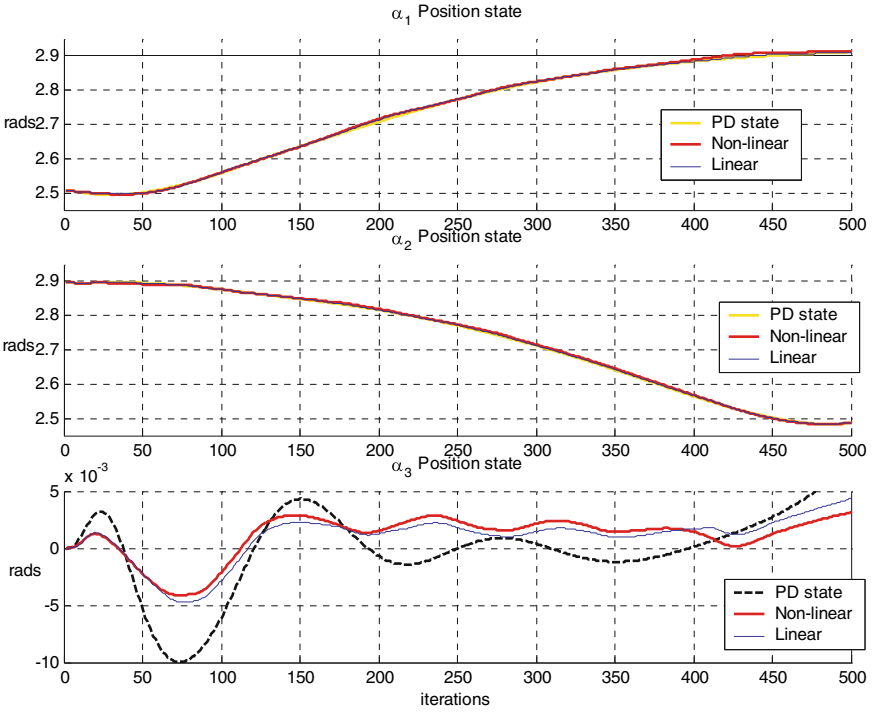


Fig. 5. Position States

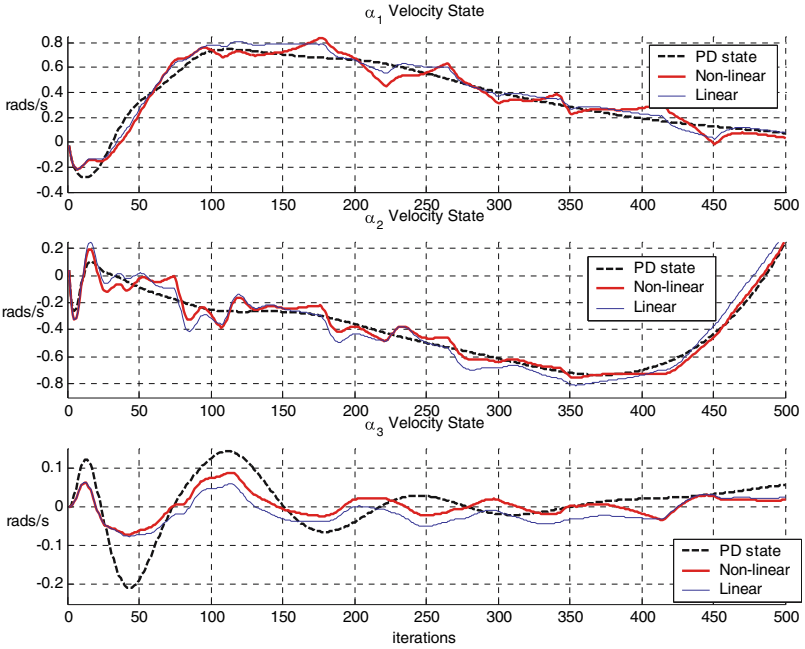


Fig. 6. Velocity States

number of linear models required. This method was repeated 100 times until the threshold reached 1. Plots of the mse and the number of linear models could be compared to identify the best compromise. For the GA optimised gains, 14 linear models were required with a threshold value of 0.14314, which allowed the linear models and the system to track the reference trajectories with similar accuracy to those of the hand tuned gains. This is because the GA was actively trying to reduce the number of linear models and system mse.

The behaviour of the state sum error (SSE) is shown in Fig. 3. The introduction of a new linear model within a few iterations of the previous occurs on two occassions. The first where the SSE initially decreases and then increases and the second where the SSE decreases but not below desired error before the next comparison occurs. It would be possible to avoid this last situation if the SSE could be increased temporarily after the introduction of a new linear model.

The difference between the velocity states is slightly larger although not excessive and is a result of the non-smooth control forces generated in the LMBC.

5 Conclusions

A multi-objective GA using three fitness functions is used to optimise the gains of the PD controllers for a LMBC used to control a planar biped system during the DS phase. The response of the system and linear models using the GA optimised gains are compared to that of a PD controller with hand tuned gains. Three parameters were used to evaluate performance; the mse between the linear and reference states and the mse between the non-linear and reference states and also the number of linear models used+.

The performance of the GA optimized PD controller performed similarly to that of the hand tuned controller when controlling the linear models. The GA optimised gains allowed the non-linear system to track the reference with better accuracy than the hand tuned gains. This improvement is coupled with a reduction of the number of linear models required to approximate the system. Although the GA approach to optimizing the PD coefficients was able to improve the tracking of the reference trajectories, it does require a considerable amount of time to assess each individual and evolve the PD coefficients. However this process only has to be performed once and can be justified by the performance improvements.

References

1. Holland, J.H. (1992) Adaptation in Natural and Artificial Systems. The MIT Press
2. Davidor, Y. (1991) A Genetic Algorithm applied to Robot Trajectory Generation. Handbook of Genetic Algorithms. Editor: Davis, L
3. Khwaja, A.A., Rahman, M.O., Wagner, M.G. (1998) Inverse Kinematics of Arbitrary Robotic Manipulators using Genetic Algorithms. Advances in Robotic Kinematics: Analysis and Control. pp. 375–382. Kluwer Academic Publishers
4. Roberts, J., Kee, D., Wyeth, G. (2003) Improved Joint Control using Genetic Algorithms for a Humanoid Robot. Australasian Conference on Robotics and Automation, Brisbane, Australia
5. Harvey, D., Virk, G.S., Azzi, D. (2003) Neural Network Modelling of a Planar 5-dof Biped. Clawar 2003, Sicily

Three-Dimensional Running is Unstable but Easily Stabilized

Justin E. Seipel[1] and Philip J. Holmes[1,2]

[1] Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, U.S.A.
[2] Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544, U.S.A.

Summary. We analyze a simple three-dimensional model for running: the spring-loaded inverted pendulum (3D-SLIP). Our formulation reduces to the vertical plane (SLIP) and horizontal plane lateral leg spring (LLS) models in the appropriate limits. Using the geometry and symmetries intrinsic to the system, we derive an explicit approximate stride-to-stride mapping and show that all left-right symmetric gaits are unstable. Continuation to fixed points for the 'exact' mapping confirms instability, and we describe a simple feedback control of leg placement at touchdown that stabilizes them.

1 Introduction

Animals with a great range of morphologies, and some legged machines, produce similar net force patterns and translational center-of-mass (COM) kinematics in the sagittal[a] and horizontal planes [1, 2]. Simple planar spring-mass models or 'templates' [3] have been developed to account for this dynamics. The spring-loaded inverted pendulum (SLIP) model [4] describes sagittal plane motions of upright-postured animals of varying leg numbers (humans, kangaroos, dogs, etc.) and of some legged robots [2, 5]. An analogous Lateral Leg Spring (LLS) model similarly captures the horizontal plane dynamics of sprawled-posture animals such as insects [6, 7]. SLIP and LLS are simple, contain few key parameters, and are analytically tractible. Furthermore, control strategies based on SLIP have proven successful in designing robotic runners [5, 8].

In this brief announcement we extend the analytical studies of [4] and [6] to a three-dimensional point-mass model (Fig. 1) and show that while families of periodic gaits exist as in SLIP and LLS, unlike those cases, all such gaits are unstable. We then describe a simple leg touchdown feedback scheme

[a] The saggital plane passes through the body center and spans the fore-aft and vertical directions.

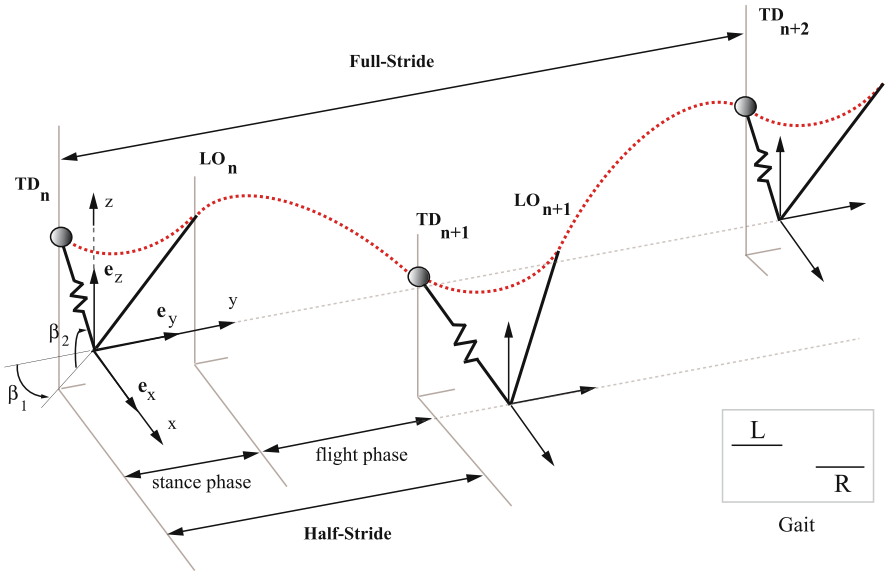


Fig. 1. The model and a typical three-dimensional gait

that stabilizes sagittal plane motions. Full details will appear in a subsequent paper [9].

2 Governing Equations and Stride Map

3D-SLIP consists of a massless, passively-sprung leg attached at a frictionless pin-joint to a point mass and (during stance) to the ground: Fig. 1. Running comprises stance and flight phases with discrete touchdown (TD) and liftoff (LO) events where the governing equations switch, making 3D-SLIP a hybrid dynamical system. TD occurs when the leg, uncompressed and deployed at a specified angle (β_1, β_2), contacts the ground. The leg compresses throughout stance, and LO occurs when it next unloads; free flight then ensues until TD occurs again: Fig. 1. Feedforward control is required following LO to reset the leg in anticipation of TD, but the system is otherwise passive and energy is globally conserved, since no impacts or impulses occur.

Applying Newton's second law and nondimensionalizing for a Hookean spring, the 3D-SLIP equations are:

$$\ddot{\mathbf{q}} = \tilde{k}(1/\zeta - 1)\mathbf{q} - \tilde{g}\mathbf{e}_z\,, \qquad \text{(stance)}\ ; \tag{1}$$

$$\ddot{\mathbf{q}} = -\tilde{g}\mathbf{e}_z\,, \qquad \text{(flight)}\ ; \tag{2}$$

$$\mathcal{S}_{\mathrm{TD}} = (\mathbf{q} \mid q_z = \sin\beta_2\ ,\ \dot{q}_z \le 0)\,, \qquad \text{(touchdown)}\ ; \tag{3}$$

$$\mathcal{S}_{\mathrm{LO}} = (\mathbf{q} \mid \zeta = 1\ ,\ \dot{\zeta} \ge 0)\,, \qquad \text{(liftoff)}\ . \tag{4}$$

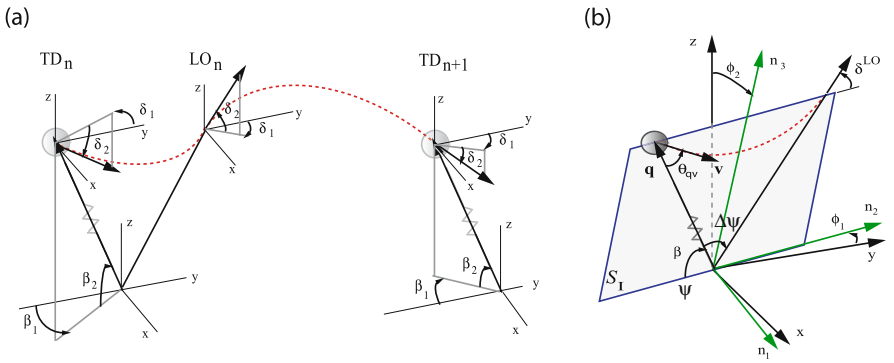


Fig. 2. 3D-SLIP, showing coordinate systems and invariant stance plane $\mathcal{S}_\mathrm{I}$

Here $\tilde{g} = \frac{g\zeta_0}{v_0^2} = \frac{E_{gravity}}{E_{kinetic}}$ and $\tilde{k} = \frac{k\zeta_0^2}{mv_0^2} = \frac{E_{spring}}{E_{kinetic}}$ are the nondimensional gravitational (an inverse *Froude* number) and stiffness parameters, and g, ζ_0, k, and m are the physical constants of the original system: acceleration due to gravity, leg spring equilibrium length, spring constant, and mass, respectively.

Due to energy conservation the velocity magnitude at TD, v_0, remains constant, so we may describe the TD state by the velocity direction (δ_1, δ_2) alone: Fig. 2a. Moreover, by left-right symmetry in (1-4) the full stride map is composed of two half-stride maps: $P_{full} = P \circ P$. The half-stride map acts from TD to *next* TD:

$$P : \left[\delta_1^{TD_n}, \delta_2^{TD_n}\right] \rightarrow \left[\delta_1^{TD_{n+1}}, \delta_2^{TD_{n+1}}\right] . \tag{5}$$

Furthermore, for L-R symmetric orbits, $DP_{full} = DP^2$, and so eigenvalues at fixed points of P_{full} are the squares of those of P. However, P is only implicitly defined since the stance equations are non-integrable.

As in [4] we approximate P by assuming that gravity forces are negligible in comparison to spring forces during stance. For a given touchdown condition, parameterized by the leg angles (β_1, β_2) and an initial state (δ_1, δ_2), the resulting central force problem evolves in an invariant plane S_I parameterized as shown in Fig. 2b. We note that other approximations that retain leading order gravitational effects have been proposed [10, 11], but in the general three-dimensional case these schemes would not result in an invariant stance plane. The approximate map P_a is composed of maps B from inertial coordinates to those of S_I at TD, P_{2D} through stance, R which returns to inertial coordinates at LO, and P_{fl} describing the flight phase:

$$P_a = P_{fl} \circ R \circ P_{2D} \circ B . \tag{6}$$

We now describe the component maps, which are written in terms of the inertial $((x, y, z), (\beta_1, \beta_2)$, and $(\delta_1, \delta_2))$ and stance plane $((\phi_1, \phi_2)$ and $(\beta, \theta_{qv}, \delta^{LO}))$ coordinates of Fig. 2.

The mapping B: The plane S_I is spanned by the leg and velocity vectors $\mathbf{q}$ and $\mathbf{v}$ and so is normal to their cross product:

$$\begin{aligned} \mathbf{n} &:= -\mathbf{q} \times \mathbf{v} = (n_x, n_y, n_z)\,, \quad \text{where} \\ n_x &= \sin\beta_2 \cos\delta_2 \cos\delta_1 - \cos\beta_2 \cos\beta_1 \sin\delta_2\,, \\ n_y &= \sin\beta_2 \cos\delta_2 \sin\delta_1 - \cos\beta_2 \sin\beta_1 \sin\delta_2\,, \\ n_z &= -\cos\beta_2 \cos\delta_2 \sin(\beta_1 - \delta_1)\,; \end{aligned} \tag{7}$$

The normalized vectors defining a *stance frame* aligned with S_I are:

$$\mathbf{n}_1 := \frac{\mathbf{n}}{||\mathbf{n}||}\,,\ \mathbf{n}_2 := \frac{\mathbf{e}_z \times \mathbf{n}}{||\mathbf{e}_z \times \mathbf{n}||}\,,\ \mathbf{n}_3 := \mathbf{n}_1 \times \mathbf{n}_2\,, \tag{8}$$

and the Euler angles ϕ_1 and ϕ_2 determining the transformation from inertial frame to stance frame are:

$$\phi_1 = \tan^{-1}\frac{n_y}{n_x}\,,\ \phi_2 = -\tan^{-1}\frac{n_z}{n_{xy}}\,, \tag{9}$$

where $n_{xy} = (n_x^2 + n_y^2)^{-1/2}$: see Fig. 2.

The angles β and θ_{qv} describing the leg and velocity in S_I are found using dot products of vectors: $\beta = \cos^{-1}(-\mathbf{n}_2 \cdot \mathbf{q})$, and $\theta_{qv} = \cos^{-1}(-\mathbf{v} \cdot \mathbf{q})$. Since the vectors $\mathbf{v}$, $\mathbf{q}$, and $\mathbf{n}$ are given in terms of the state (δ_1, δ_2) and parameters (β_1, β_2), this yields an explicit map B:

$$B : [\delta_1, \delta_2] \to [\theta_{qv}, \beta, \phi_1, \phi_2]\,. \tag{10}$$

The mapping $\mathbf{P}_{2D}$: Within S_I the COM position is described in terms of the polar coordinates (ζ, ψ); see Fig. 2b. Due to conservation of energy ($E = \frac{\dot\zeta^2}{2} + \frac{\zeta^2\dot\psi^2}{2} + \frac{\tilde k}{2}(\zeta - 1)^2$) and angular momentum about the foot ($p_\psi = \zeta^2\dot\psi$) the equations of motion can be reduced to a single, integrable first order ODE, yielding a quadrature $\Delta\psi = \Delta\psi(\theta_{qv}, \tilde k)$ involving elliptic integrals. Due to reversibility symmetry about midstride, $\Delta\psi$ gives the velocity at LO as $\delta^{LO} = \pi - \theta_{qv} - \beta - \Delta\psi(\theta_{qv}; \tilde k)$. The height at LO in the inertial frame is $z^{LO} = \sin(\beta + \Delta\psi)\cos\phi_2$. The mapping P_{2D} evolving the state in the invariant plane from TD to LO is then

$$P_{2D} : [\theta_{qv}, \beta, \phi_1, \phi_2] \to [\delta^{LO}, z^{LO}, \phi_1, \phi_2]\,. \tag{11}$$

The mapping R: To obtain expressions for the LO velocity direction (δ_1, δ_2) in the inertial frame, we apply rotation matrices to the velocity vector in the stance frame and equate it with that in the inertial frame, resulting in:

$$\delta_1^{LO} = \tan^{-1}(\tan\delta^{LO}\sin\phi_2) - \phi_1\,,\ \delta_2^{LO} = \sin^{-1}(\sin\delta^{LO}\cos\phi_2)\,. \tag{12}$$

The mapping R from invariant plane to inertial coordinates is then

$$R : [\delta^{LO}, z^{LO}, \phi_1, \phi_2] \to [\delta_1^{LO}, \delta_2^{LO}, z^{LO}]\,. \tag{13}$$

The mapping $\mathbf{P}_{fl}$: To ensure that energy is conserved from stride to stride, we must also account for the change in gravitational potential in computing the LO velocity magnitude:

$$v^{LO} = \sqrt{(v^{TD})^2 + 2g(z^{TD} - z^{LO})} = \sqrt{1 + 2g(\sin\beta_2 - z^{LO})}\,. \tag{14}$$

Finally, to find (δ_1, δ_2) at the next TD event we make use of symmetries in flight. Since the gravitational force has zero x and y components during flight, momentum is conserved in these directions and the COM path projected on the x-y plane is a straight line, implying that: $\delta_1^{TD_{n+1}} = \delta_1^{LO}$. Conservation of energy in the vertical direction along with vLO and the fact that $v^{TD} = 1$ is normalised, yields: $\delta_2^{TD_{n+1}} = \cos^{-1}(\cos\delta_2^{LO}\sqrt{1 + 2g(\sin\beta_2 - z^{LO})}\,)$. The mapping P_{fl} which evolves the flight dynamics in inertial coordinates is then

$$P_{fl} : \left[\delta_1^{LO}, \delta_2^{LO}, z^{LO}\right] \rightarrow \left[\delta_1^{TD_{n+1}}, \delta_2^{TD_{n+1}}\right]\,. \tag{15}$$

3 Existence and Stability of Symmetric Periodic Gaits

We seek gait trajectories that are reflection-symmetric about their midpoints in both stance and flight (Fig. 1). In stance, this implies that $\Delta\psi = \pi - 2\beta$ and $\phi_1 = 0$. From the geometry of Fig. 2, appealing to the reflection symmetries in stance (in S_I) and the parabolic flight phase, we see that the nondimensionalized equations (1-4) have a three-parameter family of periodic gaits characterized by the stance plane angle ϕ_2, the touchdown angle β in S_I, and $\tilde{k}$.

General $\phi_2 \in (0, \pi/2)$ motions produce very complicated return maps P_a, but confined to the sagittal (vertical) plane they simplify significantly. For such upright motions, $\beta_1 = 0$, $\theta_{qv} = \beta_2 - \delta_2$, $\phi_2 = 0$, and $\phi_1 = 0$, so we need only satisfy $\Delta\psi(\beta_2 - \delta_2; \tilde{k}) = \pi - 2\beta_2$. There is a two-parameter family of fixed points $\bar{\delta_2}$ which satisfy the previous equation. For constant β_2, the RHS is constant and so level sets of $\Delta\psi$ correspond to a one-parameter branch of fixed points over $(\delta_2, \tilde{k})$-space with constant β_2: see Fig. 3. Purely horizontal motions $\phi_2 = \pi/2$ (LLS), may also be treated as a special (simple, but degenerate) case.

Stability: The Jacobian DP_a is, in general, a complicated expression due to compositions of inverse trigonometric functions in P_a, but for the orbits confined to the vertical plane (SLIP), we obtain a simple diagonal Jacobian:

$$DP_a = \begin{bmatrix} -\sin(\beta + \delta)/\sin(\beta - \delta) & 0 \\ 0 & 1 + (1 + \tilde{g}\cos\beta\cot\delta)\frac{\partial\Delta\psi}{\partial\theta_{qv}} \end{bmatrix}. \tag{16}$$

For admissible ranges of $\beta \in (0, \pi/2)$ and $\delta \in (0, \beta)$ it follows that $-\sin(\beta + \delta)/\sin(\beta - \delta)$ is less than -1 and so all SLIP fixed points are unstable. Numerically evaluating the eigenvalues of DP_a in general, and using continuation

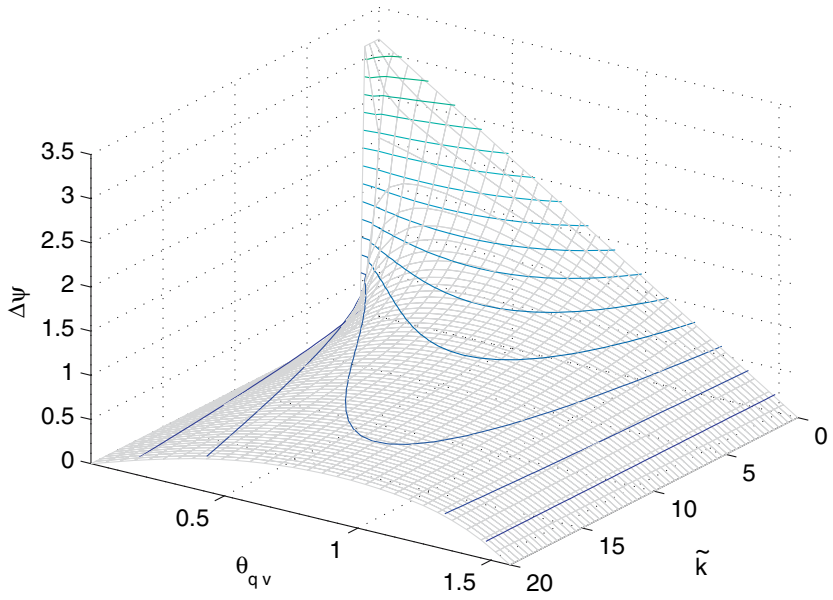


Fig. 3. Quadrature for the Hookean spring law showing level sets

methods for the exact map DP, we find *instability throughout the* $(\beta_1, \beta_2, \tilde{g}, \tilde{k})$ *parameter space*. Figure 4 shows examples of solutions and the corresponding eigenvalues for the exact mapping, with gravity present in stance. Column III shows that uncontrolled solutions are unstable. Column IV shows results with a control law to be discussed next.

4 Stabilizing with a Leg Placement Protocol

We may stabilize 3D-SLIP motions by placing the leg further out in the direction of the perturbation (changing β_1, depending on δ_1). This is perhaps the most natural, intuitive control law for this application. Rederiving DP_a with feedback we find that the condition $1 < \frac{\partial \beta_1}{\partial \delta_1} < \frac{\tan \beta}{\tan \delta}$ is sufficient for stability. Thus, if we put the leg out at an angle greater than that of the COM velocity but not too much greater, perturbations decay. Numerical simulations confirm this stabilizing effect for the exact SLIP dynamics.

To embed this control law in the full dynamics we choose the following simple feedback rule to determine $\beta_1(\delta_1)$ at touchdown from the actual lateral LO velocity angle δ_1:

$$\beta_1(\delta_1) = \hat{\beta}_1 + K(\delta_1 - \hat{\delta}_1) \; . \tag{17}$$

Here the hats denote the nominal values of β_1 and δ_1, in order to apply the rule to general as well as upright ($\phi_2 = 0$) gaits. This is essentially the same

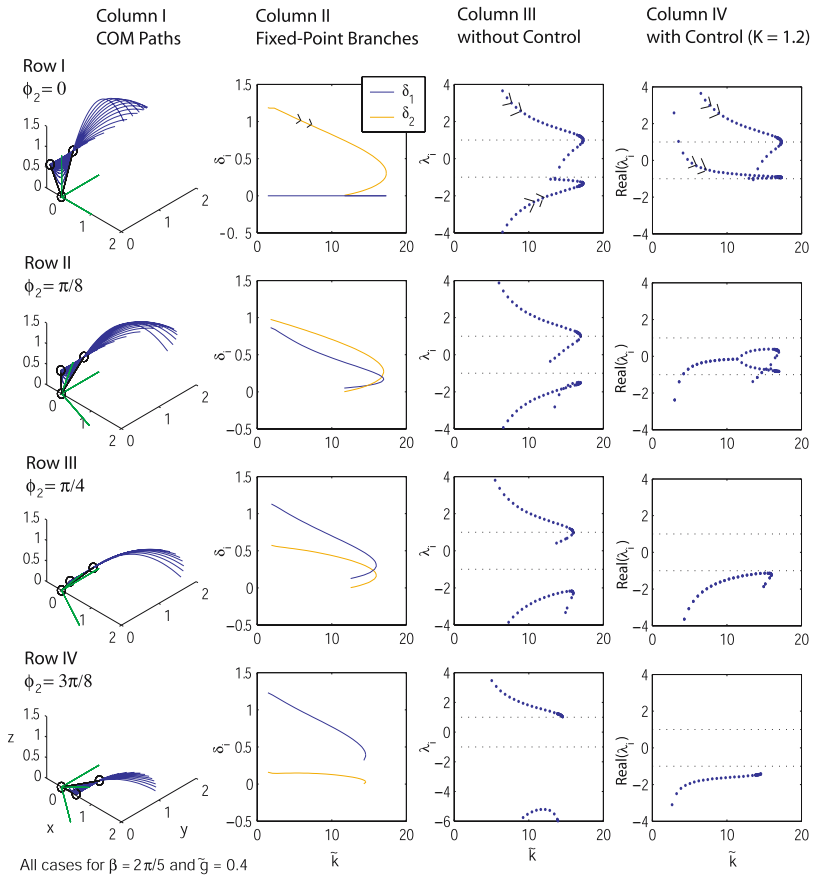


Fig. 4. COM paths in inertial space, solution branches, and corresponding eigenvalues for the exact mapping without and with control present. In column III, the lower and upper branches correspond to eigenvalues given by the top-left and bottom-right entries of DP_a (16) respectively. Note that, in the controlled case, these two real eigenvalues have become a complex conjugate pair over part of the $\tilde{k}$ range for $\phi_2 = \pi/8$ and over the entire range for $\phi_2 \geq \pi/4$ and that both eigenvalues have modulus less than one over a part of the $\tilde{k}$ range for $\phi_2 \leq \pi/8$

as the leg splay strategy employed by Kuo [12] in stabilizing a passive three-degree-of-freedom rigid leg walker.

Figure 4 shows solutions and corresponding eigenvalues for simulations with and without the control law (17) present: here the exact map is evaluated numerically and gravity is included in stance. These numerical simulations show that the control law does indeed stabilize the perturbed dynamics of the exact case near $\phi_2 = 0$, although not for large ϕ_2 (the unstable branches remain unstable for $\phi_2 = \pi/4$ and $3\pi/8$).

5 Conclusion

We have derived an explicit approximation to the stride-to-stride Poincaré mapping for a three-dimensional spring-loaded point-mass runner (or hopper) and shown that all L-R symmetric gaits are unstable. For general gaits the components of the map and its linearization are complicated, but the invariant stance plane geometry is appealingly simple and leads to clear intuitive understanding of gait families. The map is particularly simple for gaits confined to the sagittal plane, and its form suggests a simple leg-placement, feedback control which successfully stabilizes gaits of the exact map that do not depart too far from vertical.

References

1. M.H. Dickinson, C.T. Farley, R.J. Full, M.A.R. Koehl, R. Kram, and S. Lehman. How animals move: an integrative view. *Science*, 288:100–106, 2000.
2. R. Altendorfer, U. Saranli snd H. Komsuoglu, D.E. Koditschek, B. Brown, M. Buehler, E. Moore, D. McMordie, and R.J. Full. Evidence for spring loaded inverted pendulum running in a hexapod robot. In *Experimental Robotics VII*, pp. 291–302. Springer Verlag, 2002.
3. R.J. Full and D.E. Koditschek. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *J. Exp. Biol.*, 202:3325–3332, 1999.
4. R.M. Ghigliazza, R. Altendorfer, P. Holmes, and D. Koditschek. A simply stabilized running model. *SIAM J. Applied Dynamical Systems*, 2:187–218, 2003.
5. M. Raibert. *Legged Robots that Balance*. MIT Press, Cambridge, MA, 1986.
6. J. Schmitt and P. Holmes. Mechanical models for insect locomotion: Dynamics and stability in the horizontal plane – Theory. *Biological Cybernetics*, 83(6):501–515, 2000.
7. J. Schmitt, M. Garcia, R.C. Razo, P. Holmes, and R.J. Full. Dynamics and stability of legged locomotion in the horizontal plane: A test case using insects. *Biological Cybernetics*, 86(5):343–353, 2002.
8. U. Saranli and D.E. Koditschek. Template based control of hexapedal running. In *ICRA 2003*. IEEE, 2003.
9. J.E. Seipel and P.J. Holmes. Running in three dimensions: Analysis of a point-mass sprung-leg model. Submitted for publication.
10. W.J. Schwind and D.E. Koditschek. Approximating the stance map of a 2 dof monoped runner. *J. Nonlin. Sci.*, 10(5):533–568, 2000.
11. H. Geyer, A. Seyfarth, and R. Blickhan. Stability in bouncing gaits – Analytical approach. Submitted for publication, 2004.
12. A.D. Kuo. Stabilization of lateral motion in passive dynamic walking. *Int. J. Rob. Res.*, 18 (9):917–930, 1999.

A Biomimetic Approach for the Stability of Biped Robots

Javier de Lope and Darío Maravall

Department of Artificial Intelligence, Faculty of Computer Science
Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Madrid, Spain
{jdlope,dmaravall}@fi.upm.es

Summary. A robotic architecture that is able to acquire very primitive reasoning abilities from the pure coordination of perception and action is proposed. Basically, it is materialized by means of a bio-inspired mechanism, in which perceptions from the environment and robot actions manage to autonomously coordinate themselves, chiefly through the environment's responses. This architecture is applied to the problem of the stability of a biped robot under the presence of disturbances generated by external forces or by the inherent change of the center of masses position when the robot walks.

1 Introduction

We have previously developed a biomimetic approach thanks to which we have been able to solve specific manipulation and locomotion problems in both articulated mechanisms [1] and wheeled vehicles [2, 3].

In this paper, we propose a robotic architecture that is able to acquire very primitive reasoning abilities from the pure coordination of perception and action. Specifically, we try to maintain the stability of a biped robot under the presence of movements or disturbances. These disturbances can be generated by external forces such as the ones that produce a bad consistence or instability of the supporting surface or by the inherent change of the center of masses position when, for example, the robot walks during a single step.

This biomimetic approach allows to materialize two interesting features. First, the robot actions can be performed without explicitly solving the robot kinematics which can be a very hard problem due to the inherent redundancy of multi-joints robots [4]. Also, there is no need for the robot to have a formal representation of the environment to perform its movements.

To our knowledge, the cerebellar model articulation controller (CMAC), proposed by Albus in the seventies [5], is the first example of a robot controlled without explicitly employing its kinematics expressions. It is based on

learning the mapping functions, given as look-up tables, between the working space – i.e. Cartesian coordinates – and the configuration space – i.e. joints coordinates – by means of visual feedback. This line of research was pursued mainly in the field of cognitive science and produced interesting simulated robots and a few physical prototypes [6]. Its reliance on look-up tables, mapping the sensory information to the control actions, makes this paradigm very prone to combinatorial explosion, unless a local search in the mapping functions is applied. Our proposed method, however, differs significantly, as it is based on temporal and spatial utility functions, so that the robot performs the mapping between sensory information and actions by optimizing what we call *utility functions*, which dramatically reduces the search space.

The paper is organized as follows. Section 2 describes the humanoid robot and its workspace. Sections 3 and 4 introduce the basis of the proposed method and formulate the problem of the humanoid robot stability from the viewpoint of the biomimetic approach, respectively. Some experimental results obtained in a simulated environment are commented in the Sect. 5. Finally, the conclusions and future work are presented.

2 The Robotic Biped

The robot model is made up of two legs and the hip. Each leg is composed of six degrees of freedom, which are distributed as follows: two for the ankle, on the pitch and roll axes, one for the knee, on the pitch axis, and three for the hip, on each of the axes. A more detailed description of the robot dimensions and movement range is given in [7].

We can consider our biped robot as a chain of rigid links, which are pairwise connected to each other by joints. One of the robot feet will always be in contact with the ground, and it will be considered as the first link. The other foot will be free to move around the space, and it will be considered as the last link. For our particular case and considering the right foot as fixed to the ground, the assignment of the coordinate frames to the robot joints is illustrated in Fig. 1. The study of the inverse case, that is, when the left foot is fixed to the ground and the right foot is able to move, would result in a coordinate frame assignment similar to the one described above and will not be developed here.

We are considering the distribution of the masses to be uniform for every link and that every link has a symmetrical shape. The center of masses (COM) of each link will then be placed at the geometric center of the link. As this applies to each of the robot links, we can determine the global COM of the model using:

$$x_c = \frac{1}{M}\sum_{i=1}^{N} m_i x_i \; ; \quad y_c = \frac{1}{M}\sum_{i=1}^{N} m_i y_i \; ; \quad z_c = \frac{1}{M}\sum_{i=1}^{N} m_i z_i \tag{1}$$

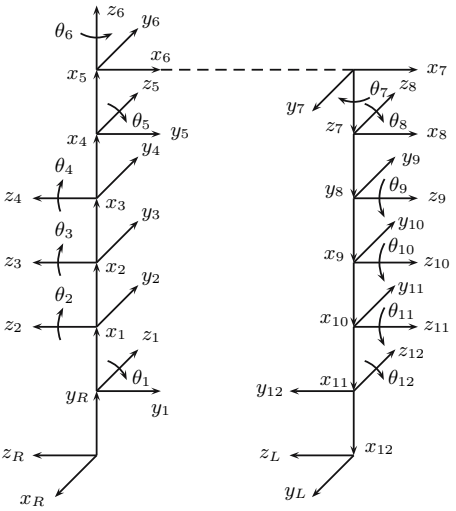


Fig. 1. Coordinate system associated with the robot joints

where (x_i, y_i, z_i) is the geometric center of the link i, m_i is the mass of the link i, M is the mass of the robot, and N the number of links.

3 Biomimetic Approach to Autonomous Robot Behavior

Our method is inspired in the way that humans and, in general, animals control and guide their articulated extremities to maintain stability and to avoid falling when moving, which is based on adaptation and learning through sensory feedback. Obviously, human beings and animals do not move their articulated members using the kinematics and dynamics models associated with their respective extremities. Rather, they accomplish this complex task by an action/reaction process based on perceptual feedback. To generalize our discussion, we shall use the term of autonomous agent to emphasize the idea of its independence of a priori knowledge injected by the designer.

An autonomous agent possesses a set of input variables or actuators responsible for their articulated member movements. Following the usual notation in control systems, let us denote these input variables as $u_1, u_2, \ldots, u_N$. When these variables coincide with the generalized variables $q_1, q_2, \ldots, q_M$ that define the degrees of freedom of the respective articulated member, then we have a direct control system. The set of generalized or internal variables may or may not coincide with the state variables appearing in the mathematical description of the system in the state-space representation. Furthermore, there is a set of external and observable variables, $y_1, y_2, \ldots, y_P$, which conform the basic sensory information needed by the agent to modify its behavior in order to control the performance of the task at hand. With the perceptual

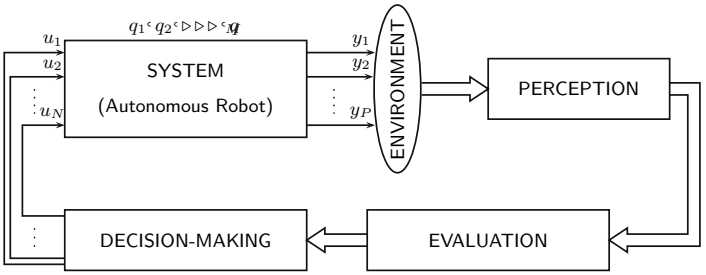


Fig. 2. Block-diagram of an autonomous agent with perceptual feedback

feedback, the control loop is closed and, then, the control or input variables $u_1,\ u_2,\ \ldots, u_N$ are updated as a result of the evaluation of the sensory information and the *reasoning* and decision-making processes. This conceptual model is shown in Fig. 2.

There are three fundamental elements interposed between the agent and its environment: perception, evaluation and decision-making. The perceptual element is the quantitative measurement of the effects of agent actions on the state of the environment which is embodied in the observed variables $y_1,\ y_2, \ldots, y_P$. This sensory information is vital for activating the reactive chain inside the physical agent aimed at changing the actions $u_1,\ u_2, \ldots, u_N$.

The reasoning or thinking capability of the agent is, actually, situated in-between the agent perception and action components. This reasoning activity has been divided into two stages: (1) evaluation and (2) decision-making. The idea is that, in the evaluation phase, the agent analyzes what is happening in the environment as regards its final goals or purposes. Therefore, the agent is endowed with a plan or set of goals, and it can assess, through its perception of the environment and its own states, whether its actions are conducive to fulfilling its plan. It is here that reasoning comes into the play as a mean of deciding which actions to apply, once the effects of its previous actions on the environment have been evaluated. As a consequence of its reasoning activity, the agent is able, again, to move its articulated members by means of the input variables. Thus, we can see now that reasoning or thinking is placed between perception and action. Note, also, that we have not identified reasoning with intelligence, as the latter clearly includes perception and action as well.

4 Biomimetic Stability

For a biped robot, the basic goal is to maintain always its stability. This complex goal can be divided into three sub-goals. First, we consider the stability in the frontal plane, i.e. to keep stable the robot by means of forward/backward movements. The goal of maintain the frontal stability can be expressed as the following index:

$$J_1 = \frac{1}{2}\left(x_c - x_r\right)^2 \tag{2}$$

where x_c is the coordinate of the center of masses in the advance axis and x_r is the coordinate of the desired position of the center of masses in that axis. Obviously, the desired position in the simplest case is the center of the support polygon.

The second sub-goal controls the robot stability in the sagittal plane, trying to maintain the robot stable by applying lateral forces. This sub-goal can be materialized with the following performance index:

$$J_2 = \frac{1}{2}\left(z_c - z_r\right)^2 \tag{3}$$

where z_c is the center of masses coordinate in the transversal axis and z_r is the coordinate of the desired position in the same axis.

Some control actions could lower the center of masses position to reach a better stability. For instance, by blending the robot's knees. This kind of postures are, however, unacceptable as the robot can adopt unnatural postures while walking. Therefore, an index to correct this problem is also needed and can be expressed as

$$J_3 = \frac{1}{2}\left(y_{\text{hip}} - y_r\right)^2 \tag{4}$$

where y_{hip} is the height of the hip and y_r is the recommended height for natural postures.

Note that the tuple (x_r, y_r, z_r) describes the desired posture to be adopted by the robot during the stabilization process and provides three parameters for controlling the posture. For example, the tuple $(0, y_{\max}, 0)$ denotes a posture with the knees completely outstretched and the projection of the center of masses in the center of the support polygon.

The minimization of each individual performance index can be materialized by a gradient descent algorithm:

$$\dot{\theta}_i^1 = -\mu_i^1 \frac{\partial J_1}{\partial \theta_i}\ ; \quad \dot{\theta}_i^2 = -\mu_i^2 \frac{\partial J_2}{\partial \theta_i}\ ; \quad \dot{\theta}_i^3 = -\mu_i^3 \frac{\partial J_3}{\partial \theta_i} \tag{5}$$

where θ_i^1, θ_i^2 and θ_i^3 are the robot actions generated by each individual sub-goal. The discrete-time version for the sub-goals are

$$\theta_i^j(k+1) = \theta_i^j(k) - \mu_i^j \left.\frac{\partial J_j}{\partial \theta_i}\right|_{\theta_i(k)}\ ; \quad j = 1, 2, 3 \tag{6}$$

Each individual action depends on the sensory-based gradients, which are computed by using the measurements provided by the robot sensors. Due to this sensory dependency, the modules of the respective control outputs tend to be disproportionate to the internal dynamics of the robot: sometimes the sensory-based gradients are too strong or, inversely, they are too small, as far as the robot's natural dynamics is concerned. To solve this problem of scale,

we have devised a look-up table mapping the module of the sensory-based gradients to the natural dynamics of the robot.

The final control action of the robot will be formed by a combination of the actions recommended by each sub-goal:

$$\theta_i(k+1) = \omega_i^1 \theta_i^1(k+1) + \omega_i^2 \theta_i^2(k+1) + \omega_i^3 \theta_i^3(k+1) \tag{7}$$

5 Experimental Results

To demonstrate the effectiveness of the proposed method for the stability of biped robots, we have considered a set of situations in which the robot has to maintain the stability. Figure 3 illustrates the robot stabilization process from the initial position (see Fig. 1) to a stable position when the robot is on a surface with an orientation of 10º on the roll and pitch axes. Figure 3a and Fig. 3b show the consecutive robot positions in the frontal and sagittal planes, respectively. Figure 3c shows the evolution of the COM projection during the stabilization process. Note that the final position of the COM projection is $(50, 0)$ that correspond to the middle point between both feet.

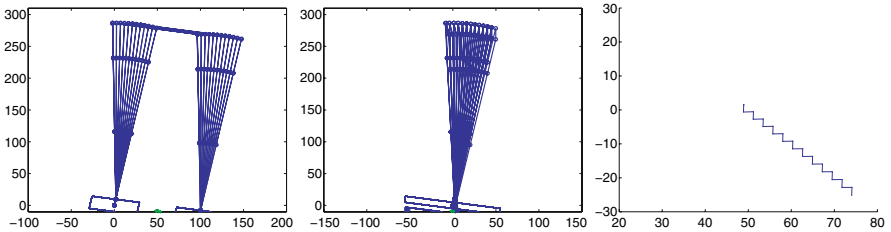
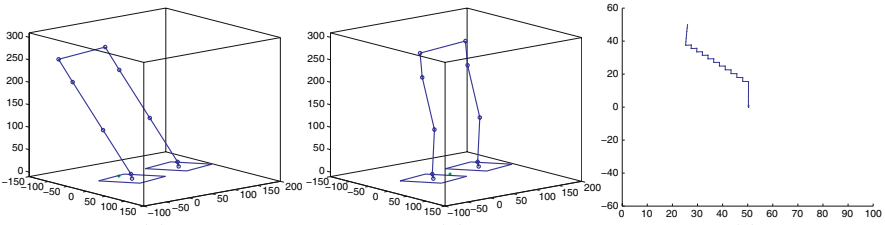


Fig. 3. Robot stabilization in frontal and sagittal planes

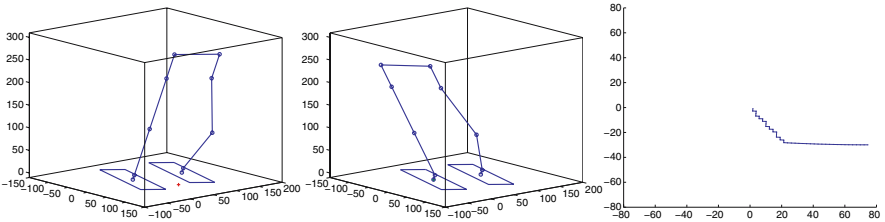
Figure 4 shows an example which requires that the robot modifies the knee joints to reach the target stability posture. Figure 4a shows the initial position that corresponds to the final position of the previous example, but in this case, the surface is rotated -10º on the roll axis and -5º on the pitch axis. Figure 4b shows the final position. Figure 4c shows the evolution of the COM projection during the stabilization process.

In previous examples, both feet are on the ground. In the example shown in Fig. 5, the robot has lightly raised its left foot. The surface is characterized by a rotation of 10º on the roll and pitch axes. The desired final position of the COM projection is at the coordinates $(0, 0)$ that correspond to the coordinate frame origin of the right ankle. Note that the controller automatically varies the COM projection coordinate in the frontal plane to reach the stable posture, and then, it tries to achieve the target stability point.



(a) (b) (c)

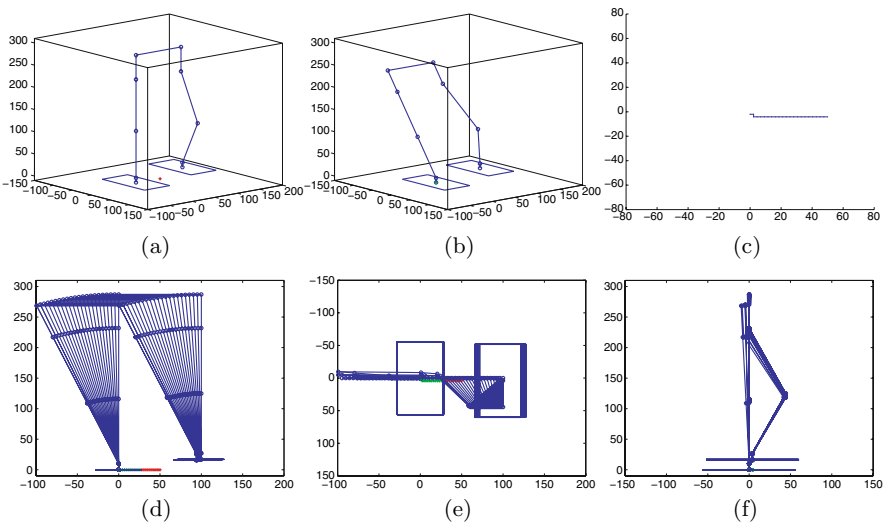
Fig. 4. Initial and final robot positions for a stabilization using the knee joints



(a) (b) (c)

Fig. 5. Robot stability with the left foot raised

Finally, Fig. 6 shows how robot tries to reach the stability during the initial phase of a single step. Figure 6a and 6b show the initial and final positions of the robot. Figure 6c shows the evolution of the COM projection. Basically, the robot rotates its right ankle to move the COM projection to the final position



(a) (b) (c)

(d) (e) (f)

Fig. 6. Robot stability in the initial phase of single step

$(0, 0)$. Figure 6d, 6e and 6f illustrate the robot consecutive movements in the frontal, transverse and sagittal planes, respectively.

6 Conclusions and Further Work

Stability of biped robot in the presence of disturbances generated by external forces or by the inherent change of the center of masses position when the robot walks has been solved by means of a bio-inspired method, in which the robot controller optimizes, in real time, the joints movements to keep the stability. The process is controlled through three parameters which define the target position of the center of masses projection and describe the desired posture to be adopted by the robot. The results of computer simulations have shown the viability of the proposed method.

The parameters commented above can be used in the generation of gaits to prepare the robot posture for a step, maintaining the stability. The next stage of our work will be to define a mechanism to generate gaits using this approach.

References

1. Maravall, D., De Lope, J. (2003) A Bio-Inspired Robotic Mechanism for Autonomous Locomotion in Unconventional Environments. In C. Zhou, D. Maravall, D. Ruan (eds.), Autonomous Robotic Systems, Physica-Verlag, Heidelberg, 263–292
2. De Lope, J. Maravall, D. (2003) Integration of Reactive Utilitarian Navigation and Topological Modeling. In C. Zhou, D. Maravall, D. Ruan (eds.), Autonomous Robotic Systems, Physica-Verlag, Heidelberg, 103–139
3. Maravall, D., De Lope, J. (2004) A biomimetic mechanism for collision avoidance by coordinating normal and tangent orientations. WAC/ISORA-2004
4. DeMers, D.E., Kreutz-Delgado, K.K. (1997) Solving Inverse Kinematics for Redundant Manipulators. In O. Omidvar, P. van Smagt (eds.), Neural Systems for Robotics. Academic Press, 75–112
5. Meystel, A.M., Albus, J.S. (2002) Intelligent Systems: Architecture, Design and Control. John Wiley & Sons, New York
6. Kun, A.L., Miller, W.T. (1996) Adaptative dynamic balance of a biped robot using neural networks. Proc. IEEE Int. Conf. on Robotics and Automation, 240–245
7. De Lope, J., González-Careaga, R., Zarraonandia, T., Maravall, D. (2003) Inverse kinematics for humanoid robots using artificial neural networks. EUROCAST-2003, LNCS 2809. Springer-Verlag, Berlin, 448–459

Parallel Manipulator Hip Joint for a Bipedal Robot

J. Hofschulte, M. Seebode, and W. Gerth

Institut für Regelungstechnik, Universität Hannover, Appelstr. 11, 30167 Hannover, Germany
http://www.irt.uni-hannover.de/

1 Introduction

By now at the institute of automatic control, the two-legged robot LISA (Legged intelligent service agent) is in completion. As a hip joint a spherical parallel manipulator with three degrees of freedom is used presumably for the first time inside a bipedal robot. This article describes the construction of the hip joint as well as the derivation of the analytical kinematic equations which describe the parallel manipulator. The article ends with a short presentation of the explored workspace for the real joint.

2 Composition of the Hip

The classic construction of a hip joint as a cardanian joint would be serial kinematic and would therefore have the usual disadvantages of a lower stiffness and precision. Since the movement of a parallel manipulator evolves from the coordinate interaction of several actors the driving forces and torques summate at least partly. For the production of a parallel manipulator the symmetry of the assembly is advantageous. A parallel manipulator offers additionally advantages especially in the hip joint of a two-legged robot since the drives do not have to be accelerated during the movement of a leg. Their masses are part of the torso and the robot corresponds closer to the inverted pendulum model [4].

The idea of the parallel manipulator hip joint was derived from the "Agile Eye" [1]: a spherical parallel manipulator for camera positioning. In [1, 2] and [3] Gosselin presents some analysis of the workspace as well as approaches for the derivations of the kinematic equations.

The "Agile Eye" consists of three identically built arms which are symmetrically ordered around the center C in 120° steps (see Fig. 1). Each arm consists of an angular bipartite outer arm and a passive inner arm. The bipartite outer arm which comprises an upper- and a forearm, is coupled at the

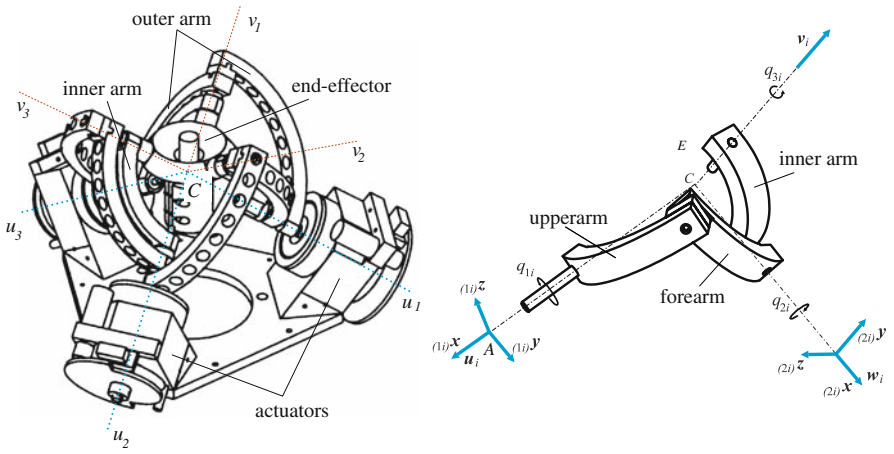


Fig. 1. *Left* "Agile Eye" from Gosselin [1] *right* Sketch of one arm of the hip joint

upper arm with the actuator rigidly and at the forearm rotatably with the inner arm. All inner arms are connected rotatably together with the end-effector of the parallel manipulator. In the hip joint the end-effector corresponds to the top of the thigh. All partial arms are circular arcs. At the hip joint both parts of the outer arms are connected rigidly together in an angle of approx. 70° in a way so the actuator u_i-axes is right angled with the w_i-axes of the bearing between the inner and outer arm. For the kinematic chain the subdivision in upper- and forearm is not important. The subdivision avoids a collision of parts.

All inner arms are connected rotatably around the v_i-axes to the end-effector. The inner arm fixes its axis v_i and w_i right angled. Finally the v_i-axes of the endeffector stand right angled to each other like the u_i-axes of the actuators do. In the following the end-effector is in its home-position when the u_i and v_i axes are aligned.

The construction of the hip joint has to emulate these degrees of freedom under consideration of the stiffness and the possibility to produce the parts. Therefore the hip joint is made almost completely out of aluminium. Including gears, bearings and drives, one hip-joint has a total weight of approx. 4.5 kg.

The outer arm, consisting of the upper and forearm, is manufactured out of one piece for stability and precision reasons. In a FEM-analysis its stiffness was tested with a load of approx. 1 kN and 190 Nm. This is similar to the maximum torque of the actuators and almost three times the total weight of the robot of 36 kg. Since the forces in the assembled hip are distributed between all three arms of the parallel kinematic this test-scenario is a worst case appraisal.

The gears are mounted onto the outer side of a bowl-like aluminium frame to assure the orthogonal orientation of the actuator axes u_i and for the main

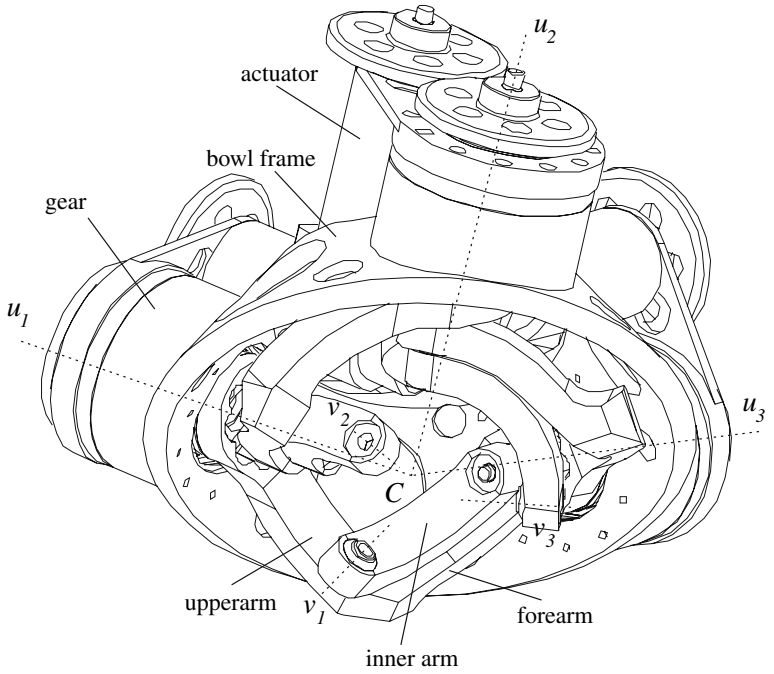


Fig. 2. Hip construction without thigh

stiffness of the structure. The engines were shifted out parallel of the actuator axes u_i and switched inside to get smaller outer dimensions and to get a leg distance of 18 cm between both centers C.

For the actuators 90W brushed-dc-motors and Harmonic Drive gears with a reduction of 100:1 are used by analogy to the robot BARt-UH which was developed at the institute earlier. These actuators can generate a maximum torque of approx. 80 Nm and a maximum speed of one rotation per second. The resulting torque at the end-effector can be simplified considered as a superposition of the actuator torques in dependency on the orientation of the thigh. If the end-effector is in its home position and a single torque around the thigh axes of 1 Nm is applied to the end-effector the actuators are all loaded with the same torque of $\frac{1}{\sqrt{3}}$ Nm. Applying a torque of 1 Nm which turns the thigh from its home position in the direction of an actuator axis u_j does not load the actuator j. In this case only two actuators are loaded each with a torque of $\frac{1}{\sqrt{2}}$ Nm.

3 Kinematic

Since the hip joint executes merely spherical movements and consists only of arc segments, the complete kinematic goes off within a sphere. All structures

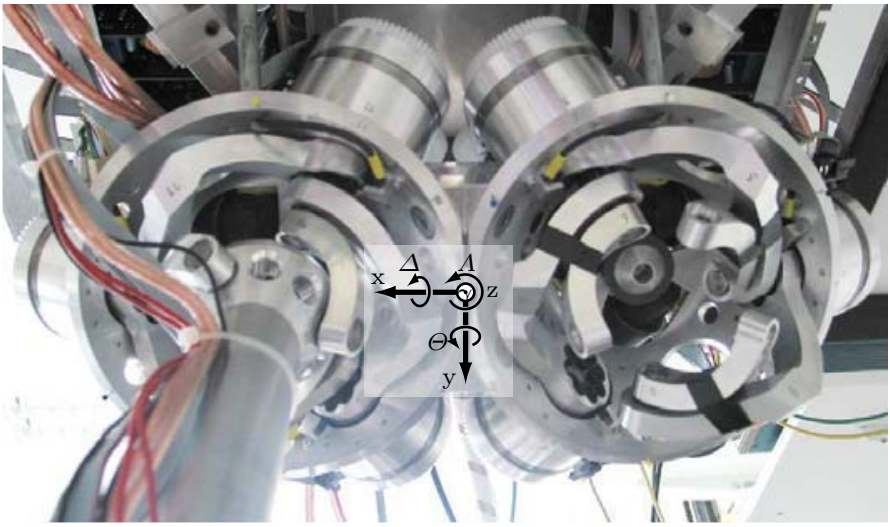


Fig. 3. Both hip joints mounted to the robot (*left thigh* unrigged)

can therefore be projected onto the unity sphere surface. Reducing the kinematic relations of the hip to the movements of the axes u_i, v_i and w_i this leads to the illustration in Fig. 4.

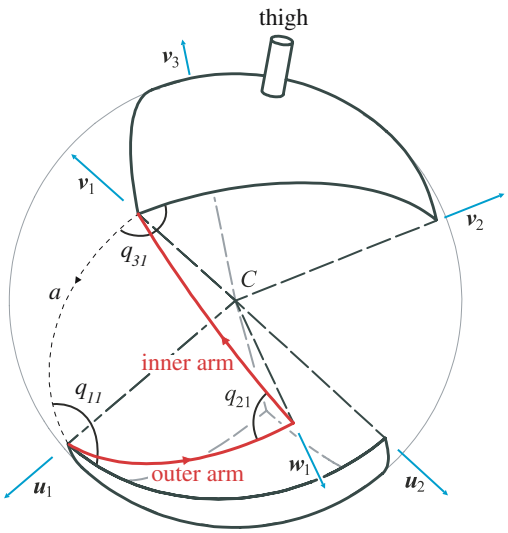


Fig. 4. Kinematic for the first actuator on a sphere

The connecting arc between u_1 and w_1 represents the kinematic relation of the outer arm. According to the construction this arc covers always 90°. The same applies for the arc between w_1 and v_1 which represents the kinematic relation of the inner arm. The angles q_{1i} are fixed by the actuator, whereas the angles q_{2i} and q_{3i} are passive degrees of freedom and therefore result by all kinematic constraints of the parallel manipulator. To derive the analytical inverse and direct kinematic equations it is advantageous to use spherical trigonometry.

3.1 Inverse Kinematic

The inverse kinematic equations describe the actuator angles $q_{1i} = f^{-1}(x)$ as a function of the thigh orientation $x = (\phi, \psi, \theta)$. ϕ, ψ and θ are the angles of the sequentiell rotation of the thigh around its x, y and z-axis (see Fig. 3). Due to the symmetry it is similar to derive the inverse kinematic equations for all actuators $i = 1, 2, 3$. So the following equations are presented for the first actuator only.

The spherical cosine law for the Euler-triangle, which is spanned by the trihedron of vectors $\triangle\,(u_1, v_1(x), w_1)$, leads to

$$\begin{aligned}\cos(a) = &\cos(\angle(u_1, w_1))\cos(\angle(v_1(x), w_1)) \\ &+ \sin(\angle(u_1, w_1))\sin(\angle(v_1(x), w_1))\cos(q_{21})\end{aligned} \tag{1}$$

in which $\angle(u_1, w_1)$ represents the angle between the vectors u_1 and w_1. With respect to $u_1 \perp w_1$ and $w_1 \perp v_1(x)$ this simplifys to $\cos(a) = \cos(q_{21})$. Since the scalar product of the unity vectors $v_1(x)$ and u_1 yields to $v_1(x) \cdot u_1 = \cos\left(\angle(v_1(x), u_1)\right)$ it is

$$\cos(a) = \cos(q_{21}) = v_1(x) \cdot u_1\ . \tag{2}$$

The spherical cosine law for the triangle $\triangle\,(u_1, v_1(x), u_2)$ results together with the accordant scalar product of $v_1(x)$ and u_2 as well as $u_1 \perp u_2$ to the similar expression $v_1(x)u_2 = \sin(\angle(u_1, v_1))\cos(q_{11})$. According to result (2) $\angle(u_1, v_1) = a = q_{21}$ this becomes $v_1(x)u_2 = \sin(q_{21})\cos(q_{11})$. With $\sin(q_{21}) = \sqrt{1 - \cos^2(q_{21})} = \sqrt{1 - (v_1(x) \cdot u_1)^2}$ one retrieves the inverse kinematic equation

$$\frac{v_1(x) \cdot u_2}{\sqrt{1 - (v_1(x) \cdot u_1)^2}} = \cos(q_{11}) \tag{3}$$

for the actuator angle q_{11}. Similar considerations to the triangle $\triangle(u_1, v_1(x), v_2(x))$ yield to

$$\frac{v_2(x) \cdot u_1}{\sqrt{1 - (v_1(x) \cdot u_1)^2}} = \cos(q_{31})\ . \tag{4}$$

Hence the analytic equations (3), (2) and (4) fully solve the inverse kinematic problem for the first arm.

3.2 Direct Kinematic

The direct kinematic equations describe the position of the thigh $x = f(q_{1i})$ by given actuator angles q_{1i}. To get the rotation between the thigh coordinates v and the base coordinates u it is necessary to know the distances between the u_i and $v_i(x)$ vectors of all arms. With respect to the inverse kinematic these distances correspond to the passive angles q_{2i}. Therefore it can be thought of a spherical shift joint between u_i and $v_i(x)$ with the spherical length q_{2i} (see Fig. 5).

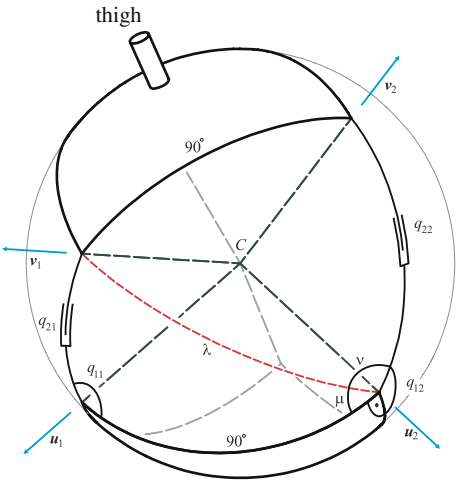


Fig. 5. Model with spherical shift joints

The spherical cosine law for the upper triangle $\triangle(v_1(x), v_2(x), u_2)$ in Fig. 5 provides

$$0 = \cos(\lambda)\cos(q_{22}) + \sin(\lambda)\sin(q_{22})\cos(\nu) \tag{5}$$

with the auxiliary variables ν and λ. For the lower triangle $\triangle(u_1, u_2, u_3)$ the spherical cosine law yields to $\cos(\lambda) = \sin(q_{21})\cos(q_{11})$ and the spherical sine law to $\sin(\lambda) = \sin(q_{11})\sin(q_{21})/\sin(\mu)$. Eliminating λ from these equations and doing some term remodelling leads to

$$0 = \cos(q_{11})\cos(q_{22}) + \sin(q_{11})\sin(q_{22})\frac{\cos(\nu)}{\sin(\mu)}\,. \tag{6}$$

The sum of angles around u_2 is $2\pi = \mu + \nu + q_{12} + \frac{\pi}{2}$, and with that it is $\cos(\nu) = \sin(-q_{12}-\mu)$ respectively $\cos(\nu) = -\sin(q_{12})\cos(\mu) - \cos(q_{12})\sin(\mu)$. This eliminates ν and leads to

$$0 = \cos(q_{11})\cos(q_{22}) + \sin(q_{11})\sin(q_{22})\left(\frac{-\sin(q_{12})}{\tan(\mu)} - \cos(q_{12})\right) \tag{7}$$

with the auxiliary variable μ. The spherical sine and cosine laws for the triangle $\triangle(u_1, v_1(x), u_2)$ again yield to $\cos(q_{21}) = \sin(\lambda)\cos(\mu)$ and $\cos(q_{21}) = \sin(q_{11})\sin(q_{21})/\tan(\mu)$. Substituting λ by previous results this expression eliminates μ again

$$0 = \cos(q_{11})\cot(q_{22}) - \sin(q_{12})\cot(q_{21}) - \sin(q_{11})\cos(q_{12})\ . \tag{8}$$

By exchanging the indices cyclically for the angles one retrieves the following system of equations for the q_{2i}

$$\begin{pmatrix} -\sin(q_{12}) & \cos(q_{11}) & 0 \\ 0 & -\sin(q_{13}) & \cos(q_{12}) \\ \cos(q_{13}) & 0 & -\sin(q_{11}) \end{pmatrix} \begin{pmatrix} \cot(q_{21}) \\ \cot(q_{22}) \\ \cot(q_{23}) \end{pmatrix} = \begin{pmatrix} \sin(q_{11})\cos(q_{12}) \\ \sin(q_{12})\cos(q_{13}) \\ \sin(q_{13})\cos(q_{11}) \end{pmatrix}\ . \tag{9}$$

With the knowledge of these angles the configurations of the arms and thus the pose of the thigh can be calculated.

4 Workspace Analysis

The equations found do not take care of mathematical or mechanical singularities neither consider any part-collisions. Nevertheless, these restrictions are particularly important for the path planning. A great mathematical effort is necessary to calculate the remaining workspace due to the part-collisions. The result is fundamental depending on the respective geometry of the arms.

For an appraised examination the workspace of the left hip joint was explored experimentally by a manual movement of the thigh. During the exploration a dot cloud is sampled in which each dot represents a possible configuration $x = (\phi, \psi, \theta)$ (see Fig. 3) of the end-effector. The whole dot cloud comprises nearly 19.000 different dots sampled every 0.1 seconds during the movement. Figure 6 shows the three dimensional dot cloud cutted into 20 equally spaced sections along the θ-axis. Each sub figure shows for a fixed orientation of θ (torsion of the thigh) the remaining possible configurations of ϕ (swinging the thigh for- and backward) on the abscissa and ψ (swinging the thigh sideward) on the ordinate. Due to the discretization each sub figure comprises the orientation of $\theta \pm 6°$. Since the manual movement of the thigh did not reach all possible positions during the examination the cloud is fragmentary and represents only an overview of the full configuration space.

Figure 6 shows a big and adequate symmetric workspace of the parallel manipulator hip joint. It shows a maximum torsion of $\theta \approx \pm 80°$ which is more than the human leg can do in this direction. The maximum angle forward/backward ϕ and sideward ψ are both in the home position $\theta = 0°$ approx. $\pm 60°$ what should be more than necessary for bipedal motions. To achieve an even better balance between forward and backward swinging the hip is mounted in an angle of approx. 30° inside the torso of the robot. So it is possible to move the thigh approx. 90° forward and approx. 30° backward which should be adequate for stair climbing and handling obstacles.

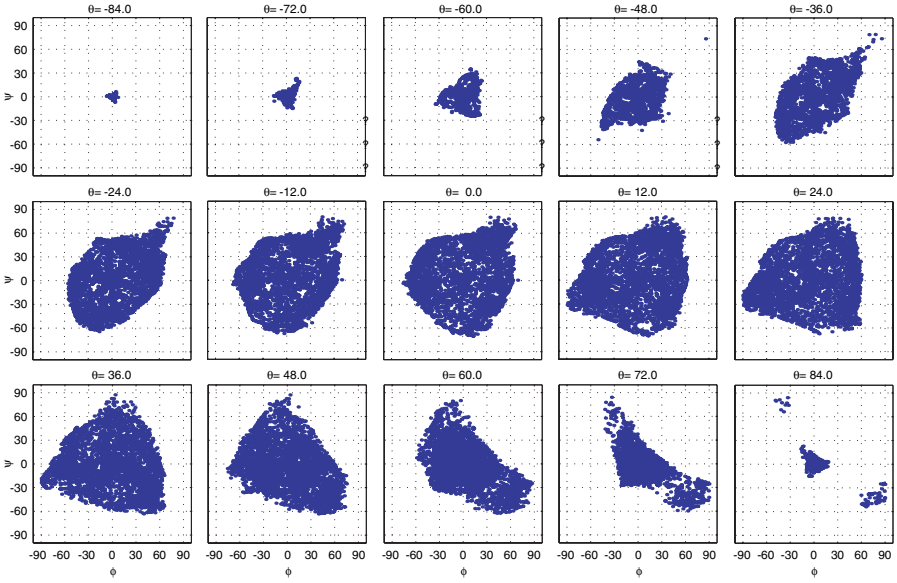


Fig. 6. Workspace of the left hip joint; all angles in degrees

5 Outlook

Besides the kinematic the dynamic equations were also derived analytically and applied to a model. Actually the bipedal robot LISA is assembled and tested. Now the research focuses on the control algorithms of the hip and the path planning for the bipedal robot. Hopefully LISA will walk with this new kind of hip within the next months.

References

1. C.M. Gosselin, E. St. Pierre, M. Gagné: *On the Development of the Agile Eye: mechanical design, control issues and experimentation*, IEEE Robots and Automation Society Magazin, Band 3, Nr. 4, S. 29–37, December 1996.
2. C.M. Gosselin, J.F. Hamel: *The Agile Eye: a high-performance three-degree-of freedom camera orienting device*, Proceedings of the IEEE International Conference on Robotics and Automation, S. 781–786, San Diego 1994.
3. C.M. Gosselin, J. Angeles: *The Optimum Kinematic Design of a Spherical Three-Degree-of-Freedom Parallel Manipulator*, Transactions of the ASME, Band 111, S. 202–207, Juni 1989.
4. S. Kajita, K. Tani: *Experimental Study of Biped Dynamic Walking in the Linear Inverted Pendulum Mode*, IEEE International Conference on Robotics and Automation, ICRA, Seiten 2885–2891, 1995

5. M. Seebode: *Regelung einer rotatorischen Parallelkinematik mit drei Freiheitsgraden*, project work, IRT, unpublished
6. A. Wedler: *Konstruktion einer Rotatorischen Parallelkinematik*, project work, IRT, unpublished

Gaits Stabilization for Planar Biped Robots Using Energetic Regulation

N.K. M'Sirdi, N. Khraief, and O. Licer

Laboratoire de Robotique de Versailles, Université de Versailles Saint Quentin en Yvelines, 10, Avenue de l'Europe 78140 Vélizy
nkms@free.fr
khraief@robot.uvsq.fr
oumnia@robot.uvsq.fr

Abstract. This paper is concerned with the walking of an underactuated biped robot on the level ground, imitating the passive walking on a given slope. First, we present the modelling of the biped robot (that is a 7 DOF robot). Then, we focus on the study of the almost passive dynamic walking. In particular, we show that the application of a nonlinear feedback control to stabilize the torso and knees leads the biped robot to perform a stable almost-passive dynamic walking when dealing with motions on a downhill a slope. More such control also leads the system trajectories to converge towards stable limit cycles. In this context, we present some results based on both Poincaré map method and trajectory sensitivity analysis to efficiently characterize the stability of the almost-passive limit cycles.

However, such limit cycles may not exist for all ground configurations especially when dealing with motions on the level ground. This involves to introduce some complementary control scheme. In this purpose, we present some theoretical and simulation results based on the use of recent control method (referred to as Controlled Limit Cycles), which considers the system energy for both controller design and system stabilization.

1 Introduction

Biped locomotion is one of the most sophisticated forms of legged locomotion. A biped robot is a two-legged robot, which may have knees and it most closely resembles human locomotion, which is interesting to study and is very complex. There has been extensive research over the past decade to find mechanical models and control systems which can generate stable and efficient biped gaits.

This paper deals with the use of stabilization of periodic gaits for control of planar biped robots. Indeed, we investigate the existence and the control of passive limit cycles.

The notion of obtaining passive gaits from mechanical models was pioneered by McGeer [5]. These gaits are only powered by gravity and are thus very energy efficient. In passive human walking, a great part of the swing phase is passive, i.e. no muscular actuation is used. Activation is primarily during the double contact phase and then it essentially turns off and allows the swing leg to go through like a double pendulum, McMahon [4]. Passive walking has been studied by several researchers, Collins, Garcia, Goswami [6], McGeer [5], M.W. Spong [8]. These have found stable passive gaits for shallow slopes where gravity powers the walk down the slope. However, it is found that no stable passive gaits exist for level ground. Consequently, a complementary control schemes is required. Thus, we will present some theoretical and simulation results based on the use of a recent control method (referred to as Controlled Limit Cycle [1–3]), which considers the system energy for both controller design and system stabilization.

Motivated by the work done so far [14, 15, 22, 23] and potential applications of biped locomotion, we conduct this paper on study of energy based control of biped robots. A new method is used to stabilize periodic cycles for such systems. The control objective is regulation of the system energy (using a nominal energetic representation) for stabilization of the walking robot with different speeds. The CLC control allows to establish this task.

We found that it is possible to develop energy mimicking control laws which can make the biped gaits stable and robust under various kind of environment. They can also be very energy efficient and make the biped gait anthropomorphic.

The paper will be organized as follows:

First we will present the modelling of the biped robot under consideration (that is a kneed robot with torso with 7 DOF). Then, we will focus on the study of the passive dynamic walking of this robot, on inclined slopes.

Moreover, we will show that under a simple control applied to stabilize a posture, trajectories of the biped robot can converge towards stable limit cycles. In this context, we will present some results based on Poincaré map method and trajectory sensitivity analysis to efficiently characterize the stability of the almost-passive limit cycles.

However, as such limit cycles may not exist for all ground configurations, a complementary control schemes is required. Thus, we will present some theoretical and simulation results based on the use of a recent control method (referred to as Controlled Limit Cycle [1–3]), which considers the system energy for both controller design and system stabilization. A very interesting idea is to use the torso and the slope to control and stabilize the robot at a desired speed.

Finally, some potential extensions for future works will be discussed.

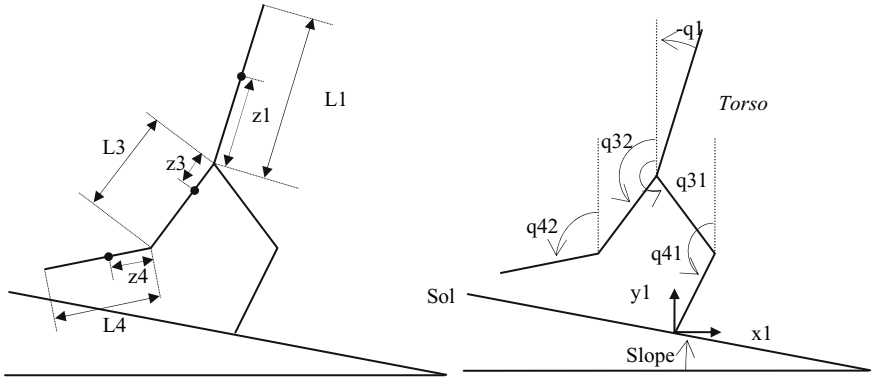


Fig. 1. A passive dynamic walking model of a 7 dof biped robot

2 The Biped Robot Model

The dynamic model of a simple planar biped robot is considered in this section. It's shown in Fig. 1. The robot has five degrees of freedom. It consists of a torso, two rigid legs, with no ankles and two knees, connected by a frictionless hinge at the hip. This linked mechanism moves on a rigid ramp of slope γ. During locomotion, when the swing leg contacts the ground (ramp surface) at heel strike, it has a plastic (no slip, no bounce) collision and its velocity jumps to zero. The motion of the model is governed by the laws of classical rigid body mechanics. It's assumed that walking cycle takes place in the sagittal plane and the different phases of walking consist of successive phases of single support. With respect to this assumption the dynamic model of the biped robot consists of two parts: the differential equations describing the dynamic of the robot during the swing phase, and the algebraic equations for the impact (the contact with the ground).

2.1 Swing Phase Model

During the swing phase the robot is described by differential equations written in the state space as follows:

$$\dot{x} = f(x) + g(x)u \tag{1}$$

Where $x = (q, \dot{q})$.

(1) is derived from the dynamic equation between successive impacts given by:

$$M(q)\ddot{q} + H(q, \dot{q}) = Bu \tag{2}$$

Where $q = (q_{31}, q_{32}, q_{41}, q_{42}, q_1), u = (u_1, u_2, u_3, u_4) : u_1$ and u_2 are the torques applied between the torso and the stance leg, and the torso and the swing leg, respectively, u_3 and u_4 are torques applied at both knees $M(q) =$

$[5 \times 5]$ is the inertia matrix and $H(q, \dot{q}) = [5 \times 1]$ is the coriolis and gravity term (i.e.: $H(q) = C(q, \dot{q}) + G(q)$) while B is a constant matrix. The matrices M, C, G, B are developed in [17].

2.2 Impact Model

The impact between the swing leg and the ground (ramp surface) is modelled as a contact between two rigid bodies. The model used here is from [18], which is detailed by Grizzle et al. in [16]. The collision occurs when the following geometric condition is met:

$$x_2 = z_2 tg\gamma \tag{3}$$

Where:

$$\begin{array}{l} x_2 = L_3(\sin q_{31} - \sin q_{32}) + L_4(\sin q_{41} - \sin q_{42}) \\ z_2 = L_3(\cos q_{32} - \cos q_{31}) + L_4(\cos q_{42} - \cos q_{41}) \ . \end{array} \tag{4}$$

Yet, from biped's behaviour, there is a sudden exchange in the role of the swing and stance side members. The overall effect of the impact and switching can be written as:

$$\begin{array}{l} h : S \rightarrow \chi \\ x^+ = h(x^-) \end{array} \tag{5}$$

Where: $S = \{(q, \dot{q}) \in \chi / x_2 - z_2 tg\gamma = 0\}$, with h is specified in []. The superscripts $(-)$ and $(+)$ denote quantities immediately before and after impact, respectively.

2.3 Overall Model

The overall 7-dof biped robot has as hybrid model:

$$\begin{cases} \dot{x} = f(x) + g(x)u & x^-(t) \notin S \\ x^+ = h(x^-) & x^-(t) \in S \end{cases} \tag{6}$$

This can be written

$$\begin{cases} \dot{x} = f(x, \Lambda_i)u + g(x, \Lambda_i), & \forall x(t) \notin C(e_i) \quad (7) \\ x^+ = h(x^-, e_i), & \forall x(t) \in C(e_i) \quad (8) \end{cases}$$

Where the discrete state set is $\Lambda = \{\mathrm{ss}_1, \mathrm{ss}_2\} = \{\Lambda_1, \Lambda_2\}$. C (e_i) is state change condition event after impact for the event e_i. The states are ss_1 (leg n° 1 is the support leg) and ss_2 (leg n° 2 is the support leg). The two events correspond to single supports on one leg and then the other. The commutation events or changes are in the set $E = \{e_1, e_2\} = \{(ss_1, ss_2), (ss_2, ss_1)\}$. We can consider now this model and transform the system by a first partial feedback, in order to allow existence of Limit Cycles corresponding to walking downhill a slope.

3 Almost Passive Dynamic Walking

3.1 Outline of Procedure

Our objective is to examine the possibility that a kneed biped robot with torso can exhibit a passive dynamic walking in a stable gait cycle, downhill a slope. In our previous work [14, 15, 22, 23] on a kneeless biped robot with torso, we have shown that such systems can steer a passive dynamic walking on inclined slopes, one time they have been transformed by a partial feedback control. Thus, the main idea of this work is to transform the 7-DOF biped robot in a system for which there exists limit cycles. The objective is then, to obtain nearly passive Limit Cycles, which will be exploited further to get a dynamic walking behaviour. This is possible when a torque is applied to stabilize the torso and knees at fixed positions. A partial feedback will be designed for, in the next section.

3.2 Non Collocated Input/Output Linearization

In this section we use some results from [9–11, 24], the objective is to get a control scheme able to stabilize the torso at a desired position.

To show this we may write the dynamic equations system (2) as:

$$M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + h_1 + \phi_1 = \tau_1 \tag{9}$$

$$M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + h_2 + \phi_2 = \tau_2 \tag{10}$$

Where $\theta_1 = q_1$ and $\theta_2 = (q_{31}, q_{32}, q_{41}, q_{42})^T$ and:

$$M(q) = \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \tag{11}$$

is the symmetric, positive definite inertia matrix, the vector functions $h_1(q, \dot{q}) \in R^1$ and $h_2(q, \dot{q}) \in R^4$ contain coriolis and centrifugal terms, the vector functions $\phi_1(q) \in R^1$ and $\phi_2(q) \in R^4$ contain gravitational terms and $(\tau_1, \tau_2)^T = Bu$ represents the input generalized force produced by the actuators, such that:

$$\tau_1 = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} u, \text{ and } \tau_2 = \begin{pmatrix} -1 & -1 & 0 & 0 \end{pmatrix}$$

From (9)–(10) we obtain:

$$M_{21}\ddot{\theta}_1 + h_2 + \phi_2 - \tau_2 = M_{22}\nu_2 \tag{12}$$

$$\ddot{\theta}_2 = \nu_2 \tag{13}$$

In order to steer some trajectories for the knees, we have to partition the vector ν_2 in two parts from (12–13) we obtain the following expression:

$$\begin{cases} M_{21}\ddot{\theta}_1 + h_1 + \phi_1 = -M_{22}\ddot{\theta}_2 = -M_{221}\ddot{\theta}_{21} - M_{222}\ddot{\theta}_{22} & (14) \\ \ddot{\theta}_2 = (\ddot{\theta}_{21}, \ddot{\theta}_{22}) = (\nu_{21}, \nu_{22}) \ ; \quad M_{22} = [M_{221};\ M_{222}] & (15) \end{cases}$$

As $rank(M_{221}(q)) = 1$ for all $q \in R^5$, the system (9)–(10) is said to be *strongly inertially coupled*. Under this assumption we may compute a pseudo-inverse matrix $M_{12}^{\dagger} = M_{12}^T(M_{12}M_{12}^T)^{-1}$ and define ν_{21} in (13):

$$\nu_{21} = -M_{221}^{\dagger}(M_{21}\nu_1 + h_2 + \phi_2 - \tau_2 + M_{222}\nu_{22}) \tag{16}$$

Or $\nu_1 \in \mathbb{R}^1$, and $\nu_{22} \in \mathbb{R}^2$ are additional controls chosen as follows:

$$\begin{aligned} \nu_1 &= \ddot{q}_1^d + k_{d1}(\dot{q}_1^d - \dot{q}_1) + k_{p1}(q_1^d - q_1) \\ \nu_{22} &= \begin{pmatrix} \ddot{q}_{41}^d + k_{d2}(\dot{q}_{41}^d - \dot{q}_{41}) + k_{p2}(q_{41}^d - q_{41}) \\ \ddot{q}_{42}^d + k_{d3}(\dot{q}_{42}^d - \dot{q}_{42}) + k_{p3}(q_{42}^d - q_{42}) \end{pmatrix} \end{aligned} \tag{17}$$

Note also that we can choose a desired trajectory to avoid hitting the ground, for example such as:

$$q_{42}^d = F_a \sin\left[\frac{\pi}{t_{\max}}t\right]^3, \quad q_{41}^d = q_{31} - \varepsilon \quad \text{and} \quad q_1^d = Cte$$

The actual control u is given by combining (14), (16) and (17), after some algebra as:

$$u = \widetilde{M}_{21}\nu_1 + \widetilde{N}\nu_{22} + \tilde{h}_2 + \widetilde{\phi}_2 \tag{18}$$

The non collocated linearization approach transfers the system (2) into two subsystems written as follows:

$$\begin{aligned} \dot{\eta} &= A\eta \\ \dot{z} &= s(\eta, z, t) \end{aligned} \tag{19}$$

Where $\eta = (\eta_1, \eta_2)^T = (q_1 - q_1^d, \dot{q}_1 - \dot{q}_1^d)^T$, $z = (z_1, z_2)^T = (q_2, \dot{q}_2)^T$,

$$A = \begin{pmatrix} 0 & 1 \\ -k_p & -k_d \end{pmatrix}^{\text{a}} \quad \text{and}$$

$$s(\eta, z, t) = \begin{pmatrix} z_2 \\ -M_{12}^{\dagger}(h_1 + \phi_1) - M_{12}^{\dagger}M_{11}(\ddot{q}_1^d - k_p\eta_1 - k_d\eta_2) \end{pmatrix} \tag{20}$$

We refer to the linear subsystem as η-subsystem. Accordingly the non linear subsystem in (19) is denoted as z-subsystem where $s(0, z, t)$ defines the zero dynamics relative to the output η_1.

[a]The matrix A must be Hurwitz.

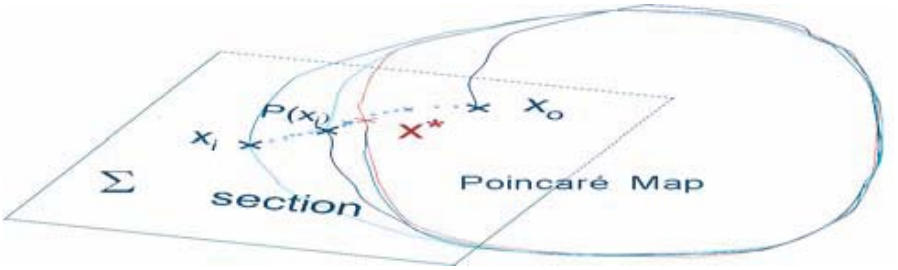


Fig. 2. Poincaré map

3.3 Finding Period One Gait Cycles and Step Period

After stabilizing the torso and the knees we proceed to simulate the motion of the biped robot downhill a slope. The walker's motion can exhibit periodic behaviour. Limit cycles are often created in this way. At the start of each step we need to specify initial conditions $(q, \dot{q})$ such that after T seconds (T is the minimum period of the limit cycle) the system returns to the same initial conditions at the start. A general procedure to study the biped robot model is based on interpreting a step as a Poincaré map. Limit cycles are fixed points of this function.

A Poincaré map samples the flow ϕ_x of a periodic system once every period [21]. The concept is illustrated in Fig. 2. The limit cycle Γ is stable if oscillations approach the limit cycle over time. The samples provided by the corresponding Poincaré map approach a fixed point x^*. A non stable limit cycle results in divergent oscillations, for such a case the samples of the Poincaré map diverge.

3.4 Gait Cycle Stability

Stability of the Poincaré map (20) is determined by linearizing P around the fixed point x^*, leading a discrete evolution equation:

$$\Delta x_{k+1} = DP(x^*)\Delta x_k \tag{21}$$

The major issue is how to obtain $DP(x)$ – The Jacobean matrix- while the biped dynamics is rather complicated; a closed form solution for the linearized map is difficult to obtain. But one can be obtained by the use of a recent generalization of trajectory sensitivity analysis [14, 15, 20, 21].

3.5 Numerical Procedure

A numerical procedure [14, 15, 20, 21] is used to test the walking cycle via the Poincaré map, it's resumed as follows:

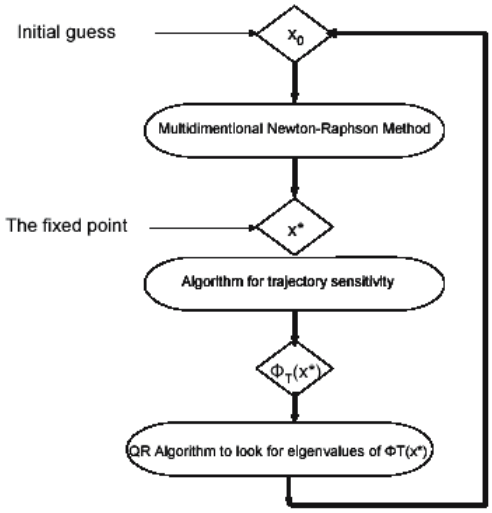


Fig. 3. Algorithm for the numerical analysis

1. With an initial guess we use the multidimensional Newton-Raphson method to determine the fixed point x^* of P^+ (immediately prior the switching event).
2. Based on this choice of x^*, we evaluate the eigenvalues of the Poincaré map after one period by the use of the trajectory sensitivity.

3.6 Simulation Results

Consider the model (1), with the following values:

We choose the hyper plane Σ as the event plane. We let the biped robot on a downhill slope with the control scheme (18). Starting with a suitable initial guess we obtain the following result:

$$\gamma = 0.075 \text{ rad}, q_1^d = -0.1 \text{ rad} ,$$
$$K_{p1} = 400, K_{\nu 1} = 120, K_{p2} = 500, K_{\nu 2} = 100, K_{p3} = 500, K_{\nu 3} = 100$$
we obtain
$$x^* = [3.188436; 3.042832; 2.857767; 2.825776; -0.100000;$$
$$- 0.838264; 0.691926; 0.818408; 0.441655; -1.149617]$$

Paramétres	TroncF	émur	Tibia
Masse kg	M_T 5 2	M_3 6.8	M_4 3.2
Longueurs m	L_T 0.5	L_3 0.4	L_4 0.4
Position des centres de masses m	Z_T 0.2	Z_3 0.16	Z_4 0.128
Premiers moments d'inerties $kg.m^2$			
$MY1$ 0.01			
$MZ1$ 4			
$MZ3$ 1.11			
$MZ4$ 0.41			
Moments d'inertiesd'ordre2 $kg.m^2$			
$XX1$ 2.22			
$XX3$ 1.08			
$XX4$ 0.93			

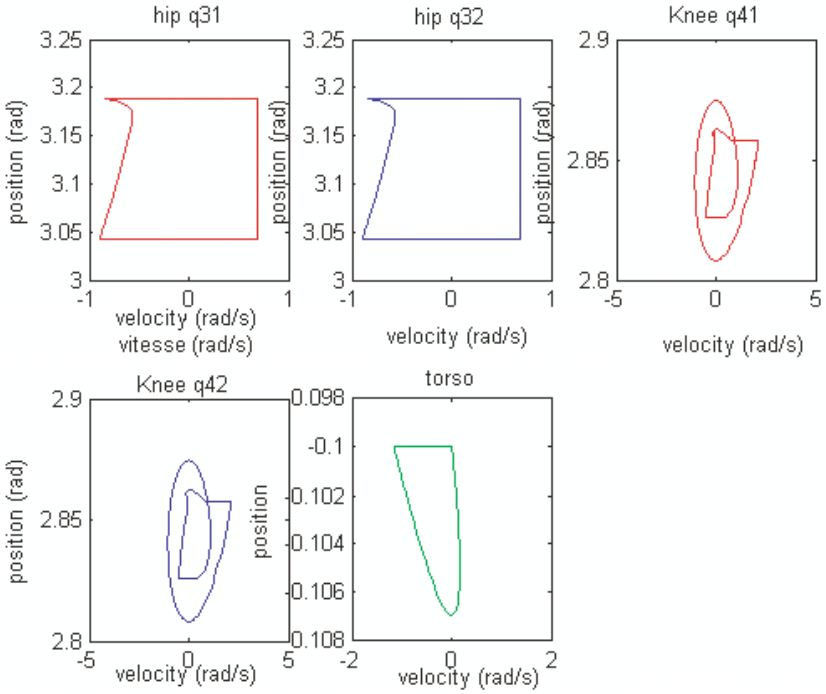


Fig. 4. Nearly passive limit cycles for the 7 dof biped robot

4 "CLC" for Active Dynamic Walking on the Level Groung

We saw the existence of *Almost-passive* limit cycles in Sect. 3. However these may not exist on all slopes, so some additional control is required. In this

section we present "CLC" (Controlled Limit Cycles) [1–3], which considers the system energy for both controller design and system stabilization.

4.1 "CLC" Control

In the state space $\eta = 0$ defines an attractive integral manifold for the system (7–8) and that the expression $\dot{z} = s(0, z, t)$ defines the zero dynamics relative to the output $q_1 - q_1^d$. In order to get a periodic walk of the biped robot on the level ground we will use an additional control which will drive the zero dynamic to a reference trajectory characterized by the energy obtained from the *Almost-passive* limit cycles.

We define a Lyapunov function as follows:

$$V = \frac{1}{2}(E - E_{ref})^2 \tag{22}$$

With E is the total energy of the system defined as follows:

$$E = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \int_0^t \dot{q}^T G(q)dt \tag{23}$$

and E_{ref} is the energy which characterizes the *Almost-passive* limit cycle.

$$\dot{V} = (\dot{E} - \dot{E}_{ref})(E - E_{ref}) \tag{24}$$

Where

$$\begin{aligned} \dot{E} &= \dot{q}^T Bu = \dot{q}^T B(u_{ref} + \overline{u}) \\ \dot{E}_{ref} &= \dot{q}^T Bu_{ref} \end{aligned} \tag{25}$$

$\overline{u}$ in (25) represents the additional control. We choose the nonlinear control law:

$$\overline{u} = -\Gamma B^T \dot{q} sign(E - E_{ref}) \tag{26}$$

Where $\Gamma > 0$. It can be shown that the manifold defined by $\{\eta = 0\} \cup \{E = E_{ref}\}$ is attractive for the closed loop system and all trajectories converge towards the *Almost-passive* limit cycle which is a stable cycle exhibiting a periodic walk motion of the biped robot.

4.2 Simulation Results

The control law (67.26) is applied to the system (7) on the level ground ($\gamma = 0$). The next figures show that the CLC control leads to a stable walking motion of the 7 dof biped robot imitating a nearly passive walking on a slope of $\gamma = 0.075$ rad.

In Fig. 5 we show that after some steps the robot converges towards the trajectories steered by the nearly passive walking.

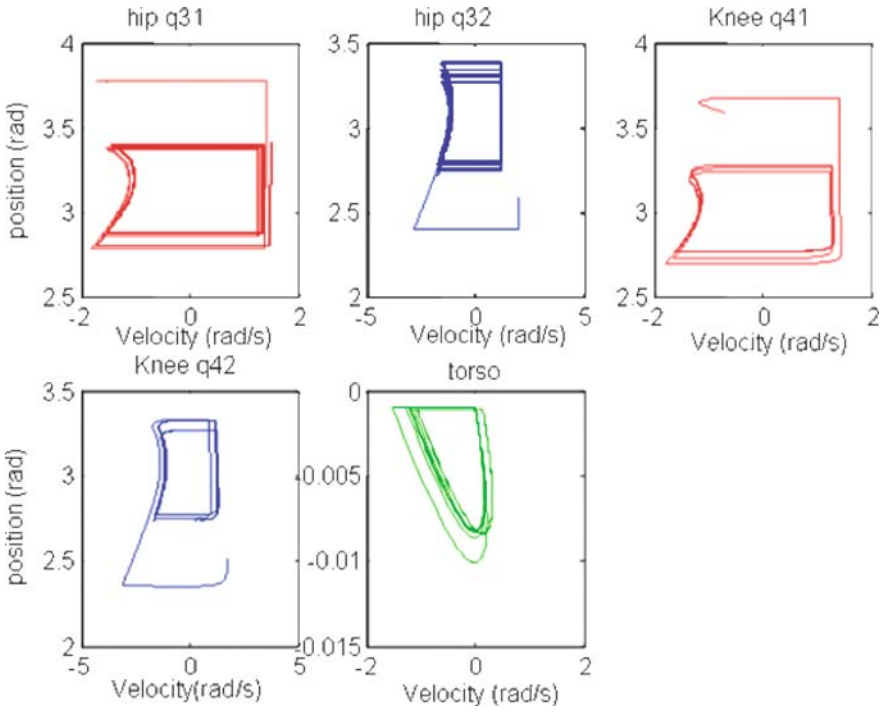


Fig. 5. Controlled limit cycles for different articulations

5 Conclusion

In this paper we present a control law which realizes a stable continuous walking on the level ground to imitate a controlled almost passive walking on the downhill slope. The last one is obtained by the use of a simple nonlinear feedback control with a numerical optimization. Next, the methodology developed uses *Controlled Limit Cycles* for stabilization of a periodic walk. The gaits correspond to limit cycles or region of attraction defined with the energy of the almost passive walking on the downhill slope. The approach is applied to a 7 DOF biped robot. Simulation results emphasize performance and efficiency of the proposed methodology.

References

1. N.K. M'Sirdi, N. Manamani and D. Elghanami, "Control approach for legged robots with fast gaits: Controlled Limit Cycles", Journal of intelligent and robotic systems, Vol. 27, no. 4, pp. 321–324, 2000.
2. N.K. M'Sirdi, N. Manamani and N. Nadjar-Gauthier "Methodology based on CLC for control of fast legged robots, Intl. Conference on intelligent robots and systems, proceedings of IEEE/RSJ, October 1998.

3. N.K. M'Sirdi, D. Elghanami, T. Boukhobza and N. khraief "Gait stabilization for legged robots using energetic regulation", accepted for IEEE/ICAR'2001.
4. S. Mochon, and McMahon, Ballistic walking: an improved model, Mathematical Biosciences, 52: 241–260, 1980.
5. T. McGeer, "Passive Dynamic Walking", the international journal of robotics research, 9(2), 62–82, April 1990.
6. A. Goswami, B. Tuillot, and B. Espiau "Compass like bipedal robot part I: Stability and bifurcation of passive gaits" INRIA Research report, No. 2996, 1996.
7. C. Mariano, A. Anindya, Ruina, and M. Coleman "The simplest walking model: Stability, complexity and scaling, 1998.
8. Mark, W. Spong, "Passivity based control of the compass gait biped" In: IFAC, world congress, 1999.
9. P.A. Chodavarapu and M.W. Spong "On Noncollocated Control of a Single Flexible Link," *IEEE Intl. Conf. on Robotics and Automation*, Minneapolis, MN, April, 1996.
10. M.W. Spong "The Control of Underactuated Mechanical System", *Plenary Address at The First International Conference on Mechatronics*, Mexico City, Jan. 26–29, 1994.
11. M.W. Spong "Partial Feedback Linearization of Underactuated Mechanical Systems", *IROS'94* , Munich, Germany, pp. 314–321, Sept. 1994.
12. F. Asano, M. Yamakita, and K. Furuta "Virtual passive dynamic walking and energy based control laws" Proc on int conf on intelligent robots and systems, vol. 2 pp. 1149–1154, 2000
13. M. Haruna, M. Ogino, K. Hosoda, and A. Minour, "Yet another humanoid walking-Passive dynamic walking with torso under a simple control–"Int, Conf on intelligent robots and system, 2001.
14. Nahla Khraief, N.K. M'Sirdi, M.W. Spong "Nearly passive dynamic walking of a biped robot", International Conference Signals, Systems, Decision & Information Technology, Mars, 2003.
15. N. Khraief, N.K. M'Sirdi, M.W. Spong "An almost passive walking of a kneeless biped robot with torso", European Control conference 2003 (ECC'03) University of Cambridge, UK: 1–4, September, 2003.
16. J.W. Grizzle, Gabriel Abba and Franck Plestan, Asymptotically stable walking for biped robots :Analysis via systems with impulse effects, IEEE TAC, 1999.
17. J.W. Grizzle, Gabriel Abba and Franck Plestan, "Proving asymptotic stability of a walking cycle for a five D.OF biped robot model". CLAWAR-99.
18. Y. Hurmuzlu and D.B, Marghilu Rigid body collisions of palanar kinematic chains with multiple contact points, the international journal of robotics research, 13(1): 82–92, 1994.
19. Ian A. Hiskens, Stability of hybrid system limit cycle : Application to the compass gait biped robot, conference on decision and control, Florida USA, December, 2001.
20. Ian A. Hiskens and M.A. Pai, Trajectory sensitivity analysis of hybrid systems, IEEE transactions on circuit and systems, 2000.
21. T. Sparker and L.O. Chua, Practical Numerical Algorithms for chaotic systems, Springer Verlag, New York, NY, 1989.
22. N. Khraief and N.K. M'Sirdi. Energy based control for the walking of a 7-DOF biped robot. To appear in a special issue of Acta Electronica en 2004' Advanced Electromechanical motion systems'.

23. N. Khraief, L. Laval, N.K. M'Sirdi. From almost-passive to active dynamic walking For a kneeless biped robot with torso. 6th International Conference on Climbing and Walking Robots (CLAWAR)-Italy September 17–19, 2003.
24. N.K. M'Sirdi, G. Beurier, A. Benallegue and N. Manamani, Passive feed-back control for robots with closed kinematic chains. IEEE-Syst. Man and Cyber. IMACS, CESA98, Hammamet, Tunisia, Avril, 1998.

Force Feedback Control Implementation for SMART Non-Linear Actuator

H. Montes, L. Pedraza, M. Armada, and T. Akinfiev

Control Department, Industrial Automation Institute, Madrid, Spain
hmontes@iai.csic.es

Summary. SMART is a non-linear actuator used in the humanoid biped robot SILO2. This actuator has been developed by the Industrial Automation Institute in order to spend less energy without losing kinematic skills. This paper presents kinematic parameters of hip joint in the lateral plane and the place where the force sensing is realized. The output and input torque are calculated with the measurement of force in one rod of SMART and the motor current, respectively; and they are compared in order to know the performance of this joint. Force control in the hip lateral joint was carried out. By means of this control, compliance movement and detection of obstacles when the robot completes a gait can be realized. Additionally with this force control the biped robot stability in the lateral plane is improved.

1 Introduction

SMART (Special Mechatronic Actuator for Robot joinT) non-linear actuator has been designed by the Industrial Automation Institute for driving joints in SILO2 biped robot [1–3]. Normally, biped robots using actuators with constant transmission ratios driven by electric motors present generally high energy consumption because the load is completely supported by the electric motor. Using other special actuators the efficiency is improved [4–7]. The authors of these actuators suggested the use of passive dynamics in the robot design, the use of motor-gear-spring to store energy in some part of the locomotion cycle, and the use of elastic fibbers or non-linear spring. In all cases, this authors design these kinds of actuators in order to decrease the energy consumption.

Several experiments were realized with SMART driving the ankle joint to demonstrate the real characteristics of this actuator [8, 9]. The relationship between the input and the output torques, measured experimentally, verified the theoretical function and it was possible to compare the savings of power expenditure related to classical constant transmission ratio actuators. These experiments showed that the energy consumption approximately decreases in a 48 percent with the use of SMART [9].

In SILO2 biped robot there are six different SMARTs, one in each ankle joint on the sagittal plane, one in each knee joint on the sagittal plane, and one in each hip joint on the lateral plane. In this paper the SMART placed in the hip joint is evaluated. First, the kinematic parameters of the actuator are presented in order to understand the non-linear characteristics of SMART; second, the output torque and the input torque are calculated using the measurements of force on the SMART rod and the motor current. When both, output and input torques, are compared the non-linear characteristic of the transmission is shown, and the pendular characteristic of output torque is presented. Finally, force control [10–12] was implemented in SMART in order to control the movement of the hip joint on the lateral plane sensing the force on the non-linear actuator. By means of this control could be possible to carry out smooth movements and also, to detect obstacles during the trajectory of the leg. The most important advantage in the use of this control is the possibility to improve the robot stability in the lateral plane using sensor fusion with ZMP detection.

2 Review of Hip Joint on Lateral Plane

The hip of the SILO2 has three degrees of freedom: one on the lateral plane, one on the sagittal plane, and one on the transversal plane. In the lateral plane, the hip is driven by SMART non-linear actuator, and the joints in the other planes are driven by classical transmissions actuators. The SMART mechanical drive has been implemented by using a four bar linkage mechanism, where individual links are of different lengths [1]. This four bar linkage is made up of two real bars (rod and crank), and two virtual bars [9] (see Fig. 1).

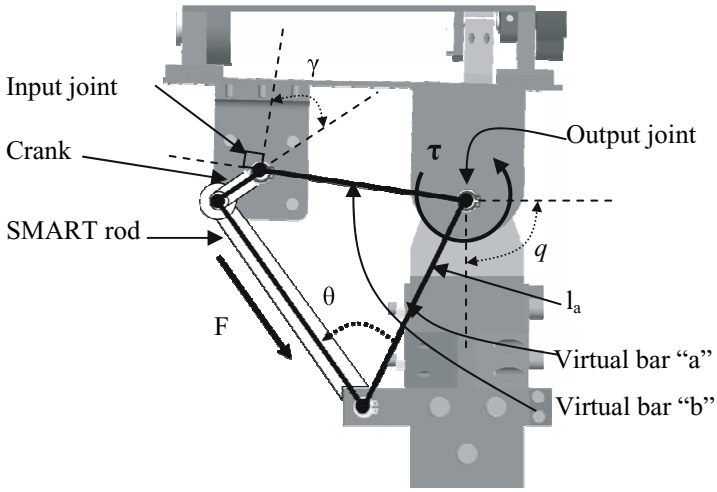


Fig. 1. Hip joint: SMART non-linear actuator and kinematics parameters

Also, the Fig. 1 shows the kinematic parameters of the hip actuated by SMART, where the output angle *"q"* is formed between the horizontal axis that crosses the hip joint and the axis parallel to the robot thigh; the input angle "γ" is formed between the crank prolongation and the perpendicular segment to the virtual bar *"b"*; "l_a" is the length of the virtual bar *"a"* which is the distance between the hip joint and the point where the force is applied; and the angle "θ" form between the SMART rod and the virtual bar *"a"*. This non linear actuator, as has been demonstrated by means of several experiments in [9], allows reducing power expenditure without losing output actuator velocity in the centre of the angular range of its trajectory.

The SMART drive kinematic parameters are determined by:

$$q = 2\arctan\left(\frac{a - \left(a^2 - b^2 + c^2\right)^{1/2}}{b + c}\right) - 2.63 \tag{1}$$

$$\theta = 2\arctan\left(\frac{a_1 + \left(a_1^2 - b_1^2 - c_1^2\right)^{1/2}}{b_1 + c_1}\right) - 2\arctan\left(\frac{a - \left(a^2 - b^2 + c^2\right)^{1/2}}{b + c}\right) \tag{2}$$

$$\tau = F \cdot l_a \cdot \sin(\theta) \tag{3}$$

Where,

$$\begin{aligned} a &= 0.473 - \sin(1.453 + \gamma)\ ; \\ b &= -1.523 - 0.9\cos(1.453 + \gamma) + 0.106\sin(1.453 + \gamma)\ ; \\ c &= -4.018 - \cos(1.453 + \gamma)\ ; \\ a_1 &= -a; \quad c_1 = -c\ ; \\ b_1 &= -2.166 - 0.824\cos(1.453 + \gamma) + 0.097\sin(1.453 + \gamma) \end{aligned}$$

The SMART actuator is characterised by the change of its reduction ratio from some value near to the centre of the trajectory (3.93 in the hip joint), up to infinitum in the extreme positions, when the output angle is –73.13° or –106.35° (according to left hip).

3 Dynamic Behaviour

The dynamic model for one degree of freedom with a non-linear transmission ratio has some differences with respect to a classical transmission dynamic model. The dominant terms of the mechanical model of SMART are mainly the inertia of the motor, the inertia of the gearbox motor and the inertia of the four bar linkage reflected in the motor axis. In addition, it is necessary to consider the friction in all cases [3].

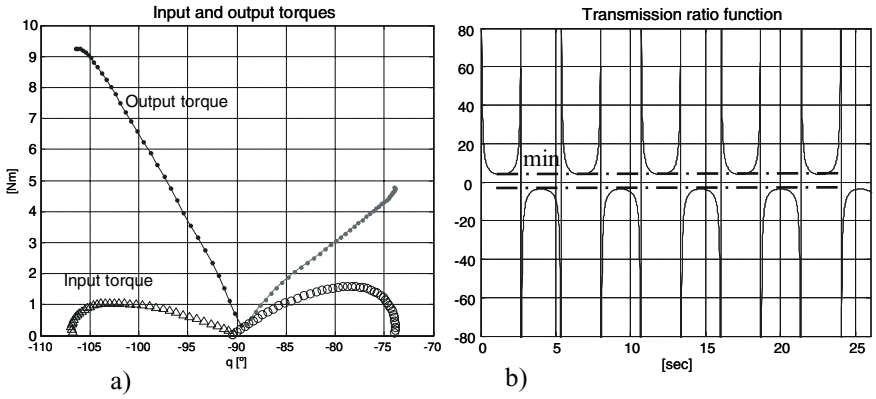


Fig. 2. Dynamic characteristics of hip joint actuated by SMART; (**a**) Comparison between input and output torque; and (**b**) transmission ratio function

Several experiments have been realized applying different velocity control strategies to the lateral hip joint. By means of these experiments it was possible to obtain the input and the output torques (see Fig. 2(a)), and the transmission ratio function obtained from real data (see Fig. 2(b)).

Figure 2a shows the absolute value of input and output torques. It is possible to observe that output torque has a pendular performance and it is similar to the torque obtained from a classic transmission actuator. Input torque has a nonlinear behaviour as it is shown in the Fig. 2a, where the torque tends to diminish near the extreme positions of the leg trajectory. If a greater load was applied during the leg movement, the input torque would have the same decreasing characteristic in the extremes positions, but the output torque would tend to increase proportional way to the applied load.

Figure 2(b) shows the transmission ratio function, where the positive values represent the movement in one direction and the negative values, in the other direction. In this SMART actuator de minimum value of the transmission is 3.93 calculated theoretically and this value has been verified experimentally.

4 Force Control and Experimental Results

There are several possibilities for controlling SMART drive. Up to this moment several non linear techniques have been used to control this actuator. For example in [3] a cascade robust control structure is used, characterized by a robust nonlinear regulator of variable structure in the internal loop and by a synthesized robust linear regulator by means of techniques of $\mathcal{H}_\infty$ in the external loop.

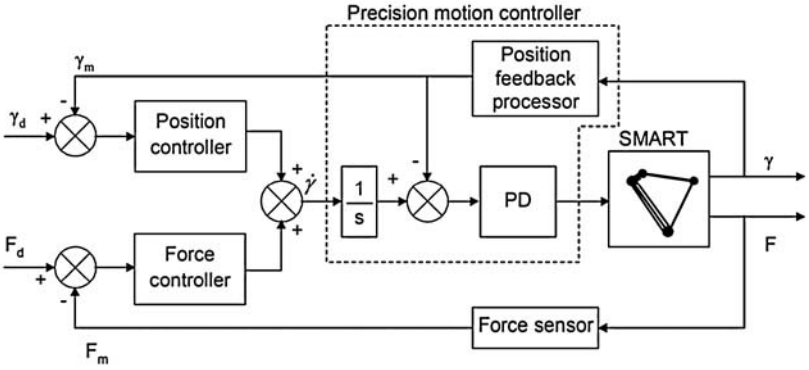


Fig. 3. Block scheme of position/force control

In this paper a position/force control is presented. Using this control it is feasible to carry out tasks involving interaction with the environment. Figure 3 shows the control scheme used in this approach.

This scheme of control has one motion controller that realizes an internal position loop with PD control, and two external loops of force and position. It can be noticed that the position error and the force error (in the external loop) are combined to provide the control signal.

Several experiments were realized with force/position control. Figure 4 shows some experimental results. The leg was moved (with this control) in a pendular movement in the lateral plane. When one force was applied to the leg, the force control dominated to the position control and the leg was displaced proportionally to the force applied. When the force was removed the position control dominated and the leg returned to de original position. The applied force was between 150 N and –150 N. This force is considered positive when the movement which produces corresponds to a decrement in the angular position q.

5 Conclusions

SMART non-linear actuator offers some advantages for driving biped robots, but in contrast, its use implies some additional complexity. When adding extra sensitivity to the SMART using a force sensor [9], it was possible to obtain the dynamic characteristics from real measured data, and it was demonstrated that there is good correspondence with the theoretical results. Also, position/force control was implemented in the hip lateral joint. When the force was applied to the leg, it was displaced in a proportional way; and when the force was released, the leg returned to the original angular position with the help of position control. The next steps will be focused on the use of force

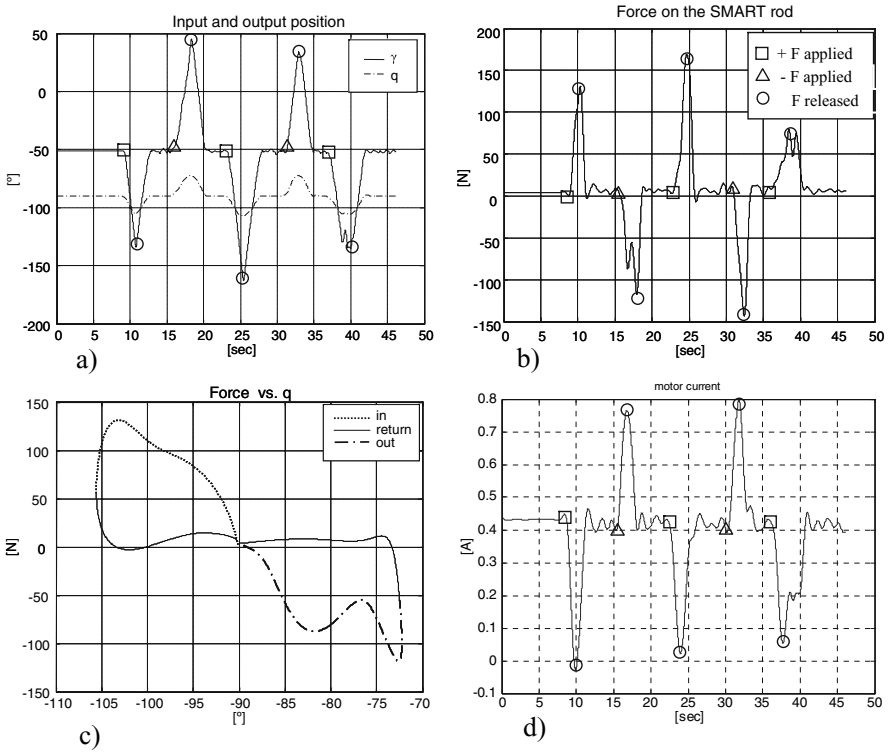


Fig. 4. Experimental results when disturbing forces are applied: (**a**) comparison between input and output position; (**b**) force measurement on SMART rod; (**c**) relationship between of the force with output angle; and (**d**) current in the motor winding

control to accomplish strategies of detecting obstacles and improving the robot stability in the lateral plane doing sensor fusion with ZMP detection.

Acknowledgments

The authors would like to acknowledge the funding form Consejeria de Educacion y Cultura de Madrid, under research projects 07T/0040/1998 and 07T/0022/2001.

References

1. Akinfiev, T., Armada, M., Caballero, R. (2000) Actuador para las piernas de un robot caminante. Spanish Patent Application

2. Caballero, R., Akinfiev, T., Montes, H., Manzano, C., Armada, M. (2002) Design of SMART actuated ROBICAM biped robot. In: Proc. of Intl. Conf. of Climbing and Walking Robots. Professional Engineering Publishing, Paris, pp. 409–416
3. Caballero, R. (2002) Control of non-linearly actuated biped robots. PhD thesis, Polytechnic University of Madrid, Spain
4. McGeer, T. (1990) Passive dynamic walking. Intl. J. of Robotic Research. 9(2): 62–81
5. Mennito, G., Buehler, M. (1997) LADD transmissions: design manufacture, and new compliance models. ASME J. of Mechanical Design. 119(2):197–203
6. Pratt, G., Williamson, M. (1997) Elastic actuator for precise force control, US Patent US5650704
7. Yamaguchi, J., Takanishi, A. (1997) Development of a biped walking robot having antagonistic driven joints using nonlinear spring mechanism. In: Proc. of the 1997 IEEE Intl. Conf. on Robotics and Automation, Albuquerque, New Mexico
8. Montes, H., Pedraza, L., Armada, M., Caballero, R., Akinfiev, T. (2003) Force sensor implementation for enhanced responsiveness of SMART non-linear actuator. In: Proc. of Intl. Conf. on Climbing and Walking Robots. Professional Engineering Publishing, Catania, Italy, pp. 887–894
9. Montes, H., Pedraza, L., Armada, M., Akinfiev, T., Caballero, R. (2004) Adding extra sensitivity to the SMART non-linear actuator using sensor fusion. Industrial Robot: An Intl. Journal 31(2):179–188
10. Whitney, D. (1987) Historical perspective and state of the art in robot force control. The Intl. J. of Robotics Research 6(1):3–14
11. Chiaverini, S., Sciavicco, L. (1993) The parallel approach to force/position control of robotics manipulators. IEEE Transactions on Robotics and Automation 9(4):361–373
12. Grieco, J., Armada, M., Frenandez, G., Gonzalez de Santos, P. (1994) A review on force control of robot manipulators. J. Informatics and Control 3(2–3):241–252

User Friendly Graphical Environment for Gait Optimization of the Humanoid Robot Rh-0

M. Arbulú, I. Prieto, D. Gutiérrez, L. Cabas, P. Staroverov, and C. Balaguer

Robotics Lab, Department of Systems Engineering and Automation, University Carlos III of Madrid, Avda. Universidad 30, 28911, Leganés, Madrid – Spain
marbulu@ing.uc3m.es

Abstract. The objective of this paper is to present the user friendly graphical environment for gait optimization of the humanoid robot Rh-0 that is currently under development at the Robotics Lab of the University Carlos III of Madrid. The robot has 21 DOF and is 1.35 m tall. The main objective of the developed environment is to solve the kinematics problem of high DOF humanoid robots: forward kinematics, inverse kinematics and path generation. This gait is computed taking into account that the minimal torques of the robot's joints' will permit to use the smallest actuators. The graphical environment permits to test different gaits in order to get smooth walking in any direction, satisfies the stability, movement ranges, speed and torque requirements. The first step consists of the selection of phases of robot's gait. The classical Denavit-Hartenberg based method for solving the forward kinematics problem is used. To solve the inverse kinematics the 3D geometric methods considering the hip and foot position are developed. The generated robot's path depends on the gait's length and frequency, etc. The paths generation controls the stable ZMP position in every moment.

Key words: humanoid robot, biped walking, ZMP, forward and inverse kinematics.

1 Introduction

There are several humanoid robots developed since 1970. The Wabot-1 [1] from Waseda Univ. that could walk, speak, and recognize voice commands; more abilities couldn't be implemented because of the absence of adequate computers; after that MIT developed COG [2], it couldn't walk but possessed advanced arm mobility and intelligence. Further on, in 1986 HONDA Mo. Co. Inc. developed the ASIMO robot [3], it's currently being the most advanced humanoid robot; it can walk as a human, in any direction, climb stairs, recognize faces, voice and gesture commands, etc. The last humanoid developed is the HRP-2 [4], additionally it can lay down and get up from the floor, sit in a

Japanese way and has a wider leg joints' range and can recognize voice commands, too; HRP-2 was designed for exterior human environments and also may lift and transport loads with human cooperation; ASIMO was designed for interior ones. The mobility of those robots is the starting point of their specifications. This paper deals with mobility aspects on humanoid robots taking into account stable motion during the walking cycle and minimal torques on each joint [5]. The paper is organized as follows. In the second section, Rh-0 specifications are described; in the third one, are the motion considerations; next, forward kinematics is solved. In the fifth section, the inverse kinematics is developed; after that walking patterns generation is described, taking into account the ZMP criterion [6] to assure stable motion; in the seventh section the results and the simulation environment are shown and discussed. Finally, the conclusions are presented.

2 Humanoid Robot RHO

The humanoid robot RH0 has 21 degrees of freedom (Figs. 1 and 2, and Tables 1 and 2), its height is 1.35 m and the weight is 45 kg.

HUMANOID ROBOT RHO

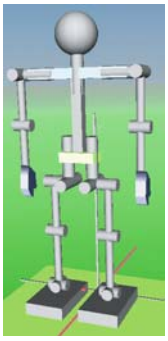


Fig. 1. Rh-0 structure

Table 1. RH0 Specifications

D.O.F.	Head		-	
	Arms	4 DOF/Arm (x2)	8	
		Shoulder 2 (x2)		(R, P)
		Elbow 1 (x2)		(P)
		Wrist 1 (x2)		(Y)
	Hands 1 (x2)		2	(open/close)
	Torso		1	(Y)
	Legs	6 DOF/Leg (x2)	12	
		Hip 3 (x2)		(R, P, Y)
		Knee 1 (x2)		(P)
		Ankle 2 (x2)		(R, P)
	Total		***21***	w ithout hands
Dimensions	Height		1350	mm
	Wide		300	mm
	Deep		230	mm
	Arm length		200	mm
	Forearm length		200	mm
	Hand length		100	mm
	Femur length		276	mm
	Tibia length		276	mm
	Ankle heght		60	mm
	Foot		330x200	mm

Fig. 2. Rh-0 design

Table 2. Rh-0 joint angle range

Joint		Axis	Range (°)
Head		R	N.E.
		P	N.E.
		Y	N.E.
Arm	Shoulder	R	-95° a 0°
		P	-180° a 60°
		Y	N.E.
	Elbow	P	-90° a 0°
		Y	N.E.
	Wrist	R	N.E.
		P	N.E.
		Y	-90° a 90°
	Hand	P	-16.5° a 60°
Torso		R	N.E.
		P	N.E.
		Y	-45° a 45°
Leg	Hip	R	-12° a 12°
		P	-80° a 80°
		Y	-15° a 15°
	Knee	P	0° a 180°
	Ankle	R	-12° a 12°
		P	-20° a 20°

3 Motion Considerations

Two kinds of motion are developed: Forward-backward motion and Rotational motion to both sides.

The forward-backward motion considerations are as follows:

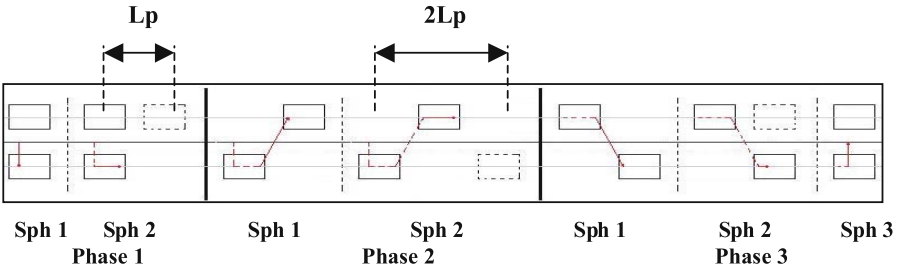


Fig. 3. Forward-backward motion phases

- Two kinematics models are developed: double support (kinematics chain closed) and single support (kinematics chain open)
- The motion is divided into three phases: start motion, walking motion and return to repose motion (Fig. 3).
- It is a cyclic motion, for that it could be characterized by two parameters: frequency (f_p) and step length (L_p).
- The hip should always stay parallel to the ground
- The legs should stay flexing on the rest position, for the adequate torque distribution between the ankle, the knees and the hip, caused by the robot weight.
- The main reference trajectories are: the hip and the foot on air trajectories. The hip trajectory defines the robot's centre of gravity motion, therefore it defines its stability, too. Otherwise, the foot on air trajectory defines its advance.

The rotational motion is divided into three phases too. The same considerations are assumed but L_p (step length) is changed for θ_p (rotation angle).

4 Forward Kinematics

Denavit-Hartenberg method has been used for solving forward kinematics (Fig. 4). So, the humanoid has been divided into four single manipulators; the hip link is their mobile base (Fig. 5). This way, X, Y, Z axes are the world's coordinate system and x', y', z' are the hip one.

5 Inverse Kinematics

Geometric methods have been used because those provide a closed solution. Legs' kinematics study will be described.

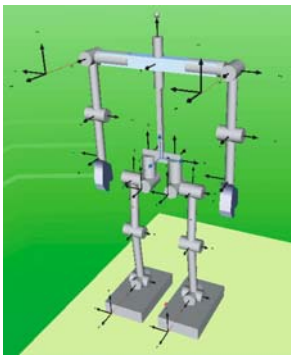


Fig. 4. D-H parameters

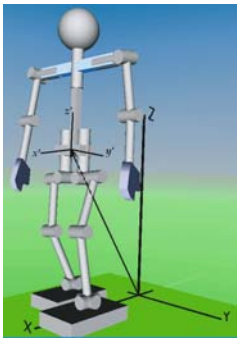


Fig. 5. Hip and world coordinate systems

Considerations:

- Each leg moves in a different plane.
- For the support leg we only need the hip position.
- For the leg on air we need the hip and foot on air positions.

The used nomenclature is the following one: q_i^j (joint angle i, on phase j); T_{bi} (Balance time of phase i); T_{si} (Advance time of phase i); T_i (Time duration of phase i $(T_{bi} + T_{si})$); L_f (Femur length); L_t (Tibia length).

Inverse kinematics in each phase has been developed. Each phase has two sub phases: Balancing and Walking ones.

Phase 1. Start motion

Sub phase 1. Balancing $(0 < t < T_{b1})$
Sub phase 2. Walking $(T_{b1} < t < T_{b1} + T_{s1})$

Phase 2. Walking motion

Sub phase 1. Balancing $(T_1 < t < T_{b2})$
Sub phase 2. Walking $(T_1 + T_{b2} < t < T_1 + T_{b2} + T_{s2})$

It is the most general phase and we will describe its equations (see (1) to (12), and Fig. 6 and Fig. 7).

$$
\begin{aligned}
&Support \ldots leg \\
&\ldots Frontal..motion \\
q_2 &= q_2^2(T_{b2}) && (1)\\
q_6 &= -q_2 && (2)\\
&\ldots Sagital..motion \\
q_4 &= \pi - a\cos\left(\frac{L_t^2 + L_f^2 - u_{g1}^2 - w_{g1}^2}{2L_tL_f}\right) && (3)\\
q_3 &= atg\left(\frac{u_{g1}}{w_{g1}}\right) + atg\left(\frac{L_f\sin(q_4)}{L_t + L_f\cos(q_4)}\right) && (4)\\
q_5 &= q_3 - q_4 && (5)\\
&\ldots Transversal..motion \\
q_1 &= 0 && (6)\\
&Leg..on..air \\
&\ldots Frontal..motion \\
q_{12} &= q_{12}^2(T_{b2}) && (7)\\
q_8 &= -q_{12} && (8)\\
&\ldots Sagital..motion \\
q_{10} &= \pi - a\cos\left(\frac{L_t^2 + L_f^2 - u_p^2 - w_p^2}{2L_tL_f}\right) && (9)\\
q_9 &= atg\left(\frac{u_p}{w_p}\right) + atg\left(\frac{L_t\sin(q_{10})}{L_{f+}L_t\cos(q_{10})}\right) && (10)\\
q_{11} &= q_9 - q_{10} && (11)\\
&\ldots Transversal..motion \\
q_7 &= 0 && (12)
\end{aligned}
$$

Phase 3. Return to repose motion

Sub phase 1. Balancing ($T_2 < t < T_2 + Tb_{31}$)
Sub phase 2. Walking ($T_2 + T_{b31} < t < T_2 + T_{b31} + T_{s3}$)
Sub phase 3. Balancing ($T_2 + T_{b31} + T_{s3} < t < T_2 + T_3$)

6 Walking Patterns Generation

Balancing and walking sub phases have been studied. In the balancing one, the hip trajectory has been generated in order to move the ZMP to the support

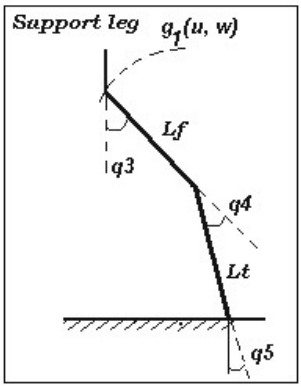


Fig. 6. Support leg in its plane

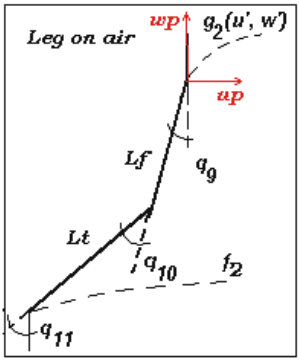


Fig. 7. Leg on air in its plane

foot. In the walking sub phase, the hip and the foot on air trajectories were generated in order to get walking. In both cases the hip trajectory keeps the ZMP in a stable position. The boundary conditions to generate those trajectories are: L_p (Step length), f (Step frequency), h (Max height of foot on air), ρ (rest knee angle), δ (max frontal ankle angle).

The next ((13) to (15), Figs. 8 and 9) describe the trajectories generated for phase 1: Start motion Sub phase 1: Balancing

$$g_2 = \left\{ \begin{array}{l} x = 0 \ldots\ldots\ldots\ldots\ldots\ldots \\ y = V_b(t_b - \frac{1}{2\pi f_b}\sin(2\pi f_b t_b) \\ z = \sqrt{R^2 - y^2} \ldots\ldots\ldots\ldots \end{array} \right\} \tag{13}$$

Sub phase 2: Walking

$$g_1(u, v, w) = \left\{ \begin{array}{l} u = \frac{V_f}{2}(t_s - \frac{1}{2\pi f_s}\sin(2\pi f_s t_s)) \\ v = 0 \ldots\ldots\ldots\ldots\ldots\ldots \\ w = \sqrt{R_1^2 - u^2} \ldots\ldots\ldots\ldots \end{array} \right\} \tag{14}$$

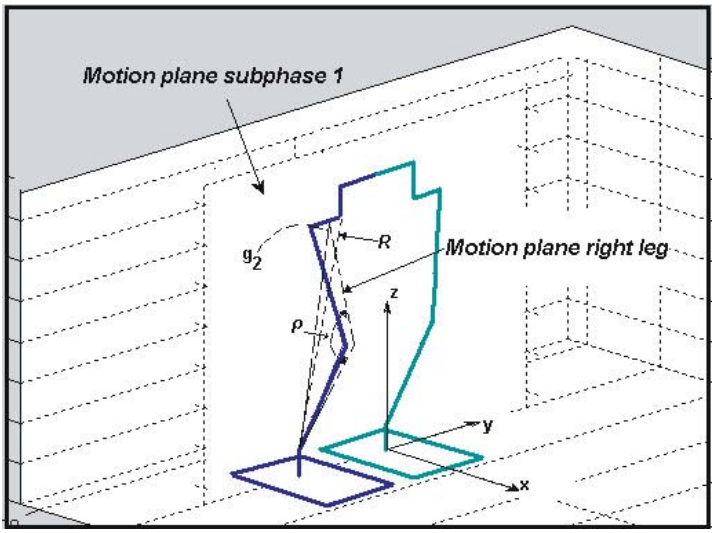


Fig. 8. Balancing

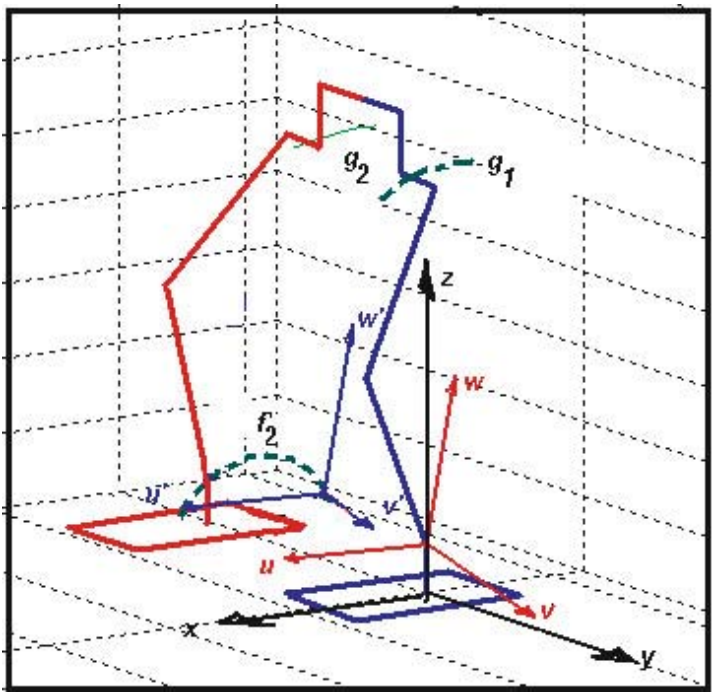
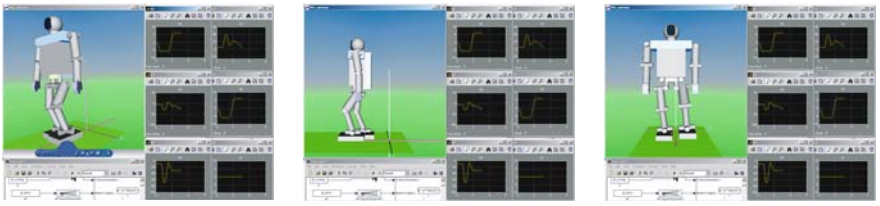


Fig. 9. Walking



(a) **(b)** **(c)**

Fig. 10. Simulation environment (**a**) Perspective, (**b**) Sagital, (**c**) Frontal view

$$f_2(u', v', w') = \left\{ \begin{array}{l} u' = V_f(t_s - \frac{1}{2\pi f_s}\sin(2\pi f_s t_s)) \\ v' = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ w' = \frac{h}{2}(1 - \cos(2\pi f_s t_s)) \ldots\ldots \end{array} \right\} \quad (15)$$

Other phases' trajectories had been generated in the same way.

7 Results

The simulation results are showed in Fig. 10 (like V-HRP [7]); torque diagrams allow us to select the adequate actuator. In order to get the smallest actuators the adequate gait was selected by using an iterative method. The results were: $L_{pmáx} = 170$ mm, $\rho = 170°, \delta = 6°$.

8 Conclusion

In this work, we have described our method for kinematics solving of the humanoid robot Rh-0. This method has the following contributions:

1. We get to test the joint trajectories before applying them on the humanoid. This way we can see how it would walk.
2. Otherwise, we could optimize the humanoid's gait in order to select the smallest and cheapest actuators.
3. This method was validated in a simulation environment that showed us rotation, speed, acceleration, and torque of each humanoid's joint, and ZMP.

A real-time control based on sensor feedback is needed to deal with uncertainties in actual surfaces. Our future study will focus on how to combine the planned walking pattern proposed in this paper and real-time control.

References

1. Kato I., Ohteru S., Kobayashi H., Shirai K., Uchiyama A., "Information-power machine with senses and limbs (WABOT – 1)", 1st CISM-IFTOMM Symposium

on theory and practice of Robots and Manipulators, Udine, Italy, 5–8 sept. 1973; pp. 11–24
2. Brooks R.A., "The COG project", Journal of Robotic Society of Japan, vol. 15, no. 7, 1997
3. Sakagami Y., Watanabe R., Aoyama C., Matsunaga S., Higaki N., Fujimura K., "The intelligent ASIMO: System overview and integration", Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots & Systems, EPFL, Lausanne, Switzerland, Oct. 2002, pp. 2478–83
4. Kaneko K., Kanehiro F., Kajita S., Yokoyama K., Akachi K., Kawasaki T., Ota S., Isozumi T.; "Design of Prototype Humanoid Robotics Platform for HRP", Proceedings of the 2002 IEEE, International Conference on Robotics & Automation, Washington, DC-May 2002; pp. 2431–2436
5. Huang Q., Yokoi K., Kajita S., Kaneko K., Arai H., Koyachi N., Tanie K., "Planning Walking Patterns for a Biped Robot", IEEE TRA, vol. 17, no. 3, june 2001
6. Vukobratovic M., Borvac B., Surla D., Stokic D., Biped Locomotion. Dynamics, Stability, Control and Application, Springer-Verlag, Berlin, 1990
7. Nakamura Y., Hirukawa H., Yamane K., Kajita S., Yokoi K., Tanie K., Fujie M., Takanishi A., Fujiwara K., Kanehiro F., Suehiro T., Kita N., Kita Y., Hirai S., Nagashima F., Murase Y., Inaba1 M., and Inoue H., "V-HRP: Virtual Humanoid Robot Platform", University of Tokyo, Electrotechnical Laboratory, MITI, Mechanical Engineering Laboratory, MITI, Mechanical Engineering Research Laboratory, Hitachi Ltd., Waseda University, Fujitsu Laboratories Ltd., ", Proc. First IEEE-RAS International Conference on Humanoid Robots (HUMANOID2000), Sep. 2000, pp. 1–14

Development of the Light-Weight Human Size Humanoid Robot RH-0

L. Cabas, S. de Torre, M. Arbulu, and C. Balaguer

Robotics Lab., Department of Systems Engineering and Automation University Carlos III of Madrid c. Butarque 15, Leganés, 28911, Madrid – Spain
lcabas@ing.uc3m.es

Abstract. This paper describes the dynamic and mechanical development of RH-0, an autonomous humanoid robot designed by the Robotics Lab of the University Carlos III of Madrid.

The main objective of the project has been to obtain the robot that would walk like a human being. Using trajectories generated previously by the kinematic analysis, a dynamic design was made, that accepted as the main hypothesis the weight of a person that is 1.20 m tall. With this weight the requirements for each joint's torque were calculated, then, by means of mechanical analysis, the structure was designed and the dimensions of the motors were determined. It was a cyclic process which was made successively until obtaining the optimal values of torque to get an anthropometrical form of walking that corresponded to a human being of his height (1.35 m).

The dynamical and mechanical designs have been made with the help of CAD programs, selecting motors and specific systems of transmission. According to the aforementioned, DC motors and Harmonic Drive gear sets have been adopted for the actuator system, and, in some cases, transmissions by means of belt drives have been placed.

1 Introduction

One of the most exciting challenges than has faced the engineering community in the last decades was to obtain that a machine of a similar form, a humanoid robot, could do the same activities as a human being and in addition, walk like him. There are several reasons to construct a robot with these characteristics. Humanoid robots will work in a human atmosphere with greater effectiveness than any other type of robots, because the great majority of environments where it would interact reciprocally with a human is constructed taking into account the dimensions of the latter. If it is supposed that a machine should complete dangerous tasks or work in extreme conditions, in the ideal case its anthropometric measures must be as close as possible to the ones of their

prototype. Inclusively, there are professionals who adduce that for a human being to interact naturally with a machine, this one must look like him [1].

The investigation and development of humanoid robots nowadays are growing every day, countries leaders as the U.S.A. and Japan have more than 30 years experience of investigation in the field of humanoid robots, obtaining in their development astonishing complexity and efficiency. The advances showed in their times by the ASIMO [7] or the HRP-2 [2] are admirable, they are able to walk linearly, to turn, to raise stairs, to carry objects, besides that they also can interact not only with the surroundings but with human beings. On the other hand, and with a budget far below, we are behind the first humanoid robot developed in Spain, the RH-0. Like a starting point, among the great amount of activities that can be applied to a robot of these characteristics, we paid more attention to the service of disabled people, so that it has been designed with specific anthropometric measures adapted to the height that has, for example, a person in a wheelchair. This is a very specific limitation of the objective that we are searching for with this prototype. Anyway, as it was said above, the applications for which a robot of these characteristics can be destined nowadays are many and very different, as much in the industrial sector as in the one of services.

2 General Characteristics

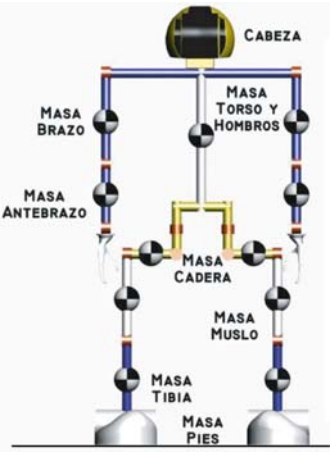
The design began with a very thorough study of the "State-of-the-art" of humanoid robots made until the moment [3], [6]. In this case, a special attention was paid not only to the form, number of degrees of freedom, their distribution, structural disposition of actuators and characteristics of each one, but fundamentally, what goal served every one of them, for example a robot destined to elevate loads has a very different engineering analysis from a robot which will only serve people.

As a result of this, it was stated that the primary objective of the prototype RH-0 was the accomplishment of a humanoid robot with a height that goes up to around 1.20 meters, able to walk in any direction, and with variation of heights, to manipulate light objects of up to 500 grams with the end of the arm and to recognize by means of the sensors located in its head the place towards which it has to go as well as voice instructions.

With these general parameters specified and trying to walk optimally and smoothly, without falling in redundancy, it was chosen to equip the robot with 21 degrees of freedom (DOF) distributed of the following way: in the lower part, 12 DOF were placed, 6 in each leg, of which 2 were in charge of the frontal turn of each ankle, other 2 were in charge of the saggital turn of those; in each knee 1 DOF was placed, responsible for the movement of the joint in the saggital plane and the other 6 DOF were in charge of the saggital, frontal and cross-sectional turns of the hip, on each side. In the upper part, we have 9 DOF: 1 DOF in the neck, in the cross-sectional plane responsible

for turning the shoulders and arms simultaneously; the arms, the left one as well as the right one, were distributed in the following way: in the shoulder, we had 2 DOF, one for the saggital plane and the other one for the frontal plane, another DOF was in the elbow, in the saggital plane and finally we have a DOF in the wrist, in the cross-sectional plane.

These DOFs and the workspaces corresponding to each joint give the robot an interesting form of walking, the angles of every one of them are presented in the following figure:



(a) Diagram of free body

Articulación		Eje	RH0
Cabeza		R	N.E.
		P	N.E.
		Y	N.E.
Brazo*	Hombro	R	-95º a 0º
		P	-180º a 60º
		Y	N.E.
	Codo	P	-90º a 0º
		Y	N.E.
	Muñeca	R	N.E.
		P	N.E.
		Y	-90º a 90º
	Mano	P	-16.5º a 60º
Cintura		R	N.E.
		P	N.E.
		Y	-45º a 45º
Pierna*	Cadera	R	-12º a 12º
		P	-80º a 80º
		Y	-15º a 15º
	Rodilla	P	0º a 150º
	Tobillo	R	-12º a 12º
		P	-100º a 100º

(b) Comparative diagram-Turning angles

Fig. 1. General characteristics

To the left, the diagram of free body of the prototype in predesign stage can be observed, where the distribution of the different degrees of freedom can be seen clearly

In addition to this, like an improvement that already is being implemented in the prototype is the system of recognition of objects, voice and artificial vision, that has been given to the robot by a camera located in the head, Fig. 2, that has an independent motor that will cause that, although the head is immovable, the system of recognition and vision will have a DOF more thus increasing its effectiveness.

In the Fig. 3 to the left the initial design can be observed during the stage in which specific things were considered like the location of the DOFs, general dimensions of various components and an approximation of the actuator and gear systems. In the central figure, we can see the prototype in CAD/CAM that has already been almost finalized, previous to its entrance to the factory, after adjusting to the maximum all the constructive and engineering parame-

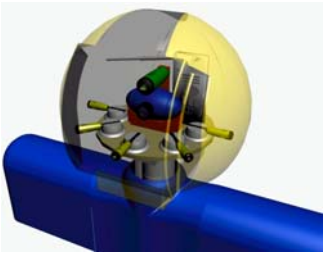
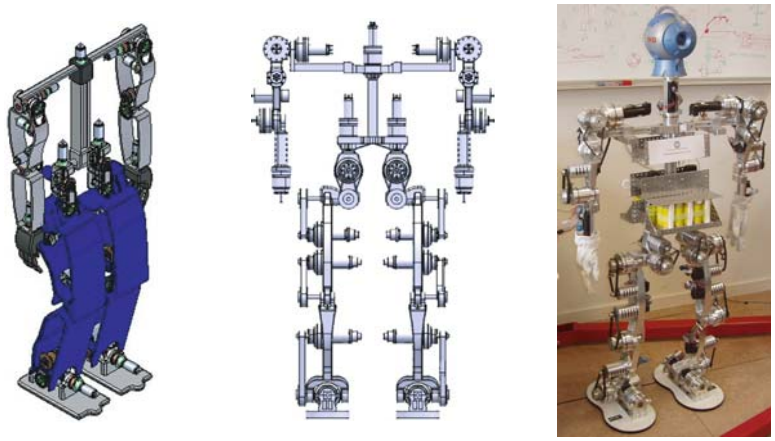


Fig. 2. Details of the head with sensors



(a) Initial design (b) Developmentin CAD/CAM (c) Final Prototype

Fig. 3. Development of the RH-0

ters that we considered. To the right, it is possible to see what is the first prototype of the RH-0, mounted:

Entering a little bit in what is called the structural design, we can emphasize two important points in the final configuration of the robot. One, it was the design of the hip, in form of double Cantilever, similar to the one used by the robot HRP-2 [2]. It was decided to choose for the six DOFs of the hip the disposition of a beam on a blacket or Cantilever due to the advantages from the point of view of the rigidity that it possesses as opposed to the configuration of a supported beam, used in the ASIMO, since in the last one the value that reaches the flexion moment can even produce a failure in the operating motors of the hip in the saggital plane and in the cross-sectional plane. In the second place, it is also necessary to emphasize the disposition obtained in the joints of the shoulder and the ankle, in which the axes of the frontal and saggital planes have been crossed in a point, by doing so it was possible to simplify the kinematics due to the absence of the height difference among

them, facilitating the control and thus optimizing the movement that allows a harmonic and flexible gait, the closest possible approximating to the human one.

3 Dynamic Analysis – Selection of Actuators

Since the RH-0 was going to be a project that would be built in reality, it was necessary to consider a great number of constructive mechanical aspects of the robot, for example, the limitations of the movement (see Table 1), loads to which the structure will be exposed, torques that must be able to support the motors in each one of the joints, etc.

According to the last one, we can say that although for the human beings the action to walk is something mechanical and practically intuitive, the detailed study of the way to walk and why we do it that way, is something complex and difficult to describe. By our part, by means of a complex kinematic analysis we studied the trajectories that our humanoid would have to follow, where a number of considerations were established in order to be able to get the optimal trajectory, without which the robot would lose the balance and fall down on the ground. For the same reason, the concept of the ZMP (Zero Moment Point) was taken into account at the time of designing the movement of the robot [4].

Dynamic Analysis

To define the movement of an artificial mechanism with two feet, it is always supposed to make some certain simplifications, some forced ones (for example, that a motor acts like several muscles) and others for practical reasons. With those, the dynamic calculation was reduced to the study of a simple inverted pendulum in whose end the total mass of the robot was concentrated. This study obviously was approximated, but the obtained results serve to have an orientative idea of the torques that are going to take place in the critical joints like for example the ankles and the knees of the biped. Therefore the necessity arises to make a dynamic study whose objective is the calculation of the torsion efforts that are going to take place in the joints, so that at the time of selecting the motors that are going to move the robot, these are the most optimal possible, preventing the unnecessary oversizing of the motors as well as their failure due to efforts that they are incapable to support. In order to obtain the dynamic model of the biped, it was chosen to use a number of functions available in the toolbox of robotics of Matlab, that was optimal for us and therefore presented totally trustworthy and precise results.

In the frame shown next, the general characteristics of the gait developed like torque, speed and weight, of the humanoid robot RH-0, can be seen.

Fig. 4. Final results

Selection of Actuators

Once collected all the necessary data for each joint we come to the selection of the motion system.

Starting with the motors, at the time of choosing there are two factors that overwhelm the rest: the first, the minimal possible weight; and the second, maximal ratio nominal final torque/weight. The others are important, but absolutely distinctive. Thus, within the great variety of motors presented in the market, it was choosed to equip the robot with the "DC-Micromotors Graphite Commutation" type. For practical reasons, all the robot, its 21 GDLs are equipped only with 3 different types of these motors:

Table 1. Used motors

Type 024	Commut.	Outer (mm)	Length (mm)	Shaft (mm)	Nom. Voltage (V)	No-Load Speed (RPM)	Stall Torque (mNm)	Out. Power (Watt)
Type 1	Graphite	26	42	4	24	6400	139	23.2
Type 2	Graphite	32	57	5	24	5900	547	84.5
Type 3	Graphite	38	63	6	24	6700	1290	226

It is obvious that these motors do not provide the necessary torque in each joint's axis, for that reason it was decided to equip each one of them with a gear set. This set is composed by a gearhead and in some cases by a belt transmission where it had to make the right choice given the space available and the joint's torque. After an exhaustive analysis considering several parameters and gear producing companies, it was decided to use for all the joints

the gear family of Harmonic Drive AG. In order to complete this analysis we can say that in spite of a presumable more competitive price, the use of other technologies in the design, like conventional planetary reducers or of the çycle technology absolutely does not compensate for their lower qualities.

Also it is necessary to be mentioned, that in all the joints that consist of belt transmission, an emphasis was made that this one was made via parallel axes. Given that the humanoid robot is a machine that has a mechanical movement with static loads, dynamic and repetitive cyclical changes, it turns out that to decide on another type of transmission (for example, by a pinion and a crown at 90° or a rack) can be very troublesome and therefore not very effective.

4 Structural Analysis

It is of great importance to know if the structure will be the able or not to support the weight of the great amount of mechanical, electrical and electronic components that the robot will have onboard. During what was the structural analysis, a measurement of all the structure was made with its later verification by means of the method of finite elements (FEM). Before we enter in it, it is helpful to mention that all the structure of the robot was made from the aeronautical aluminium 7075.

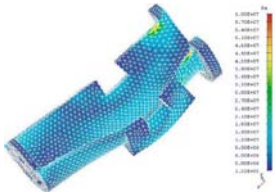
Then, during this verification, we will present the analysis of the key pieces that are in our understanding the most important within all the structure of the robot [5]. In the six analyzed pieces we have considered loads and moments proportional to the weight of the structure that they support, adding a safety factor 2, applying them to the points where the maximum efforts take place.

In the analysis of the ankle (in all the pieces a similar methodology was followed), loads equivalent to all the weight of the robot were applied in each contact surface with the supports of the axis, simulating what in fact was to hold all the weight of the structure during the simple support phase. The load has been considered 45 kilograms of weight and multiplied by a factor of safety 2 what gives 90 kilograms, that is to say about 900 Newtons that have been distributed in two halves on each side of the ankle, where the lodgings of the axes are located. In the case A, with "fillets", that is with radios of curvature (eliminating therefore the effect of stress concentration) and in the case B without them. Cases C and D correspond to flexo-torsion efforts:

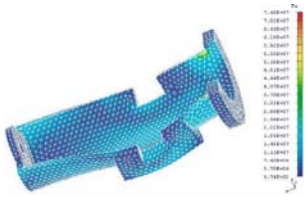
In the analysis of the tibia, loads in the contact with the thigh were applied, with angles of 10° in the case A and 30° in the case B, because they are the working angles, the initial and the final one for each case. Cases C and D represent the flexo-torsion. Here we also consider the weight of all the robot, multiplied by a safety factor.

For the analysis of the hip, we have divided it into upper and lower hip. In the analysis of the lower hip, one of the most critical pieces and of a difficult conception, in the case A we applied a load simulating a lateral effort,

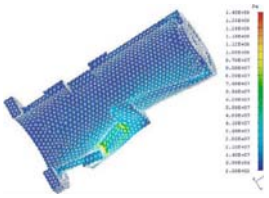
(a) Initial Design



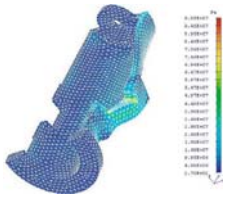
(b) Case A



(c) Case B



(d) Case C



(e) Case D

Fig. 5. Detail of the analysis in finite elements of the ankle

(a) Initial Design

(b) Case A

(c) Case B

(d) Case C

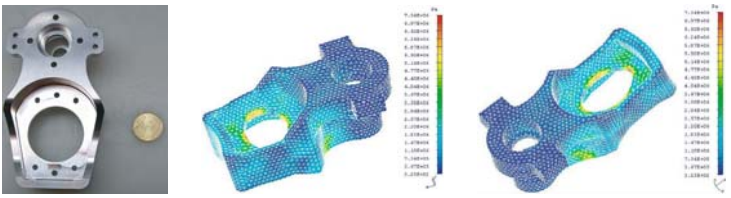
(e) Case D

Fig. 6. Detail of the analysis in finite elements of the tibia

and in the case B another one than represented the weight of the structure (considering a structure of a weight equal to that of all the robot). In the analysis of the upper hip, compression loads were applied in the case A, and traction loads in the case B. The case C represents the effect of loads applied in the union with the lower hip.

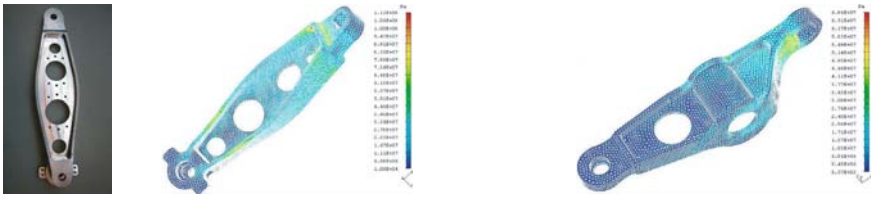
(a) Initial Design (b) Case A (c) Case B

Fig. 7. Detail of the analysis in finite elements of the lower hip



(a) Initial Design (b) Case A (c) Case B

Fig. 8. Detail of the analysis in finite elements of the superior hip



(a) Thigh (b) Case A (c) Analysis of the arm

Fig. 9. Details of the analysis in finite elements of the thigh and the arm

In the analysis of the upper hip, compression loads were applied in the case A, and traction ones in the case B. The case C represents the effect of loads applied in the union with the lower hip. In the analysis of the arm, we applied loads of 25 kg at its end simulating the weight of the arm. In the end, in the final analysis, we simulated torsion in the thigh.

Tall this analysis had like objective to verify that the structure was the sufficiently resistant to support the different types of movements that the robot would carry out.

5 Conclusions

As it was already said at the beginning of this article, this project is simply a prototype of what is intended to become a reality in the future that will be within the reach of everyone, able to help and to simplify the life of the people.

Concerning the things previously exposed, it is possible to say that we have obtained a very satisfactory result of the conceived robot, with high quality parts, and that means the starting point for innumerable investigations.

At the moment, to the subject of this article, still many improvements and corrections are to be done since it is in the process of development. Among the objectives that we have in mind are to complete the design of the previously mentioned clamps for the hands, the development of kinematics for the upper part of the robot in such a way that it would help the stability of the robot and the movements of the robot when walking would resemble ones of a human person and the possibility to incorporate remote control of the humanoid via voice or a PDA.

References

1. Gordon Wyeth, Damien Kee, Mark Wagstaff, Nathaniel Brewer, Jared Stir-zaker, Timothy Cartwright, Bartek Bebel. “Design of an autonomous Humanoid Robot”. School Computer science and electrical engenieering, University of Queensland, Australia.
2. Kenji Kaneko, Fumio Kanehiro, Shuuji Kajita, Kazuhiko Akachi, Toshikazu Kawasaki, Shigehiko Ota, Takakatsu Isozumi, “Design of Prototype Humanoid Robot Platform for HRP”. Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland – October 2002.
3. Kazuo Hirai, Masato Hirose, Yuji Haikawa, Toru Takenaka. “The development of Honda Humanoid Robot”. Proceedings of the 1998 IEEE. International Conference on Robotics and Automation, Leuven, Belgium – May 1998.
4. Ambarish Goswami. “Postural stability of biped robots and the foot-rotation indicator (FRI) point”. Department of Computer and information science. University of Pennsylvania.

5. Giusseppe Carbone, Hun-ok Lim, Atsuo Takanishi, Marco Ceccarelli. “stiffness analysis of the humanoid robot Wabian RIV: Modelling. Proceedings of the 2003 IEEE. International Conference on Robotics and Automation, Taipei, Taiwan – September 2003.
6. K. Hirai, “Current and Future Perspective of Honda Humanoid Robot”. Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp. 500–508, 1997.
7. M. Hirose, Y. Haikawa, T. Takenaka, and K, Hirai, “Development of Humanoid Robot ASIMO”, Proc. Int. Conference on Intelligent Robots and Systems, Workshop 2 (Oct. 29, 2001), 2001.

Human Machine Interface for Humanoid Robot RH-0

I. Prieto, C. Pérez, and C. Balaguer

Robotics Lab, Department of Systems Engineering and Automation, University Carlos III of Madrid, C/ Butarque 15, Leganes, 28911, Madrid-Spain
100026772@alumnos.uc3m.es
{cmperez,balaguer}@ing.uc3m.es

Abstract. The objective of this paper is to give an overview of the Human Machine Interface (HMI) for the humanoid robot RH-0 that is currently under development at the Robotics Lab of the University Carlos III of Madrid.

The first section lists the features that a HMI must include to run properly and be an efficient tool to manage a system. Then, the HMI developed in the initial phase of the Project is described. Based on this first HMI, a second one was designed to operate as an interface between the physical robot and the RH-0 users. A complete description, including its working modes, is explained in the third section.

There are three working modes that provide the movement of a joint, the load of an externally generated trajectory and, finally, a movement editor, which allows the design of a complex trajectory with regard to built-in patterns. Moreover, a test mode is available for all these options, which allows some movements that are usually forbidden because they involve unsteady positions, which may cause damage to the system. Other important features of this HMI are the modules of communication robot-workstation, which is integrated on the interface, and the security one, that checks all movements that will be executed by the humanoid. This last module is based on the concept of the ZMP[1], which is calculated using a mathematical model. Ongoing checks ensure that it is always on a stable position; otherwise, the movement will be cancelled, the robot will not received any order and the user will be warned of this error.

1 Introduction

The functionality of developed systems relies on having tools to interact with them and allows an easy management. A user-friendly and useful interface improves the efficiency of the system since the parameters related to its running can be displayed.

Regarding the interface – currently under development – for the humanoid robot RH-0, its design expects meeting two endpoints:

[1]ZMP: Zero Movement Point. See appendix for more details

1. Integrate all the works developed by the different teams within this Project, joining each of the contributing parts of the system.
2. Simulate and control all the parameters related to the joint trajectories under a user-friendly and easy to operate graphical environment.

1.1 HMI Importance

One critical aspect in the process of a system design, especially in robotics, is trying to get the higher adaptability between the machine and humans and vice versa. That is why the way in which information is exchanged must be carefully handled in order to make it as quick and simple as possible.

Whenever users judge a system, they mainly take into account its interface. That is the reason why the designing process is essential. A poor interface decreases the functionality and could drive the user to make catastrophic errors, being this situation critical in case of a robot. So the design of this component and the whole system cannot be independent. It is fundamental to include the users in the designing process in order that the application can meet the actual needs without an unnecessary increase of the inherent complexity of the system, allowing its manipulation without a deep knowledge of the system architecture.

1.2 HMI Features

Before the design of a graphic interface that lets an efficient communication between the machine and humans, one of the most important aspects that should be taken into account is how easy-to-use can be. This feature entails learning simplicity and intuitive manage. Besides, minimizing the user error rate is critical. Apart from the aforementioned characteristics, a high speed performance should lead users to an overall subjective satisfaction with the system.

The human-computer interaction (HCI) development consists of two steps:

- Study of the interaction, based on how the interface works and the feelings that generates on users.
- Interaction system programming, that includes graphical user interface (GUI), error detection and handling, alarm signals, etcetera.

The designer should best estimate the future usage of the system, the environment and the application to provide the most effective and appropriate solution.

The main features of a high-quality HMI coincide with its advantages:

- Easy-to-use.
- Easy and quick to learn, even for inexperienced users.
- Users may switch between different processes in a timely manner and interact with different running applications.
- The required information is immediately displayed and allows for a quick access and interaction.

2 Previous Works: Simulation HMI Description

It is well known that robotics is a science that combines many other technological fields. So the development of a humanoid robot needs a tremendous amount of efforts, knowledge and work. In spite of those different technological profiles, all tasks need to be aligned so that the maximum functionality of the system can be achieved after integration.

At the first development stage, two goals have been reached. Firstly, joining each tasks performed in the designing phase of RH-0 humanoid robot. Second1y, providing an easy-to-use interface to manipulate and display joints trajectory data. This data is generated by the kinematic and dynamic model study and it is essential. For example, in order to select the proper actuator size in the electro-mechanic design.

The main functionality of this first interface release is the simulation of joints trajectory and its parameters. Besides, this application displays the pseudo-ZMP throughout every movement, so the robot stability can be checked.

Since graphical environments in operating systems became commonly used, a GUI is often implemented in a window, and so this interface. This window allows direct manipulation and control of objects and icons to send commands and visualize system outputs. These actions are executed through a computer mouse.

Some advantages of this kind of implementation are:

- Continuous displays of relevant objects and actions.
- Physical events or pressing of labelled buttons instead of commands with complex syntax.
- Quick and reversible operations that affect relevant objects and can be immediately seen.

The main disadvantage lies in the manipulation of the computer mouse that strongly depends on the user skills.

Matlab® is the programming application used for the interface, as it integrates tools to obtain GUIs easily and efficiently and also, because the works carried out in the designing phase were modelled with it.

HMI can be divided into two parts:

- Integration of algorithms that resolve kinematic and dynamic issues.
- Simulation and visualization of movements.

The simulation and visualization matter have been based on the Denavit-Hatenberg robot model. An accurate structure display can be obtained through minimal modifications to this model. On the other hand, movement data of each joint, such as angular rotation, velocity, acceleration and torque, can be shown by just pressing a button. Temporal evolution and maximum and root-mean-square values of data are also available.

Because of all aforementioned, this HMI has been proved as an efficient tool for the initial designing stage of the RH-0 Project.

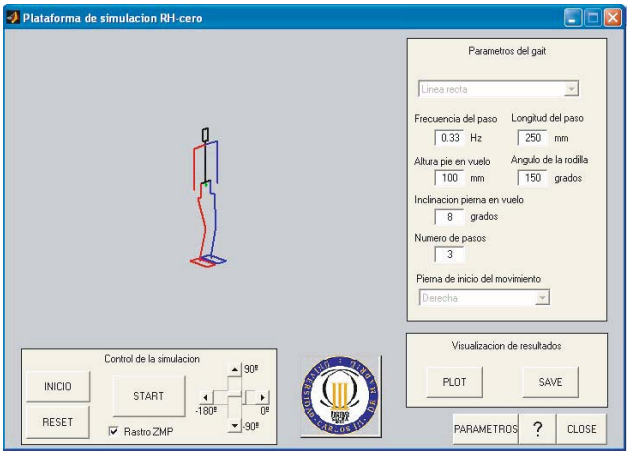


Fig. 1. HMI Simulation Phase

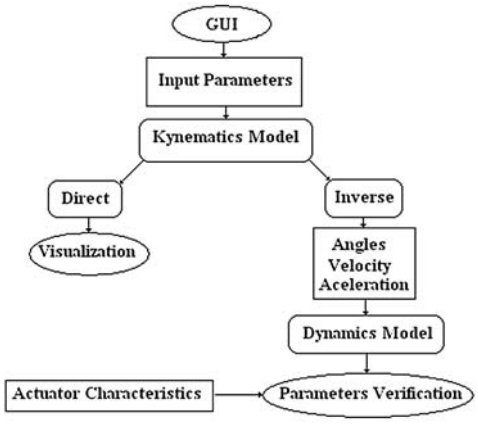


Fig. 2. Process Scheme

3 Physical System HMI

The HMI developed at the first designing stage is used as baseline for the interface that finally will be part of the real system. But, new functionalities must be implemented and integrated in a different environment.

The whole system is compounded of two distinguishable parts, the humanoid robot and a workstation. The HMI runs in the second together with a kinematic module and a communication manager. So, the workstation application is made up by these three modules.

The kinematic module generates each robot joint trajectory using a mathematical model previously developed.

The communication architecture is designed following the client/server model. According to this, the workstation communication manager acts as client and the robot software control acts as server in a wireless network. The goal of the communication manager module is to establish the information transfer – commands and data – between the HMI and the robot.

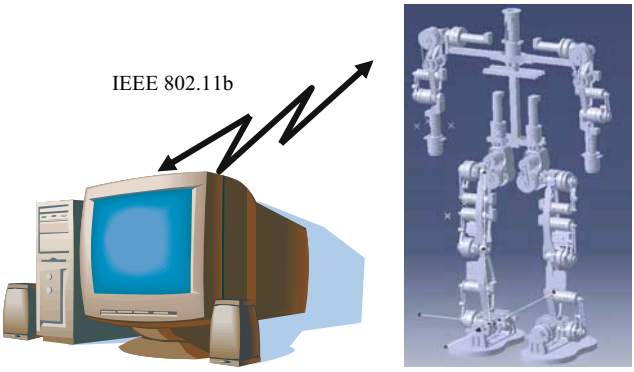


Fig. 3. System hardware distribution

Visual Basic.NET is the programming language used for the HMI, because it allows developing easy and powerful GUIs. Besides, this new generation of VB adds new functionalities for OOP[2] and multi-task. This feature is essential in this case, since the HMI and the communication manager must run at the same time, and this was not possible before with other VB versions due to limitations based on sequential execution.

The HMI has been planned for two kinds of users, system developers and potential final users. Both of them have different requirements from the system, so diverse operating modes are provided.

System developers are responsible for checking the execution of different trajectories and system performance in detail. Therefore, it is necessary to allow them to have total control over each joint and over the communications established between the workstation and the robot.

Final users expect the system to carry out all requested tasks. Hence, just a check register and security elements should be available for them.

Users can choose different operating modes, depending on their profile, and order a certain movement. Data from the model will go to the interface. Then, it will be turned into the communication protocol developed and sent to the robot via wireless, and finally, executed.

Not only users can ask for certain robot movements but also, they can obtain precise data related to its state parameters.

[2]Object Oriented Programming

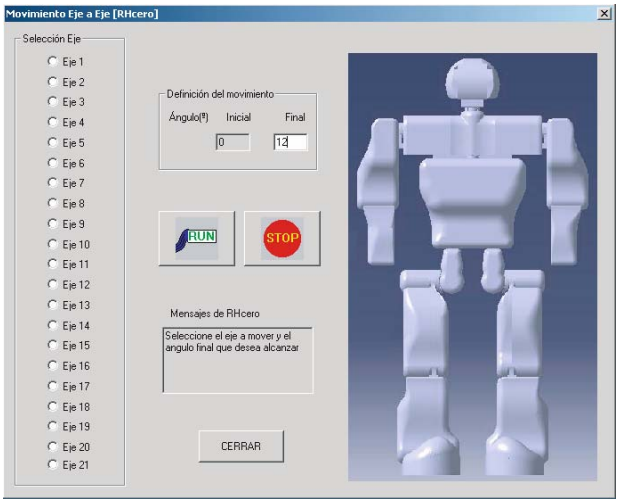


Fig. 4. Workstation HMI Screenshot

According to the three layers model, that was chosen in the software architecture [3], [4], HMI is placed in the upper layer, called 'Organization'. This layer comprises four components:

1. A module that generates joint trajectories, where some mathematical functions have been implemented. These functions provide the solution for the direct and indirect kinematic problems to obtain the desired joint trajectories.
2. A communication manager in charge of the information – data and commands – exchange with the robot software according to the protocol designed *ad-hoc* and through a wireless link IEEE 802.11b.
3. The graphic interface, GUI, where the user can obtain information and operate the robot.
4. A hub process that works as a command translator between both user and robot. This process requests services to the aforementioned components, adapting data and commands to the protocol and carrying out the Organization layer tasks.

There are two ways of selecting movements. On the one hand, a collection of complete movements are available as files. On the other hand, an editor can be used to design them based on primary trajectories within the HMI. Afterwards, they can be either saved to the file library or be executed by the humanoid robot.

An emergency stop can be launched during the robot execution. Besides, every message between the humanoid and the Workstation is recorded in a tracking file.

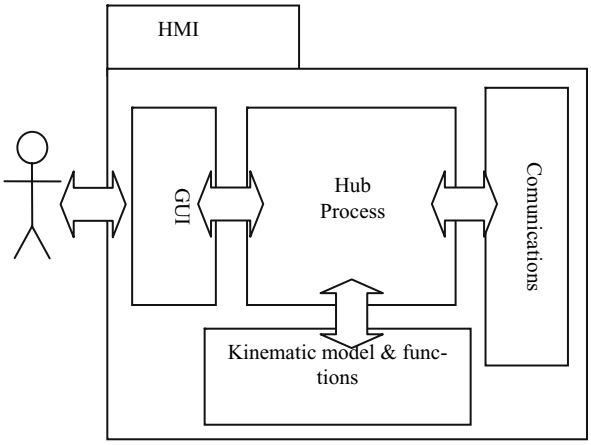


Fig. 5. HMI block diagram

Developers have access to another operating mode different from the ones above. They can control each joint and pause the system without using the emergency stop, so, the robot can continue with the former execution. Moreover, the message tracking file can be displayed for troubleshooting.

4 Future Work

In the future, new modes to manage the humanoid robot will be added to the application. Also, a simulation of the commands that the user send to the RH-0 will be displayed.

As aforementioned, the currently manage of the system is restricted to command simple movements, for example rotate a specific joint, take a step or run a predefined file. At the following designing phases, new and more complex movement functions will be added. Once the system, the mathematical model, joint trajectories, gait and so on, have been debugged, the user is not likely to require the basic modes that are now under development. This is supposed to be done automatically, as humans do.

More complex trajectories are also under development. These consider the robot RH-0 as a mobile entity, so that users only need to specify the point where they want the robot to go as a command input. The robot itself will calculate how many steps are needed and the length of the gait, so, as a result, the joint trajectories.

Since robot sensorization is the next stage of the Project, high-level features will be taken into account.

Finally, a virtual display will be included in the GUI. The HMI will simulate the system performance during the actual robot execution. In this way,

users and development engineers can check both the physical movement and the reactions of the robot software depending on the commands.

5 Conclusion

A graphic interface design for a system is not an easy task that can be independently developed from the whole system. Many considerations regarding the potential users must be taken up-front. It must provide all the required system functionalities and data; allow debugging the design of the different components of the monitorized system and the interface itself. A high-quality HMI is a very valuable tool for the system development and management.

An HMI, easy-to-use and intuitive, as well as efficient and powerful, is being designed at the Robotics Lab in the Carlos III University. It will be used by the project's developers to troubleshooting and manage the humanoid robot RH-0.

Specific tasks can be done with this interface, such as moving a certain joint or the entire robot in a room just by giving a direction command.

Besides, system performance can be analyzed by displaying the state parameters and comparing the real movement of the robot with a simulation at the same time.

Finally, it is expected that this HMI will become an example, even a model, for future projects of humanoid robotics.

References

1. Katzel J (2003) Tools for HMI Applications. Control Engineering December 1, 2003. http://www.manufacturing.net/ctl/article/CA339682?industry=Information+Control&industryid=22071&spacedesc=communityFeatures
2. Diseño de una interfaz grafica. http://www.uag.mx/66/int1.htm
3. Brooks RA (1986) A Robust Layered Control System for a Mobile Robot. In: IEEE Journal of Robotics and Automation, Vol. 2, No. 1, pp. 14–23
4. Shi Zongying, Xu Wenli, Wen Xu, Jiang Peigang (2002) Distributed Hierarchical Control System of Humanoid Robot THBIP-1. In: Proceedings of the 4th World Congress on Intelligent Control and Automation, June 10–14, 2002, Shanghai, P.R.China

Appendix

Pseudo-ZMP Tracking as Movement Stability Criteria

All trajectories are checked before being sent to the robot control software for execution. This is done to test the position of the pseudo-ZMP inside a stable area during the desired movement.

At some stage in biped locomotion, when a foot leans on the floor, a pressure distribution appears in the sole. It can be condensed into a single force applied to a certain point. Because the momentum addition of all forces actuating over the humanoid robot respect to this point is zero, it is call Zero Movement Point (ZMP). In the static robot model, a simplification is used: make the robot gravity centre point projection equal to the sum of all floor reacting forces.

The ZMP calculation requires knowing the weigh of each humanoid piece, its gravity centre point and its displacement throughout the movement. This gives an accurate estimation of the condensed gravity centre. A projection of this point allows checking if it is always in a stable position, such as the sole – during a step or a simple support – or the area defined between the two feet – during a static stand or double support–. After all trajectories evaluation and stability checks, the humanoid robot can perform the movement in a secure manner.

Trajectory Planning for the Walking Biped "Lucy"

Jimmy Vermeulen, Dirk Lefeber, Björn Verrelst, and Bram Vanderborght

Vrije Universiteit Brussel, Pleinlaan 2, Brussels, Belgium
Jimmy.Vermeulen@vub.ac.be
http://lucy.vub.ac.be/trajectory.html

Summary. A real-time joint trajectory generation strategy for the dynamic walking biped "Lucy" [1, 2] is proposed. This trajectory planner generates dynamically stable motion patterns by using a set of objective locomotion parameters as its input, and by tuning and exploiting the natural upper body dynamics. The latter can be determined and manipulated by using the angular momentum equation. Basically, trajectories for hip and swing foot motion are generated, which guarantee that the objective locomotion parameters attain certain prescribed values. Additionally, the hip trajectories are slightly modified such that the upper body motion is steered naturally, meaning that it requires practically no actuation. This has the advantage that the upper body actuation hardly influences the position of the ZMP. The effectiveness of the developed strategy is proven by simulation results.

1 Introduction

The upper body of a biped or a humanoid robot usually contains the on-board hardware of the control system, as well as the batteries in case of electric actuation, or a pressurized vessel or a compressor in case of pneumatic actuation. Therefore the upper body is generally the robot link with the largest mass and inertia. As a consequence, keeping this body upright requires significant action of the actuators, which might cause problems concerning the position of the Zero Moment Point [3]. In this paper, a planning strategy is developed that uses the angular momentum equation to estimate the *natural* dynamics of the upper body, or in other words, the motion of the upper body when it is unactuated. The trajectories for the leg links are then established such that this natural behaviour of the upper body approximates a given prescribed behaviour. When this desired behaviour, which will be modelled by a polynomial function, is used as a reference trajectory for the upper body actuator, then the work of this actuator is limited to overcoming the minor differences between the natural and the polymomial trajectory. The advantage of this strategy is that the resulting motion of the ZMP stays well within the boundaries of the stability region during the whole locomotion process. During the

single support phase the ZMP remains in the near vicinity of the ankle joint of the supporting foot. During the double support phase the ZMP travels from the rear ankle to the front ankle joint.

A planar walking biped "Lucy" has been developed by the Multibody Mechanics Research Group at the Vrije Universiteit Brussel [1]. The trajectory planner described in this paper is tested by simulation on the model of Lucy. Since its effectiveness has been verified, this strategy will be implemented on the real robot in the near future. The simulated motion of Lucy contains single as well as double support phases, while also taking the impact phase into account. The strategy developed consists of a limited number of elementary computations, which makes it useful as a real-time trajectory planner for the biped Lucy.

2 Double Support Phase

In Fig. 1 the model of the planar biped Lucy is depicted during a double support phase. The R stands for Rear foot, whereas the F stands for Front foot. Since both feet are in contact with the ground, a closed kinematic chain is formed by the two legs and the ground. It is desired that both feet stay in contact with the ground and that the feet do not slip during a double support phase. The horizontal and vertical distances between the two ankle joints are respectively called step length and step height and are given by:

$$X_{F_F} - X_{F_R} = \lambda \quad \text{(step length)} \tag{1}$$

$$Y_{F_F} - Y_{F_R} = \delta \quad \text{(step heigth)} \tag{2}$$

Due to the two holonomic constraints, the robot's number of DOF is equal to three.

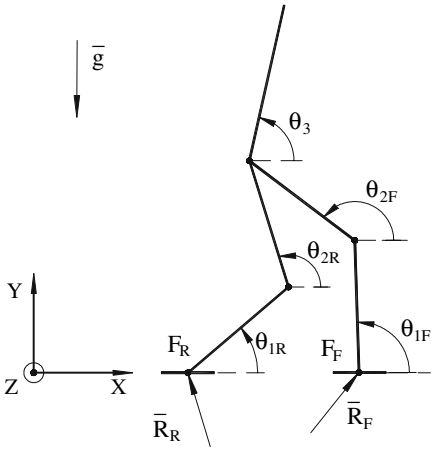


Fig. 1. Lucy during DSP

2.1 Hip Motion During DSP

Suppose that the following Lagrange coordinates are chosen to describe the motion:

$$q_D = (X_H, Y_H, \theta_3) \tag{3}$$

where X_H and Y_H respectively represent the horizontal and vertical position of the hip joint. In order to obtain smooth joint trajectories for the leg links, fifth order polynomial functions are established to be tracked during the double support phase. Suppose initially that only the two knee actuators are used. The polynomial functions have to connect the following initial and final boundary values for the hip motion:

$$\begin{array}{rrrcrrr} X_H(t_+), & \dot{X}_H(t_+), & \ddot{X}_H(t_+) & \rightarrow & X_H(t_D), & \dot{X}_H(t_D), & \ddot{X}_H(t_D) \\ Y_H(t_+), & \dot{Y}_H(t_+), & \ddot{Y}_H(t_+) & \rightarrow & Y_H(t_D), & \dot{Y}_H(t_D), & \ddot{Y}_H(t_D) \end{array}$$

where t_+ indicates the time instance immediately after the impact phase, and t_D represents the end of the double support phase. The duration of the double support phase is then equal to $T_D = t_D - t_+$. Note that if the robot starts the double support phase from rest, that $t_+ = 0$ can be chosen, combined with $\dot{X}_H(0) = 0$, $\ddot{X}_H(0) = 0$, $\dot{Y}_H(0) = 0$ and $\ddot{Y}_H(0) = 0$.

2.2 Upper Body Motion During DSP

In order to derive the natural motion of the upper body during the double support phase, it is assumed that no actuator torque is acting on it. In that case, the upper body behaves as an inverted pendulum with a moving supporting point, being the hip point H. Considering the free body diagram of the upper body in Fig. 2, and applying the angular momentum theorem with respect to the hip point H, yields:

$$\dot{\bar{\mu}}_H = \overline{HG}_3 \times m_3 \bar{g} + m_3 \left(\bar{v}_{G_3} \times \bar{v}_H\right) \tag{4}$$

It can be shown that under the assumption of small rotations of the pendulum, equation (4) results in the following differential equation [4]:

$$\ddot{\theta}_3 = C \left[\ddot{X}_H - \left(\ddot{Y}_H + g\right)\left(\frac{\pi}{2} - \theta_3\right)\right] \quad \text{with} \quad C = \frac{m_3 \gamma l_3}{I_3 + \gamma^2 l_3^2 m_3} \tag{5}$$

and g representing gravity acceleration and I_3 being the moment of inertia of the upper body with respect to its COG.

When $X_H(t)$ and $Y_H(t)$ are given functions of time, and when initial conditions $\dot{\theta}_3(t_+)$ and $\theta_3(t_+)$ are specified, this equation can be numerically integrated to obtain the natural upper body motion during the double support phase. However, the goal here is not to determine a natural motion exactly, but to develop a trajectory for the upper body angle which corresponds to a

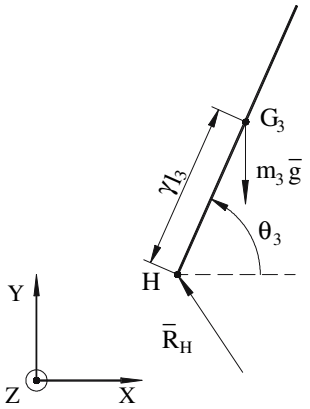


Fig. 2. FBD of upper body

motion close to a natural motion. A rough approximation of the natural upper body motion can be found by considering only the horizontal hip motion (by assuming that $\theta_3 \approx \frac{\pi}{2}$):

$$\ddot{\theta}_3^{nat} = C\ddot{X}_H \tag{6}$$

By twice integrating (6) over time, one obtains:

$$\begin{aligned}\theta_3^{nat}(t) = \theta_3(t_+) + (t-t_+)\dot{\theta}_3(t_+) \\ + C\left[X_H(t) - X_H(t_+) - (t-t_+)\dot{X}_H(t_+)\right]\end{aligned} \tag{7}$$

Suppose now that the duration of the double support phase is given the following value:

$$T_D = \frac{X_H(t_D) - X_H(t_+)}{\dot{X}_H(t_+)} \tag{8}$$

then roughly speaking the upper body angle at the end of the double support phase is equal to:

$$\theta_3^{nat}(t_D) = \theta_3(t_+) + T_D\dot{\theta}_3(t_+) \tag{9}$$

A fifth order polynomial function will be established for the upper body angle, connecting the following initial and final boundary values:

$$\theta_3(t_+), \quad \dot{\theta}_3(t_+), \quad \ddot{\theta}_3(t_+) \quad \rightarrow \quad \theta_3(t_D), \quad \dot{\theta}_3(t_D), \quad \ddot{\theta}_3(t_D)$$

with

$$\theta_3(t_D) = \theta_3^{nat}(t_D) \tag{10}$$

$$\ddot{\theta}_3(t_D) = C\left[\ddot{X}_H(t_D) - \left(\ddot{Y}_H(t_D) + g\right)\left(\frac{\pi}{2} - \theta_3^{nat}(t_D)\right)\right] \tag{11}$$

A suitable value for the angular velocity $\dot{\theta}_3(t_D)$ will be calculated in section 3.2, where calculations for the single support phase are performed.

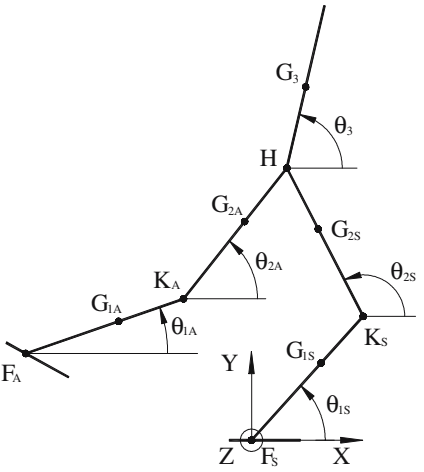


Fig. 3. Lucy during single support phase

3 Single Support Phase

In Fig. 3 the biped Lucy is depicted during a single support phase. The S stands for supporting leg, whereas the A stands for swing (air) leg. Since it is assumed that the supporting foot stays in contact with the ground and does not slip during a single support phase, the number of DOF is equal to five.

3.1 Hip and Swing Foot Motion During SSP

Suppose that the following Lagrange coordinates are chosen to describe the motion:

$$q_S = (X_H, Y_H, X_{F_A}, Y_{F_A}, \theta_3) \tag{12}$$

Assuming initially that no external ankle torque is exerted, so that only the knee and hip actuators are used, the robot is an underactuated mechanism. Two fifth order polynomial functions for the leg links of the supporting leg are established, which connect the following initial and final boundary values for the hip motion:

$$\begin{array}{ccccccc} X_H(t_D), & \dot{X}_H(t_D), & \ddot{X}_H(t_D) & \rightarrow & X_H(t_S), & \dot{X}_H(t_S), & \ddot{X}_H(t_S) \\ Y_H(t_D), & \dot{Y}_H(t_D), & \ddot{Y}_H(t_D) & \rightarrow & Y_H(t_S), & \dot{Y}_H(t_S), & \ddot{Y}_H(t_S) \end{array}$$

where t_S represents the end of the single support phase. The duration of the single support phase is then equal to $T_S = t_S - t_D$. This single support phase duration is defined as:

$$T_S = \frac{X_H\,(t_S) - X_H\,(t_D)}{\nu} \tag{13}$$

where ν is an objective locomotion parameter defining the mean horizontal hip velocity during a single support phase.

Two sixth order polynomial functions for the leg links of the swing leg are established, which connect the following initial, intermediate and final boundary values for the swing foot motion:

$$X_{F_A}(t_D), \dot{X}_{F_A}(t_D), \ddot{X}_{F_A}(t_D) \rightarrow X_{F_A}(t_i) \rightarrow X_{F_A}(t_S), \dot{X}_{F_A}(t_S), \ddot{X}_{F_A}(t_S)$$
$$Y_{F_A}(t_D), \dot{Y}_{F_A}(t_D), \ddot{Y}_{F_A}(t_D) \rightarrow Y_{F_A}(t_i) \rightarrow Y_{F_A}(t_S), \dot{Y}_{F_A}(t_S), \ddot{Y}_{F_A}(t_S)$$

The intermediate condition at $t = t_i$ is used to lift the foot whenever an obstacle has to be cleared during the swing phase. Note also that in all cases

$$\dot{X}_{F_A}(t_D) = \ddot{X}_{F_A}(t_D) = 0 = \dot{Y}_{F_A}(t_D) = \ddot{Y}_{F_A}(t_D)$$

since during the double support phase the feet remain fixed to the ground.

3.2 Upper Body Motion During SSP

In order to obtain the natural upper body motion, it is initially assumed that the ankle actuator is not used. In that case one can write the angular momentum equation with respect to the ankle joint of the supporting foot as follows:

$$\dot{\mu}_{F_S} = -Mg\,(X_G - X_{F_S}) \tag{14}$$

where X_G is the horizontal position of the global COG.

The kinematical expression of the angular momentum can be written as:

$$\mu_{F_S} = A_3\dot{\theta}_3 + h \tag{15}$$

with the function h being independent of the angular velocity of the upper body $\dot{\theta}_3$. The complete expressions for the functions h and A_3 can be found in [4] (see http://lucy.vub.ac.be/trajectory.html).

Upper Body Angle

Integrating (14) from $u = t_D$ to $u = t$, gives:

$$\mu_{F_S}(t) - \mu_{F_S}(t_D) = -Mg \int_{t_D}^{t} X_G\,du \tag{16}$$

A second integration from $t = t_D$ to $t = t_S$ yields:

$$\int_{t_D}^{t_S} \mu_{F_S}(t)\,dt - \mu_{F_S}(t_D)T_S = -Mg \int_{t_D}^{t_S} (T_S - t)\,X_G\,dt \tag{17}$$

Now introducing (15) into the lhs of (17) and solving for $\dot{\theta}_3(t_D)$ gives:

$$\dot{\theta}_3(t_D) = F + \frac{1}{T_S A_3(t_D)} \int_{t_D}^{t_S} A_3 \dot{\theta}_3 \, dt \tag{18}$$

with

$$F = \frac{1}{A_3(t_D)} \left[-Mg \int_{t_D}^{t_S} (T_S - t) X_G \, dt + h(t_D) T_S - \int_{t_D}^{t_S} h \, dt \right] \tag{19}$$

It can be shown [4] that when assuming small rotations of the upper body in the neighborhood of $\frac{\pi}{2}$, as well as small vertical motions of the hip joint, the function A_3 can be approximated as a constant:

$$A_3(t) \approx I_3 + m_3 \gamma^2 l_3^2 + m_3 \gamma l_3 Y_H(t) \approx A_3(t_D) \quad (t_D \leq t \leq t_S) \tag{20}$$

Expression (18) then becomes:

$$\dot{\theta}_3(t_D) = F + \frac{\theta_3(t_S) - \theta_3(t_D)}{T_S} = F + \frac{\Delta\theta_3^S}{T_S} \tag{21}$$

Now recalling expression (9), which calculates the upper body rotation during the double support phase:

$$\Delta\theta_3^D = T_D \dot{\theta}_3(t_+) \tag{22}$$

and demanding that $\Delta\theta_3^D + \Delta\theta_3^S = 0$ allows one to determine a necessary value for $\dot{\theta}_3(t_D)$:

$$\dot{\theta}_3(t_D) = F - \frac{T_D}{T_S} \dot{\theta}_3(t_+) \tag{23}$$

If this specific value for the angular velocity of the upper body is used for the construction of the polynomial function for the upper body during the preceding double support phase (see Sect. 2.2), then the upper body rotation will be compensated during the next single support phase, without the use of an ankle actuator.

Upper Body Angular Velocity

Evaluating (16) at $t = t_S$ and introducing the kinematical expression (15) yields:

$$\dot{\theta}_3(t_S) = \dot{\theta}_3(t_D) + \frac{1}{A_3(t_D)} \left[h(t_D) - h(t_S) - Mg \int_{t_D}^{t_S} X_G \, dt \right] \tag{24}$$

By varying the integral on the rhs of equation (24), different values for $\dot{\theta}_3(t_S)$ can be obtained. In practice a specific angular velocity $\dot{\theta}_3(t_S)$ can e.g. be attained by iteratively shifting the horizontal position of the hip point $X_H(t_D)$ at the end of the double support phase.

Upper Body Angular Acceleration

Evaluating (14) at $t = t_D$ and introducing the kinematical expression (15), gives:

$$A_3(t_D)\ddot{\theta}_3(t_D) + \dot{A}_3(t_D)\dot{\theta}_3(t_D) + \dot{h}(t_D) = -Mg\big[X_G(t_D) - X_{F_S}(t_D)\big] \quad (25)$$

Note that this equation corresponds to a zero ankle torque, or in other words to a ZMP located exactly at the ankle joint. Since $\ddot{\theta}_3(t_D)$ is imposed by the polynomial function during the double support phase, introducing expression (11) in (25) yields a condition which has to be satisfied at the beginning of the single support phase. Satisfying this equation results in a transition from double to single support phase with a ZMP coinciding with the ankle joint. In practice this can e.g. be achieved by tuning the hip accelerations $\ddot{X}_H(t_D)$ and $\ddot{Y}_H(t_D)$. For example, if one chooses a specific value for $\ddot{X}_H(t_D)$ (< 0), then (25) combined with (11) can be solved for $\ddot{Y}_H(t_D)$.

An analogous reasoning can be done at the end of the single support phase $t = t_S$, yielding a condition on $\ddot{X}_H(t_S)$ and $\ddot{Y}_H(t_S)$. This condition has to be satisfied in order to have the ZMP located at the ankle joint of the supporting foot before the impact occurs.

Tracking Function

In the preceding paragraphs, the natural upper body motion was manipulated such that the upper body is steered without requiring an ankle actuator, which corresponds to a ZMP located at the ankle joint during the single support phase. In order to compensate for modelling errors and possible external disturbances, a fifth order polynomial function will be established. This polynomial function is constructed with the boundary values from a natural motion, meaning that it is a reasonable approximation of the natural motion. It connects the following boundary values:

$$\theta_3(t_D), \quad \dot{\theta}_3(t_D), \quad \ddot{\theta}_3(t_D) \quad \rightarrow \quad \theta_3(t_S), \quad \dot{\theta}_3(t_S), \quad \ddot{\theta}_3(t_S)$$

Consequently, the torque exerted by the ankle actuator is low, meaning that the ZMP remains in the near vicinity of the ankle joint, which is the most dynamically stable position. Note also that due to the developed strategy during the double support phase, the ZMP automatically transfers from the rear ankle to the front ankle, without requiring external torques. Indeed, polynomial trajectories are constructed, connecting two successive single support phases, each with the ZMP located exactly at the ankle joint of its supporting foot.

4 Simulation

To test the effectiveness of the trajectory planner, a number of different simulations were performed [4]. The results of one steady walking pattern where

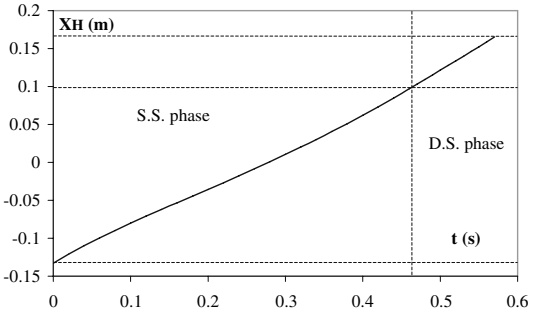


Fig. 4. Horizontal hip trajectory

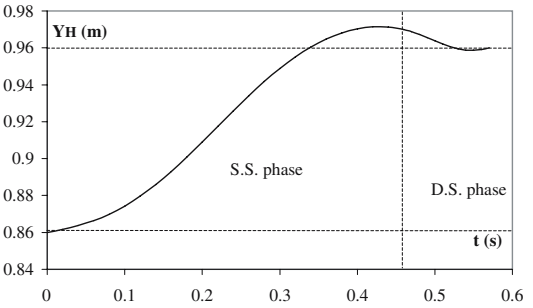


Fig. 5. Vertical hip trajectory

Lucy walks on a stairway are briefly reported here. The following objective parameters form the input for the trajectory planner:

- mean horizontal hip velocity $\nu = 0.5\ \frac{\mathrm{m}}{\mathrm{s}}$
- step length $\lambda = 0.3\,\mathrm{m}$
- step height $\delta = 0.1\,\mathrm{m}$

The total step duration is approximately 0.58 s, with a single support duration $T_S = 0.46\,\mathrm{s}$, and a double support duration $T_D = 0.12\,\mathrm{s}$.

Figures 4 and 5 show respectively the horizontal and vertical hip position during 1 full step, whereas Figs. 6 and 7 show respectively the horizontal and vertical swing foot trajectory during a single support phase. For this simulation, expression (23) yields $\dot{\theta}_3\,(t_D) = -0.19\ \frac{rad}{\mathrm{s}}$, which results in $\Delta\theta_3^S = -\Delta\theta_3^D = 0.022\ rad$. In expression (24) $\dot{\theta}_3\,(t_S) = \dot{\theta}_3\,(t_D)$ was chosen, which results in $X_H(t_D) - X_{F_S}(t_D) = -0.133\,\mathrm{m}$ at the beginning of the single support phase.

In expression (25), the value $\ddot{Y}_H(t_D) = 0$ was chosen, yielding $\ddot{X}_H(t_D) = -2.4\ \frac{\mathrm{m}}{\mathrm{s}^2}$. These accelerations guarantee that the ZMP is located exactly at the ankle joint of the supporting foot. An analogue calculation is performed at the end of the single support phase. Choosing $\ddot{Y}_H(t_S) = -3.5\ \frac{\mathrm{m}}{\mathrm{s}^2}$, leads

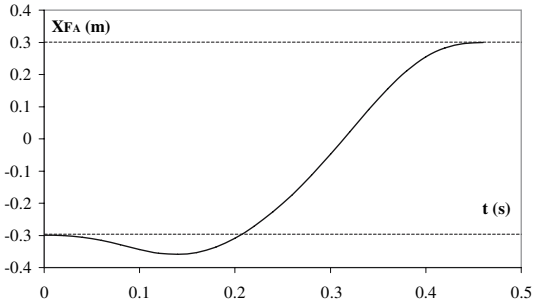


Fig. 6. Horizontal swing foot trajectory

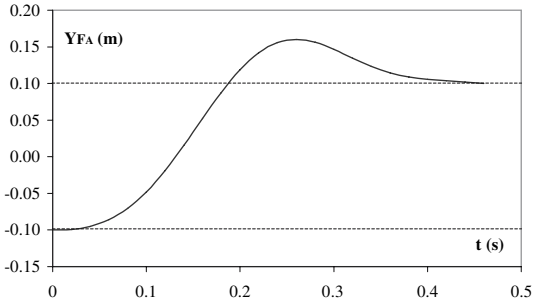


Fig. 7. Vertical swing foot trajectory

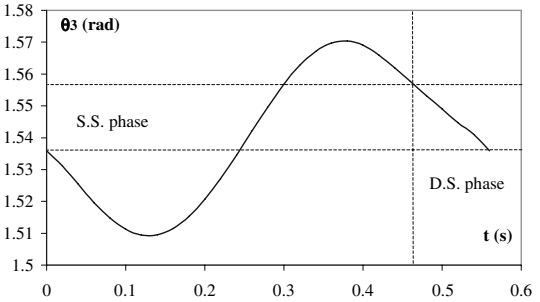


Fig. 8. Upper body angle

to $\ddot{X}_H(t_S) = 1.4\ \frac{\mathrm{m}}{\mathrm{s}^2}$. The polynomial trajectory for the upper body angle which approximates a natural trajectory, is shown in Fig. 8. It can be verified that the overall upper body oscillation is very small, although practically no actuation is required to achieve this. The corresponding trajectory of the ZMP is given in Fig. 9. Indeed during the single support phase the ZMP is located in the near vicinity of the ankle joint. During the double support phase the ZMP travels from the rear ankle to the front ankle joint. This motion of the

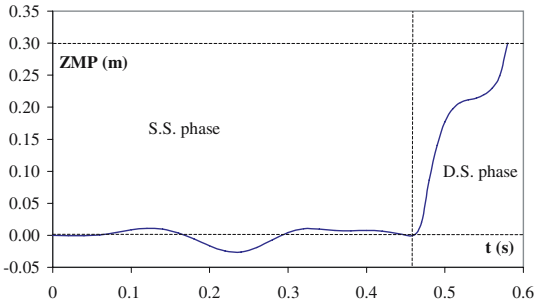


Fig. 9. Horizontal position of ZMP

ZMP is caused by the trajectory planner, which generates trajectories for the leg links by a priori taking the ZMP position into account. This is in fact an open-loop ZMP control strategy.

5 Conclusions

A trajectory planner for the planar walking biped Lucy has been developed. A set of objective locomotion parameters form the input for this planner, while the output are polynomial tracking functions for each of the robot links. The strategy tunes the natural upper body dynamics by manipulating the angular momentum equation, so that practically no actuation is required to keep the upper body upright. As a result, external ankle torques are extremely low, which means that the upper body actuation does not cause the ZMP to move out of the stability region. The leg link angle trajectories are established in such a way that the ZMP stays in the near vicinity of the ankle joint during the single support phase, and travels from the rear to the front ankle joint during the double support phase. Since the ZMP position is taken into account when establishing the tracking functions, the trajectory planner is in fact an open loop ZMP controller. The method is very effective, as can be seen by the simulation results. Since only a limited number of computations are performed, the strategy can be used in real time.

References

1. B. Verrelst, R. Van Ham, F. Daerden and D. Lefeber (2002) Design of a Biped Actuated by Pleated Artificial Muscles. In: Proceedings of the 5th International Conference on Climbing and Walking Robots (CLAWAR 2002), pp. 211–218
2. http://lucy.vub.ac.be

3. M. Vukobratovic and B. Borovac (2004) Zero-Moment Point – Thirty Five Years of its Life. International Journal of Humanoid Robotics (IJHR) 1(1):157–173
4. J. Vermeulen (2004) Trajectory Generation for Planar Hopping and Walking Robots: An Objective Parameter and Angular Momentum approach. PHD Thesis, Vrije Universiteit Brussel, Belgium

Height Control of a Resonance Hopping Robot

Roemi Fernández, Teodor Akinfiev, and Manuel Armada

Industrial Automation Institute, Spanish Council for Scientific Research,
Automatic Control Department. La Poveda 28500 Arganda del Rey, Madrid, Spain
roemi@iai.csic.es

Summary. This paper presents a model-based height controller for a one-legged resonance hopping robot. The particular construction of the robot with compensation of energy losses during robot's flight, and the use of a special dual drive with changeable transmission ratio for allowing an additional decrease of energy expenses, impose some specific requirements on the height controller. Equations of motion in vertical dimension are derived and solved to produce the actuator command that allows the robot to regulate its apex height. The reference control signals are obtained using minimum time and minimum energy as optimization criteria. Some experiments, carried out with a prototype of the resonance hopping robot, are also reported to prove the effectiveness of the proposed controller.

1 Introduction

Generally, the height of a hop is controlled by measuring and manipulating the system's energy. In this way, most of the height controllers for hopping robots are based on the height control loop of Raibert's three-part control system [1]. Raibert controlled hopping height by delivering a specified thrust value during the stance. The hopper would repeatedly reach a specified apex height, where the energy introduced during the stance would equal the energy lost due to friction and air resistance. Nevertheless, for such kind of control, it is necessary to use an electromotor enough powerful to accomplish the losses compensation by the end of the stance. Bearing in mind that the stance phase is between five and ten times less than a complete cycle time, it seems to be advantageous to make the compensation of losses during the flight phase instead of during the stance [2–4]. The height controller described in this paper has been designed taken into account this last consideration. Since this procedure reduces the required average motor power, it is possible to use a lower power motor, that results in a significant decrease of robot's weight. Other requirements for the height controller have been drawn from the particular design of the resonance robot. A detailed description of the resonance hopping robot and a first height control approach were presented in [5]. The robot has a body in which a leg

with a mechanical stop block is anchored to be able to accomplish forward movements. Four extension springs are installed between the body and the leg of the robot. A motor with a reduction gear is connected to the control system and is fixed on the robot's body. A cam mechanism is fitted to the shaft of the motor-reducer, and interacts with the robot's leg by means of a bearing fixed on it. The working part of the cam has a special shape to provide dual properties that enable to increase the motor efficiency adjusting the transmission ratio for two different movements: the first one, during the process of stretching of the springs, when the motor should turn the cam a certain angle relatively slow and in presence of external load; and the second one, during the process of releasing the springs, when the motor should turn the cam the same previous angle in reverse direction, several times faster and without external load. With the first height control approach presented in [5], the robot was only able to jump with the maximum possible apex height, but was not able to regulate the apex height. The height control loop proposed in this paper functions as follows. At lift-off the controller solves equations of motion in order to calculate the proper springs' extension for achieving the goal apex height in next hop. This extension is performed during the flight phase and is maintained until touch-down, at which point the energy stored in springs is released. At lift-off the cycle begins again. Since the stance period could be of very short during, a desired control objective could be to release the stored energy in a minimal time, using all the possibilities that the electromotor and the transmission have available. Matsuoka [6] analyzed hopping in humans with a one-legged model and derived a time-optimal state feedback controller that stabilized his system, assuming that the leg could be massless. Nevertheless, for the flight phase, the minimum time is not the best choice. If the movement is performed too fast, the motor will have to hold the springs' extension a longer time, causing a bigger energy consumption. Therefore, in this situation, it is more convenient to use a minimum energy optimization criterion, employing most of the flight time.

To add stability, instead of using directly time optimal and minimum energy controllers, a combination of time optimal control, minimum energy control, and backstepping is proposed, in such a way that, the optimal state trajectories are used as reference values in the controller that is designed using the integrator backstepping [7–9]. Thus, the approaches will be nearly optimal rather than exactly optimal.

The rest of the paper is organized as follow. In Sect. 2 the mathematical model of the resonance hopping robot is described. In Sect. 3 the time optimal control problem and the minimum energy problem for the calculation of the reference control signals are presented. In Sect. 4 a nonlinear controller is designed using backstepping technique. The performance of the resulting controller is validated in Sects. 5 through simulations and experimental tests. Finally, Sect. 6 summarizes major conclusions and future research directions.

2 System Description

The robot's movement equations are based on a progressive use of the corresponding mechanical energy conservation laws for each stage of movement, including impacts between the leg and the robot's body. The equations that described robot's movement during the cycle number i were detailed derived in [5]:

$$MgS_i + \frac{MV_{1i}^2 + cl^2}{2} = \frac{c(S_i + l)^2}{2} \ , \tag{1}$$

$$\frac{\eta c(S_i + l)^2}{2} = \frac{MV_{2i}^2}{2} + Mg(S_i + l) \ , \tag{2}$$

with

$$V_{1i} = \sqrt{2gh_i}, \qquad MV_{2i} = (M + m)V_{3i}, \qquad \frac{V_{3i}^2}{2} + gl = gh_{i+1} \ , \tag{3}$$

and where M is the sum of the robot's body mass, the mass of all elements rigidly connected with the robot and a half of a mass of resilient elements, m is the sum of the robot's leg mass and a half of a mass of resilient elements, g is the gravitational acceleration, S_i is a displacement magnitude of the robot's body before full stop, V_1 is the robot's velocity before robot's leg impact against bearing surface, V_2 is the velocity of the robot's body immediately before the impact of the leg stop block against the robot's body, c is the spring constant, η is the efficiency of springs, h_i is the apex height of previous hop, h_{i+1} is the desired apex height of next hop, and l is the springs' extension.

If losses inside springs are neglected, then $\eta = 1$, and the relationship between the required deformation of springs l and next apex height of hop h_{i+1} is given by:

$$l = \frac{-2\,mgM - m^2 g + \sqrt{gC_l}}{cM} \tag{4}$$

where

$$C_l = m^2 g\left(4M^2 + 4mM + m^2\right) + 2ch_{i+1}\left(M^2 + 2Mm + m^2\right) - 2cM^3 h_i \ . \tag{5}$$

The angular position of the cam θ as function of the springs deformation is given by:

$$\theta = 0.5 + \left(\frac{l}{\zeta}\right)^2 \tag{6}$$

where ζ is a parameter determined by the shape of the cam. The dynamic equation of the cam is given by:

$$J\ddot{\theta} = K_G M_o - M_{ext} \ , \tag{7}$$

where J is the inertia of the cam, $\ddot{\theta}$ is the angular acceleration of the cam, K_G is the constant transmission ratio, M_o is the moment that actuates over the cam, and M_{ext} is the external moment. The rotor equation is given by:

$$(J_M + J_G)\,\ddot{\theta}_M = \tau_M - M_o - (b_M + b_G)\,\dot{\theta}_M \ , \tag{8}$$

where J_M is the rotor inertia, J_G is the gearhead inertia, $\ddot{\theta}_M$ is the angular acceleration of the rotor, $\dot{\theta}_M$ is the angular velocity of the rotor, τ_M is the motor torque, M_o is the moment that actuates over the rotor, b_M is the viscosity friction coefficient on the motor shaft, and b_G is the viscosity friction coefficient on the gearhead. The motor torque τ_M is given by:

$$\tau_M = \frac{K_m}{R_M}\left[u - K_E\dot{\theta}_M\right] \ , \tag{9}$$

where K_m is the torque constant, u is the input control motor terminal voltage, R_M is the motor resistance, and K_E is the back – EMF constant.

The angular position of rotor as function of the angular position of the cam is given by:

$$\theta_M = K_G\theta \ . \tag{10}$$

Combining (7–10), the dynamic model of the system becomes:

$$\left(\frac{J}{K_G^2} + J_M + J_G\right)\ddot{\theta}_M + \left[\frac{K_E K_m}{R_M} + b_M + b_G\right]\dot{\theta}_M + \frac{M_{ext}}{K_G} - \frac{K_m}{R_M}u = 0 \ . \tag{11}$$

If the state variables are denoted by: x_1 – angular position of the rotor and x_2 – angular velocity of the rotor, then the space state model is given by:

$$x = \begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}x_1 \\ x_2\end{bmatrix} \tag{12}$$

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{J_{eq}}\left[-\left(\frac{K_E K_m}{R_M} + b_M + b_G\right)x_2 - \frac{M_{ext}}{K_G} + \frac{K_m}{R_M}u\right]\end{aligned} \tag{13}$$

where

$$J_{eq} = \frac{J}{K_G^2} + J_M + J_G \ . \tag{14}$$

3 Time Optimal Control and Minimum Energy Control

In this section, the time optimal control problem and the minimum energy control problem [10] are solved to obtain the optimal laws of motion that will be used as reference control signals for the controller. The Pontryagin's Minimum Principle is applied to find the necessary conditions for optimality. Next, the equations for the state and costate vector that satisfy the necessary conditions are determined and subsequently, the control sequences that are candidates for optimal control are obtained.

3.1 Time Optimal Control

The control problem is to minimize the cost functional

$$\Psi(u_i) = \int_{t_0}^{T} dt = T - t_0, \qquad T \text{ is free} . \tag{15}$$

The Hamiltonian function for the system (13) and the cost functional (15) is given by:

$$H = 1 + x_2(t)\, p_1(t) - K_1 x_2(t)\, p_2(t) - K_2 p_2(t) + K_3 u(t)\, p_2(t) \ , \tag{16}$$

where

$$K_1 = \frac{1}{J_{eq}} \left(\frac{K_E K_m}{R_M} + b_M + b_G \right), \quad K_2 = \frac{M_{ext}}{J_{eq} K_G}, \quad K_3 = \frac{K_m}{J_{eq} R_M} . \tag{17}$$

The control which absolutely minimizes the Hamiltonian is given by:

$$u^*(t) = -\lambda \text{sgn}\{K_3 p_2(t)\} = \pm\lambda, \qquad \text{where} \quad \lambda = \text{constant} . \tag{18}$$

The costate variables $p_1(t)$ and $p_2(t)$ satisfy the equations:

$$\dot{p}_1(t) = -\frac{\partial H}{\partial x_1(t)} = 0 , \tag{19}$$

$$\dot{p}_2(t) = -\frac{\partial H}{\partial x_2(t)} = -p_1(t) + K_1 p_2(t) \ . \tag{20}$$

If π_1 and π_2 are the initial values of the costate variables,

$$\pi_1 = p_1(0), \qquad \pi_2 = p_2(0) \ , \tag{21}$$

then, from (19) and (20) it follows that:

$$p_1(t) = \pi_1 = \text{constant}; \tag{22}$$

$$p_2(t) = \left(\pi_2 - \frac{\pi_1}{K_1} \right) e^{K_1 t} + \frac{\pi_1}{K_1} . \tag{23}$$

From (23) it is immediately concluded that for all possible values of π_1 and π_2, the function $p_2(t)$ can be zero at most once; it follows from (18) that the time optimal control is piecewise constant and can switch, at most once, because the problem is normal [10]. Then, only the four control sequences

$$\{+\lambda\}, \{-\lambda\}, \{+\lambda, -\lambda\}, \{-\lambda, +\lambda\} \tag{24}$$

can be candidates for time-optimal control. Note that two of the necessary conditions (namely, (18) and (22), (23)) have helped us to isolate only four

possible control sequences. Since, over a finite interval of time, the time optimal control is constant, the system (13) can be solved using (18) with the initial conditions

$$x_1(0) = \xi_1 \qquad \text{and} \qquad x_2(0) = \xi_2 \tag{25}$$

to obtain the relations:

$$x_1(t) = \xi_1 + \frac{\xi_2}{K_1}\left(1 - e^{-K_1 t}\right) + \frac{K_3\lambda - K_2}{K_1^2}\left(K_1 t - 1 + e^{-K_1 t}\right) \tag{26}$$

$$x_2(t) = \xi_2 e^{-K_1 t} + \frac{K_3\lambda - K_2}{K_1}\left(1 - e^{-K_1 t}\right) \tag{27}$$

Next, the time t is eliminated to find that:

$$\begin{aligned} x_1 = \xi_1 &+ \frac{\xi_2}{K_1} - \frac{\xi_2\left(x_2K_1 + K_2 - K_3\lambda\right)}{K_1\left(\xi_2K_1 + K_2 - K_3\lambda\right)} \\ &+ \frac{K_3\lambda - K_2}{K_1^2}\log\left(\frac{\xi_2K_1 + K_2 - K_3\lambda}{x_2K_1 + K_2 - K_3\lambda}\right) \\ &+ \frac{K_3\lambda - K_2}{K_1^2}\left(\frac{x_2K_1 + K_2 - K_3\lambda}{\xi_2K_1 + K_2 - K_3\lambda} - 1\right), \end{aligned} \tag{28}$$

where

$$t = \frac{1}{K_1}\log\left(\frac{\xi_2K_1 + K_2 - K_3\lambda}{x_2K_1 + K_2 - K_3\lambda}\right). \tag{29}$$

Equation (28) is the equation of the trajectory in the plane x_1x_2 originating at $(\xi_1,\ \xi_2)$ and due to the action of the control (18). Since the control must be piecewise constant, it is possible to find the locus of points $(x_1,\ x_2)$ which can be transferred to $(0,\ 0)$ using $u = \pm\lambda$. In this way, the γ_+ curve is the locus of all points $(x_1,\ x_2)$ which can be forced to $(0,\ 0)$ by the control $u = +\lambda$, and the γ_- curve is the locus of all points $(x_1,\ x_2)$ which can be forced to $(0,\ 0)$ by the control $u = -\lambda$. Then, the γ curve, called the switch curve, is the union of the γ_+ and γ_- curves. Its equation is:

$$\begin{aligned} \gamma = \Bigg\{(x_1, x_2) : x_1 = &-\frac{x_2}{K_1}\left(1 - \frac{K_3\lambda x_2}{K_1x_2\left|x_2\right| - K_3\lambda x_2}\right) \\ &- \frac{K_3\lambda x_2}{K_1^2\left|x_2\right|}\log\left(\frac{K_1x_2\left|x_2\right| - K_3\lambda x_2}{-K_3\lambda x_2}\right) \\ &+ \frac{K_3\lambda x_2}{K_1^2\left|x_2\right|}\left(1 + \frac{K_3\lambda x_2}{K_1x_2\left|x_2\right| - K_3\lambda x_2}\right)\Bigg\} = \gamma_+ \cup \gamma_- \end{aligned} \tag{30}$$

The time optimal control u^* as function of x_1 and x_2 is given by the equation:

$$u^* = \lambda sign\left\{-\frac{K_1x_1}{K_3} - \frac{x_2}{K_3} - \frac{\lambda x_2}{K_1\left|x_2\right|}\log\left(\frac{\lambda K_3x_2}{K_1x_2\left|x_2\right| + \lambda K_3x_2}\right)\right\} \tag{31}$$

The method of evaluating the optimal time t^* is to compute the time required to force (x_1, x_2) to the switch curve γ (t_{sw}) and then compute the time required to go from the point of intersection (w_1, w_2) on the switch curve to the origin $(0, 0)$, using the time optimal control law provided by (18). Its equation is given by:

$$t^* = t_{sw} - \frac{1}{K_1} \ln \left(\frac{K_3 u^*}{K_3 u^* + w_2 K_1} \right) \tag{32}$$

where switching time t_{sw} is defined as:

$$t_{sw} = -\frac{1}{K_1} \ln \left(\frac{w_2 K_1 - K_3 u^*}{-K_3 u^* + \xi_2 K_1} \right) \tag{33}$$

3.2 Minimum Energy

The minimum energy control problem is solved by determining the control input $u(t)$, which takes the system (13) from the initial state $x_1(0) = \xi_1, x_2(0) = \xi_2$ to the final state $x_1(T_f) = x_2(T_f) = 0$ and minimizes the cost energy functional

$$E = E(u, x_2) = \frac{1}{R_M} \int_0^{T_f} \left(u^2 - K_E x_2 u \right) dt \,, \tag{34}$$

where T_f is specified. In order to find the optimal control, it is necessary to find the control that satisfies the necessary conditions, i.e., the extremal control. Then, the first step is to determine the Hamiltonian for the system (13) and the cost functional (34); the Hamiltonian is given by the equation:

$$\begin{aligned} H = {} & \frac{1}{R_M} [u^2(t) - K_E x_2(t) u(t)] + x_2(t) p_1(t) \\ & - K_1 x_2(t) p_2(t) - K_2 p_2(t) + K_3 u(t) p_2(t) \,. \end{aligned} \tag{35}$$

The extremal control must minimize the Hamiltonian; since the Hamiltonian is a quadratic function of $u(t)$, it is possible to find the extremal control by setting $\frac{\partial H}{\partial u(t)} = 0$ and by checking whether or not $\frac{\partial^2 H}{\partial u(t)^2}$ is positive. Since

$$\frac{\partial H}{\partial u(t)} = \frac{2u(t)}{R_M} - \frac{K_E x_2(t)}{R_M} + K_3 p_2(t) \,, \tag{36}$$

and since

$$\frac{\partial^2 H}{\partial u(t)^2} = \frac{2}{R_M} \,, \tag{37}$$

it is concluded that the extremal control is given by:

$$u_E{}^* = \frac{R_M}{2} \left(\frac{K_E x_2(t)}{R_M} - K_3 p_2(t) \right) \qquad t \in [0, T_f] \,. \tag{38}$$

The costate variables $p_1(t)$ and $p_2(t)$ must satisfy the differential equations

$$\dot{p}_1(t) = -\frac{\partial H}{\partial x_1(t)} = 0 \,, \tag{39}$$

$$\dot{p}_2(t) = -\frac{\partial H}{\partial x_2(t)} = \frac{K_E u(t)}{R_M} - p_1(t) + K_1 p_2(t) \,. \tag{40}$$

Bearing in mind that π_1 and π_2 are the initial values of the costate variables, it is possible to find p_1 and p_2. The next step is, therefore, the determination of π_1 and π_2 in terms of ξ_1 and ξ_2, and T_f.

In minimum energy control, the choice of the terminal time T_f is quite critical. In our case, if the movement is performed too fast, the motor will have to hold the springs' extension a longer time, causing a bigger energy consumption. Therefore, the 95% of the flight time $t_{of\ flight}$ will be used as T_f, keeping a 5% as reserve to compensate possible disturbances. The time of flight $t_{of\ flight}$ is given by:

$$t_{of\ flight} = \sqrt{\frac{2h_{i+1}}{g}} + \sqrt{\frac{2\,(h_{i+1} - l)}{g}} \,. \tag{41}$$

4 Controller

The backstepping design technique has been used to develop a Lyapunov-based controller for the resonance hopping robot that achieves asymptotic tracking of the reference control signals which have been suitably selected using both time optimal and minimum energy control methods. Since the objective is to regulate the apex height, some energy should be added to or removed from the system by adjusting the stored elastic energy during flight in preparation for impact and by releasing it during the stance. In this way, two events in the hopping cycle determine the actuation of the controller: lift-off and touchdown. At lift-off, the controller uses both the previous apex height and the goal apex height to calculate the required springs deformation and subsequently the required angular position of the motor with (4) and (10) respectively. With this computed position of the motor and the results from the minimum energy problem, the reference control signals are obtained and utilized in real time by the controller to find out the control input signal during flight time. In the same manner, at touchdown, the controller determines the control input signal using the final state information of the flight phase and the results from the time optimal problem.

The backstepping procedure is applied as follows. Firstly, a coordinate transformation is introduced for the system (13):

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= -K_1 x_2 - K_2 + K_3 u - \ddot{r} \,, \end{aligned} \tag{42}$$

where $e_1 = x_1 - r$ and $e_2 = x_2 - \dot{r}$; e_1 and e_2 express the position and the velocity tracking errors.

$$r = x_1^* \text{ computed with } \begin{cases} {u_E}^* - \text{flight phase} \\ u^* - \text{stance phase} \end{cases} \tag{43}$$

$$\dot{r} = x_2^* \text{ computed with } \begin{cases} {u_E}^* - \text{flight phase} \\ u^* - \text{stance phase} . \end{cases} \tag{44}$$

Now a smooth, positive definite Lyapunov like function is defined:

$$V_1 = \frac{1}{2} e_1^2 . \tag{45}$$

Its derivative is given by:

$$\dot{V}_1 = e_1 e_2 . \tag{46}$$

Next, e_2 is regarded as a virtual control law to make $\dot{V}_1$ negative. This is achieved by setting e_2 equal to $-k_1 e_1$, for some positive constant k_1. To accomplish this, an error variable z_2 that we would like to set to zero is introduced:

$$z_2 = e_2 + k_1 e_1 . \tag{47}$$

Then $\dot{V}_1$ becomes:

$$\dot{V}_1 = z_2 e_1 - k_1 e_1^2 . \tag{48}$$

To backstep, the system is transformed into the form:

$$\begin{aligned} \dot{e}_1 &= z_2 - k_1 e_1 \\ \dot{z}_2 &= -K_1 (z_2 - k_1 e_1 + \dot{r}) - K_2 + K_3 u - \ddot{r} + k_1 \dot{e}_1 \end{aligned} \tag{49}$$

Now, a new control-Lyapunov function V_2 is built by augmenting the control-Lyapunov function V_1 obtained in the previous step using a stabilization function. This function penalizes the error between the virtual control and its desired value. So, taking

$$V_2 = V_1 + \frac{1}{2} \kappa z_2^2 , \tag{50}$$

as a composite Lyapunov function, we obtain:

$$\dot{V}_2 = -k_1 e_1^2 + \kappa z_2 \Big(\frac{e_1}{\kappa} - K_1 z_2 + K_1 k_1 e_1 - K_1 \dot{r} - K_2 + K_3 u - \ddot{r} + k_1 \dot{e}_1 \Big) . \tag{51}$$

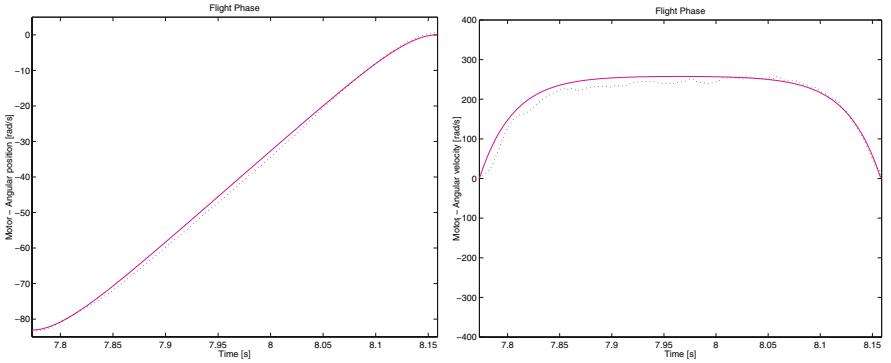
Choosing the control input

$$u = \frac{1}{K_3} \Big(-\frac{e_1}{\kappa} - K_1 k_1 e_1 + K_1 \dot{r} + K_2 + \ddot{r} - k_1 \dot{e}_1 \Big) \tag{52}$$

yields

$$\dot{V}_2 = -k_1 e_1^2 - \kappa K_1 z_2^2 , \tag{53}$$

where $k_1, k_2 > 0$. This implies asymptotical stability according to Lyapunov stability theorem.



(a) (b)

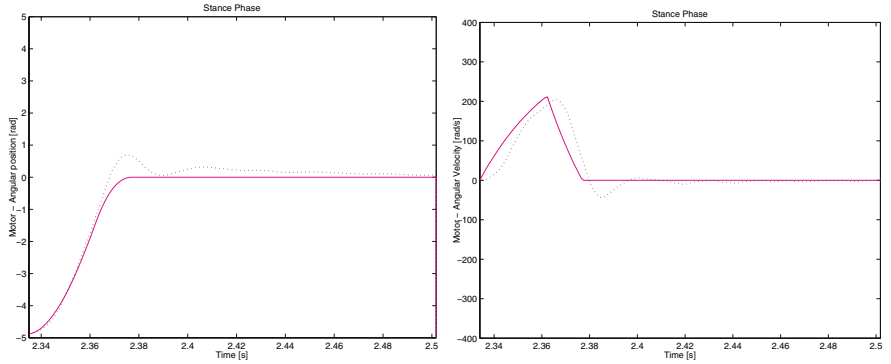
Fig. 1. Experimental results for the flight phase

5 Simulations and Experimental Results

To validate the results of calculations and evaluate the performance of the height controller, simulations and several experimental tests were carried out using a prototype of the resonance hopping robot (Fig. 3a.). In all experiments an acceptable tracking of the optimal reference signals was obtained with a reasonable control effort. Figures 1 and 2 show the tracking performance during flight and stance phases respectively.

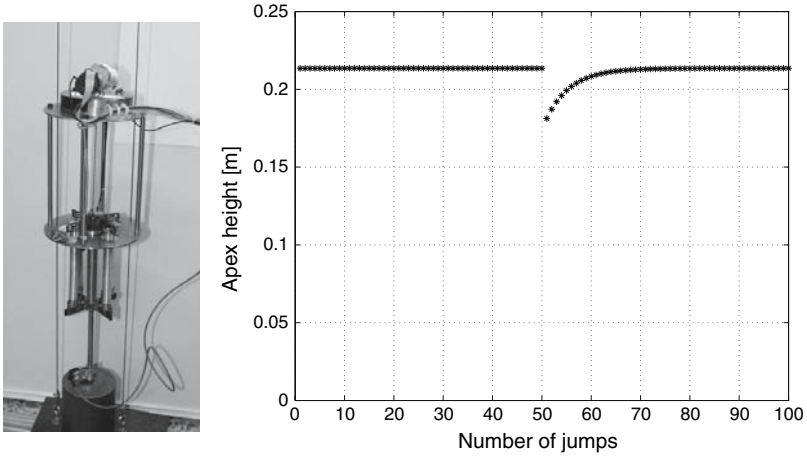
It was found by means of calculations, simulations and experiments, that the resonance hopping robot has a specific property: during one cycle it is possible to change the height of hope only within certain limits

$$h_{i+1}^{\min} \leq h_{i+1} \leq h_{i+1}^{\max} \ , \tag{54}$$



(a) (b)

Fig. 2. Experimental results for the stance phase



(a) (b)

Fig. 3. a. Prototype of the resonance hopping robot. **b**. Apex height vs. Number of jumps

where values of $h_{i+1}^{\min}$ and $h_{i+1}^{\max}$ are determined by the height of previous jump h_i and the desired height h_{i+1}. This fact is connected with natural control limitations: during one cycle it is possible to add only certain value of energy from zero up to a maximum value that is defined by springs deformation in condition of maximum radius of cam. This effect is especially important when desired height of a jump is near extremes. Figure 3b illustrates this effect. The hopper under consideration was working with a maximum possible apex height of 0.22 m during 50 jumps. Then, the desired value of height was changed to 0.18 m. The controller regulated the apex height during one hop (the desired value was in accord with (54)). After this jump of 0.18 m, the height set point was changed to its maximum value 0.22 m again. As this value is not in accord with (54), it was impossible for the controller to achieve the desired value of height in one cycle, requiring several cycles to accomplish it.

6 Conclusions and Future Developments

In this paper, the method for controlling the apex height of a special resonance hopping robot has been presented. The solution includes the derivation of the physical model, the analytical calculation for the time optimal and minimum energy problems in order to facilitate an on-line computation of the optimal reference control signals, and the design of a tracking controller using the integrator backstepping. Experimental results show the controller effectiveness and demonstrate that the control objectives were accomplished.

Future developments are directed towards development of controllers that would provide desired robustness properties in the presence of inevitable modelling uncertainties.

Acknowledgment

The authors would like to acknowledge the financial support from Ministry of Science and Technology of Spain (Project "Theory of optimal dual drives for automation and robotics") and from Ministry of Culture and Education of Spain (Fellowship F.P.U.).

References

1. Raibert M (1986) Legged Robots that Balance. MIT Press, Cambridge, MA.
2. Zeglin G (1999) The bowleg-hopping robot. PhD Thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
3. Ringrose R (1997) Self-stabilizing running. In: IEEE Int. Conf. Robotics and Automation (ICRA), IEEE Press.
4. Akinfiev T, Armada M, Fernández R, Montes H (2002) Hopping robot and its control algorithm. Patent Application, P200201196, Spain.
5. Akinfiev T, Armada M, Fernández R, Gubarev V, Montes H (2004) Dual Drive for Vertical Movement of Resonance Hopping Robot. International Journal of Humanoid Robotics, World Scientific Publishing Company. (In press).
6. Matsuoka K (1979) A model of repetitive hopping movements in man. In: Proc. Fifth World Congr. Theory of Machines and Mechanisms, IFIP.
7. Khalil H (2002) Nonlinear Systems. Prentice Hall, New York.
8. Kokotovic P (1992) The Joy of Feedback: Nonlinear and Adaptive. In: IEEE Contr. Sys. Mag., 12:7–17.
9. Krstic M, Kanellakopoulos I, and Kokotovic P (1995) Nonlinear and Adaptive Control Design. Wiley, New York.
10. Athans M, Falb P (1966) Optimal Control. McGraw-Hill Book Company, New York.

Zero Moment Point Modeling Using Harmonic Balance

R. Caballero* and M. Armada

Industrial Automation Institute, Spanish Council for Scientific Research, Madrid, Spain

Abstract. This paper present Zero Moment Point (ZMP) frequency domain based mathematical models are developed to establish a relationship among the robot angular trajectories and the stability margin of the foot (or feet) contact surface and the supporting ground. This proposed ZMP modeling approach not only considers classical rigid body model uncertainties but also non-modeled robot mechanical structure vibration modes. The effectiveness of this model is tested with a simplified biped robot prototype subjected to sinusoidal external disturbances.

1 Introduction

The control design of biped robotic systems has been an important research area for many scientists in the last few years. Due to fact that, biped machines are so complex, most of these control designs involve the use of simplified dynamical modeling. Usually, biped robots are not only modeled with rigid links and ideal torque sources, but also, considered sagittal and lateral plane dynamics decoupled. It is clear that this simplification is important for control architecture design and for real time motion planning. However, the use of an oversimplified model in presence of non-modeled structure elastic modes and non-modeled joint mechanical backlash could compromise the foot-ground stability.

As it is well known, one of the most effective ways to analyze the foot-ground stability of biped robots is the so called Zero Moment Point (ZMP), introduced by Vukobratovic [3] to be used as an index of stability for the walking cycle. It has been used successfully by many authors for biped robot trajectory selection [5–7, 9]. The ZMP can be considered as an extension of the centre of mass projection [10]. Commonly, researchers achieve this objective using adaptive gaits like approach to constrain the ZMP within the convex hull of the foot support area.

*Universidad Tecnológica de Panama, Panama, Rep. of Panama

In this paper, a ZMP frequency response model is proposed. This proposed ZMP modeling approach not only considers classical rigid body model uncertainties but also non-modeled robot mechanical structure vibration modes. The effectiveness of this model is tested with a simplified biped robot prototype with ZMP direct feedback subjected to external disturbances.

2 ZMP Model

2.1 Single Support Biped Robot Model

A general single support biped robot dynamical model with n degrees of freedom can be expressed as:

$$M(Q)\ddot{Q} + V\left(Q, \dot{Q}\right) + G(Q) = T \tag{1}$$

Where,

$Q = \left[\, q_1\ q_2 \cdots q_i \cdots q_n \,\right]^T$ is the angular position vector of robot joints
$\dot{Q} = \left[\, \dot{q}_1\ \dot{q}_2 \cdots \dot{q}_i \cdots \dot{q}_n \,\right]^T$ is the angular velocity vector of robot joints
$\ddot{Q} = \left[\, \ddot{q}_1\ \ddot{q}_2 \cdots \ddot{q}_i \cdots \ddot{q}_n \,\right]^T$ is the angular acceleration vector of robot joints
$T = \left[\, \tau_1\ \tau_2 \cdots \tau_i \cdots \tau_n \,\right]^T$ is the torque vector of robot joints
$M(Q)$ is the inertia matrix
$V\left(Q, \dot{Q}\right)$ is a torque vector produced by centrifugal and coriollis forces
$G(Q)$ is a torque vector produced by gravity force

2.2 Zero Moment Point in Single Support

The zero moment point is the point on the ground where the total moment generated due to gravity and inertia equals zero [4]. Usually, the biped foot mass and ankle height are not important in comparison with full robot. Then, the ZMP (zero moment point) could be estimated using D'Alambert principle (see Fig. 1). Consequently, the ZMP will be function of ankle support foot torque, foot geometry and ground reaction forces:

$$ZMP(t) = \alpha = \frac{\tau_1 - \beta R_y}{R_z} \tag{2}$$

Where,

τ_1 is the ankle support foot torque
β is the ankle height
R_y is the horizontal ground reaction force
R_z is the vertical ground reaction force

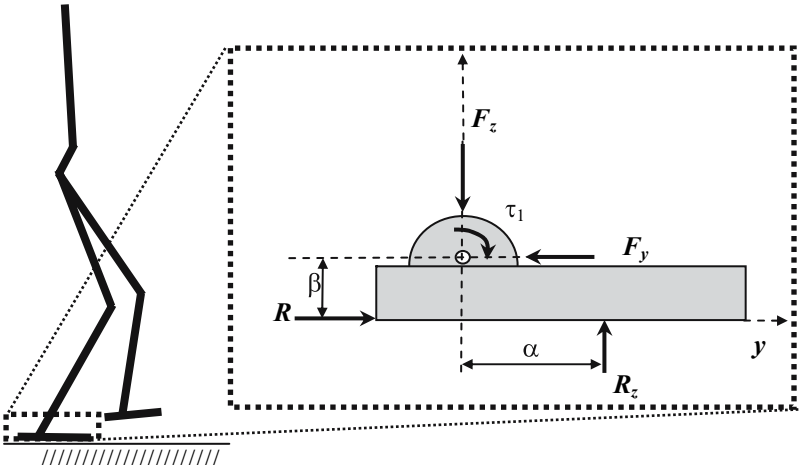


Fig. 1. Present forces in single support for ZMP estimation

3 ZMP Frequency Response Model

3.1 Modeling

Even though, the nonlinear ZMP model (2) is very useful for offline motion planning, it poses some important limitations for real time trajectory planning and/or for being used in designing ZMP based control systems. These limitations are due to the fact that (2) is based in rigid bodies dynamics without backlash effects. One way to overcome these drawbacks is using ZMP frequency response model.

The nonlinear ZMP model (2) could be analyzed in two sections (see Fig. 2). The first section is linear and the second is nonlinear. However, the nonlinear section could be modeled using harmonic balance not only because it has low-pass filter properties but also is time invariant.

$$ZMP(j\omega) = \sum_{i=1}^{n} G_{Ai}\,(j\omega) q_i\,(j\omega) \tag{3}$$

Where,

$$G_{Ai}(j\omega) = G_{NLi}(j\omega)[a_{3i}(j\omega)^2 - a_{1i}]$$

$G_{NLi}(j\omega)$ is the equivalent harmonic balance transfer function for $q_i(j\omega)$

One way to improve the harmonic balance model is using a model that incorporates uncertainties due to links flexibilities, joints backlashes, saturation effects and modeling errors (see Fig. 3).

3.2 Model Correlation

In order to obtain ZMP system uncertainties the biped robot must be excited with a variable frequency sinusoidal signal in each joint. This joint oscillation

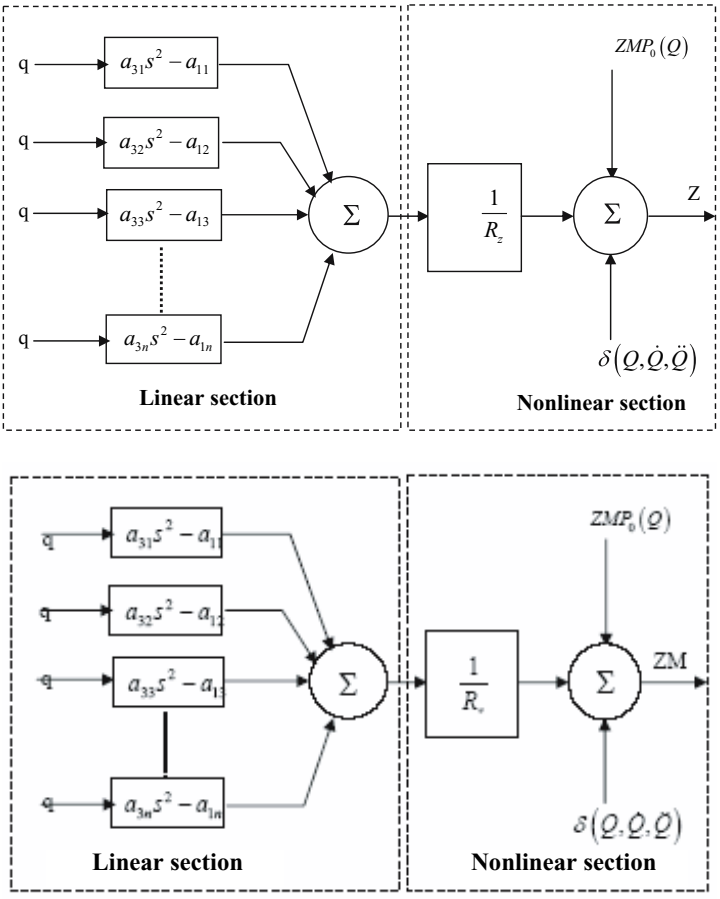


Fig. 2. Nonlinear ZMP single support model

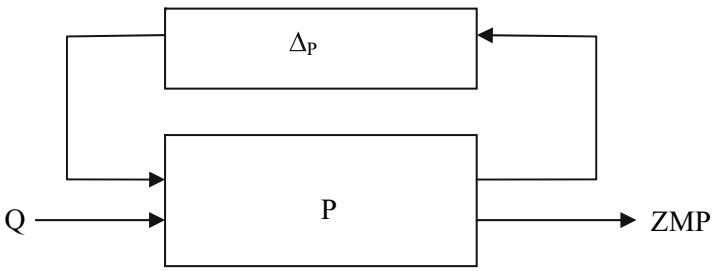


Fig. 3. Single support ZMP model incorporating system uncertainties

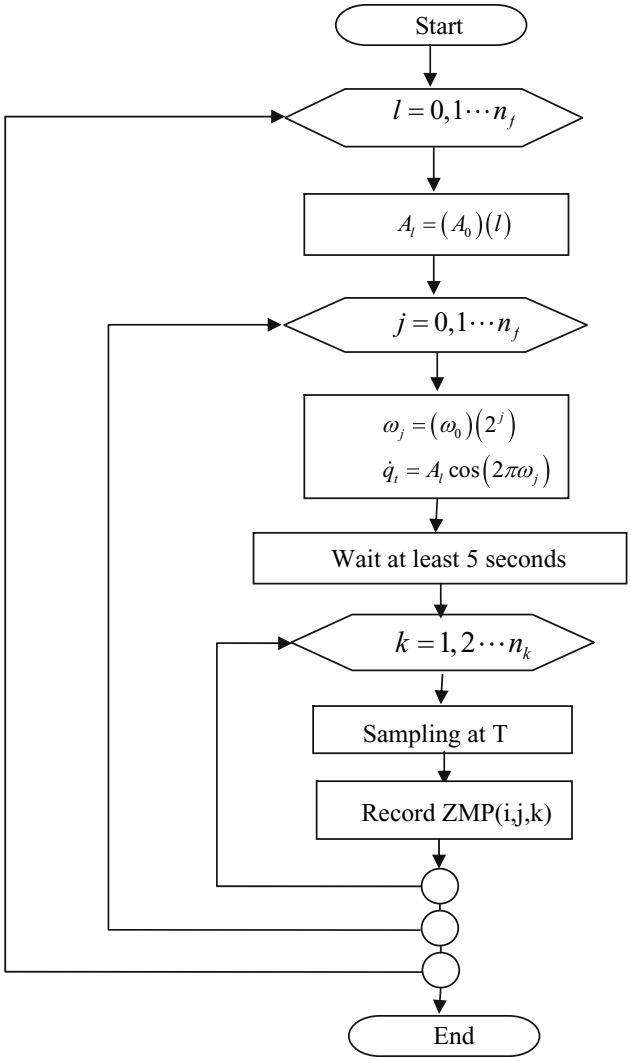


Fig. 4. Algorithm for building-up a data base relating ZMP values with the amplitude and frequency of the harmonic input signal

will generate a ZMP oscillation in ZMP sensor located inside the biped robot foot. Now, the exciting and ZMP signal are processed to obtain $G_{Ai}(j\omega)$ using the algorithms shown in Fig. 4 and Fig. 5.

Now is possible to compare the nominal $\hat{G}Ai(j\omega)$ with the experimental ones and obtain a robust additive uncertainty model framework (see Fig. 6) or a robust multiplicative uncertainty model framework (see Fig. 7), each of them composed by three transfer functions:

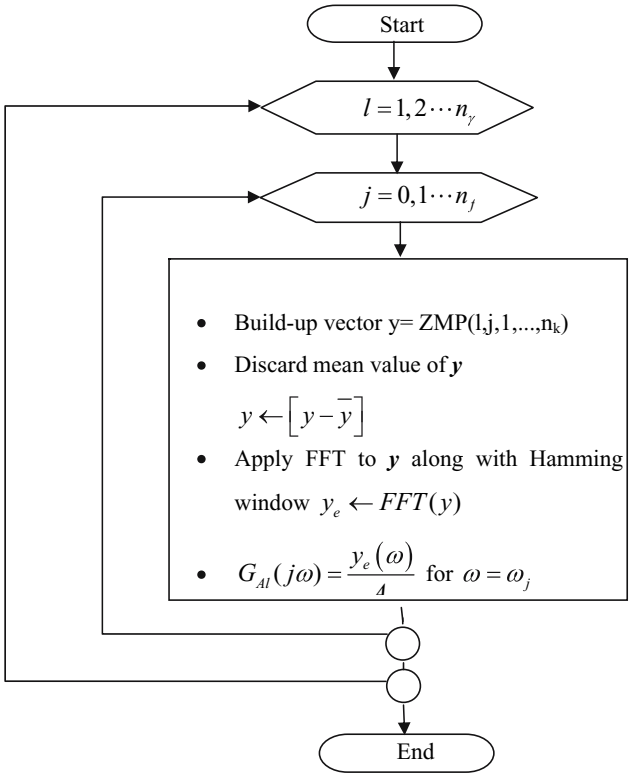


Fig. 5. Algorithm used for finding out $G_{Ai}(j\omega)$ vectors

$\hat{G} \wedge Ai\,(j\omega)$ is the nominal transfer function
$W_G\,(j\omega)$ is the weighting transfer function for multiplicative uncertainty
$\Delta\,(j\omega)$ is each possible transfer function with the property $\|\Delta\,(j\omega)\|_\infty \leq 1$

Consequently, all possible transfer function could be represented using the additive or multiplicative robust framework models shown in Figs. 6

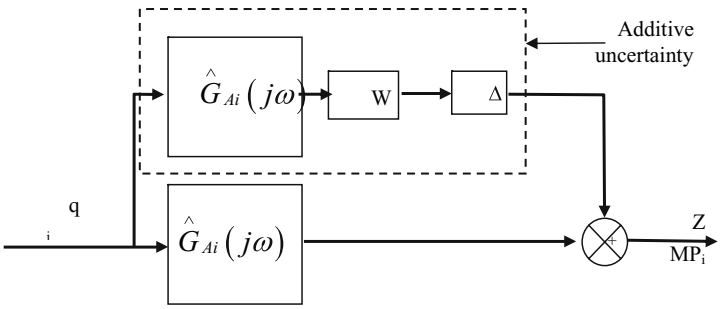


Fig. 6. Robust $G_{Ai}(j\omega)$ model with additive uncertainties effect

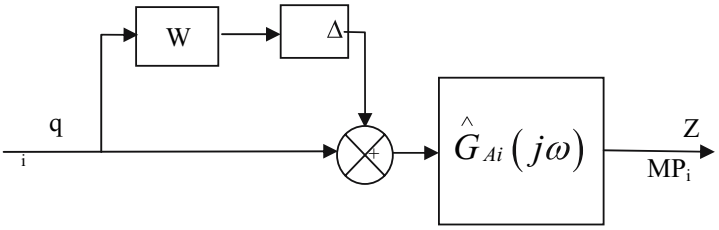


Fig. 7. Robust $G_{Ai}(j\omega)$ model with multiplicative uncertainties effect

and 7. Therefore, each feedback controller that performs an adequate set point regulation for the ZMP robust framework model, will also do it for the nonlinear ZMP model.

4 Experimental Results

The experimental setup consists in a five kilogram two of degree of freedom footed linkage (see Fig. 8). The ZMP model is obtained applying sinusoidal oscillation in each joint and ZMP measuring with the help of foot force sensors.

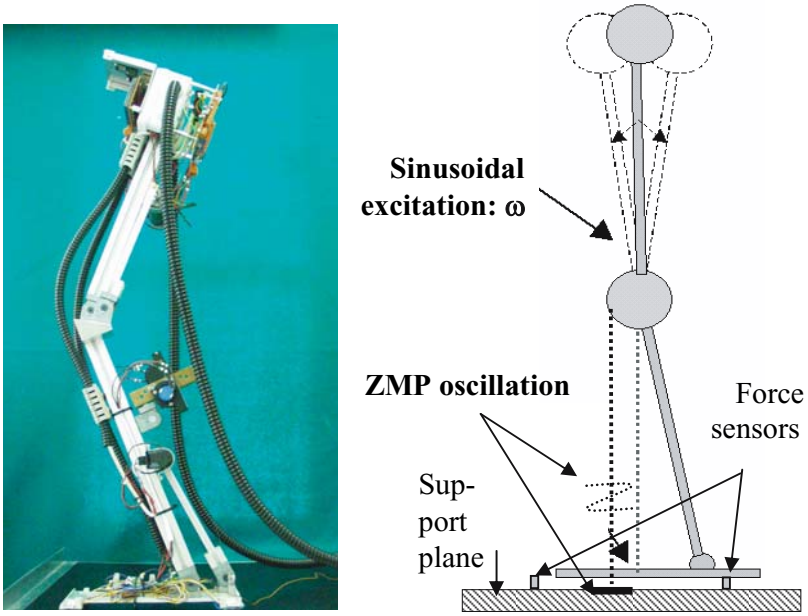


Fig. 8. Two degrees of freedom prototype. ZMP frequency domain model estimation using harmonic oscillations

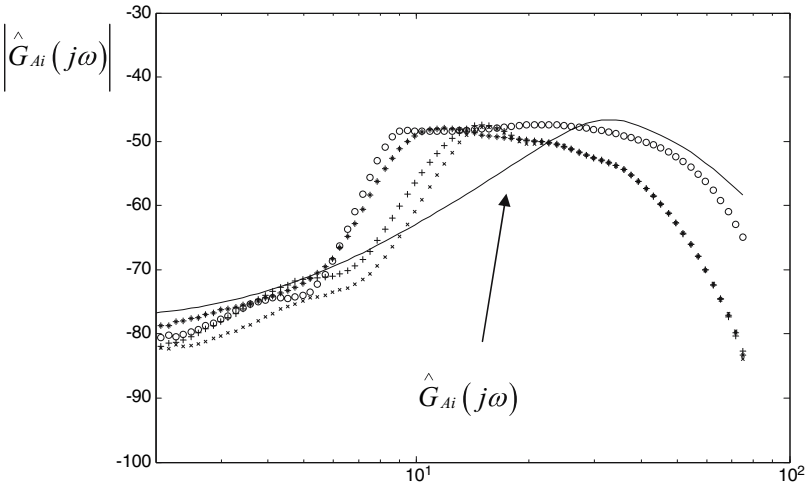


Fig. 9. ZMP amplitude frequency response

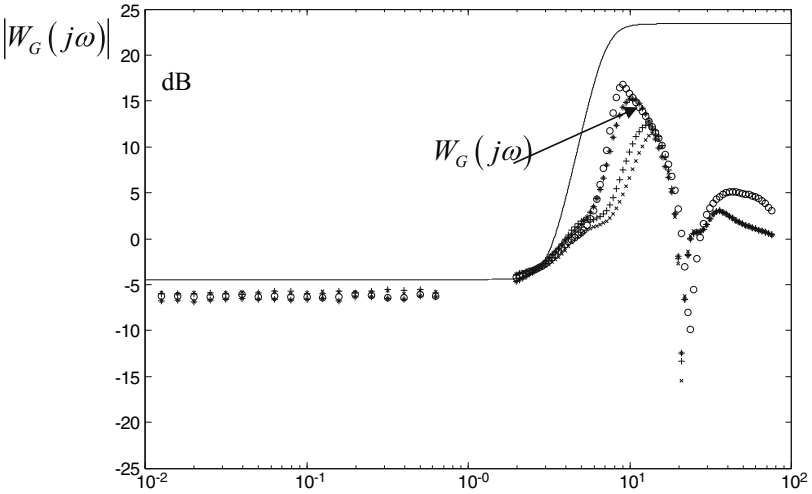


Fig. 10. $W_G\,(j\omega)$ modelling

The ZMP frequency response for different sinusoidal amplitudes and set points is described in Fig. 9. The weighting transfer function $W_G(j\omega)$ for multiplicative uncertainty should be stable and minimum phase. It could be modeled correlating the experimental data for $|W_G(j\omega)| \geq |\frac{G_{Ail}(j\omega)-\hat{G}\,Ai(j\omega)}{\hat{G}\,Ai(j\omega)}|$ (see Fig. 10). In this particular case the experimental data could be correlated with:

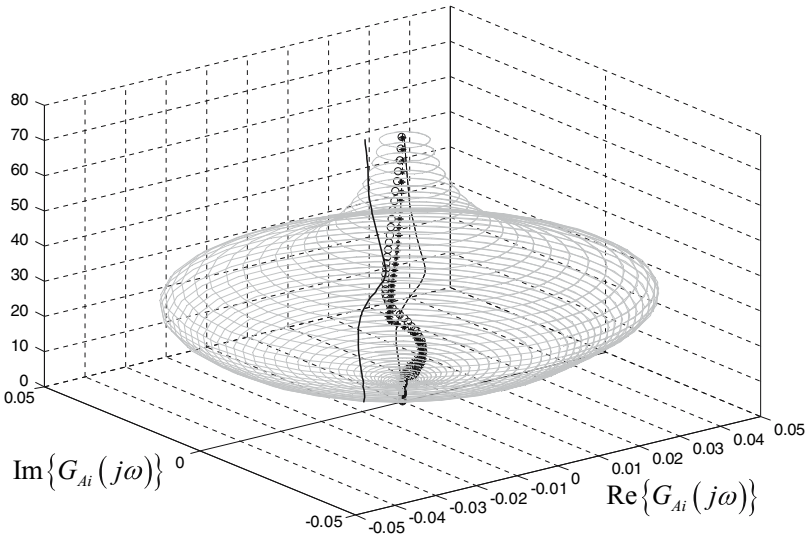


Fig. 11. Nominal model $\hat{G}\,Ai\,(j\omega)$ and uncertainty effect in a 3D Nyquist-frequency response plot

$$W_G\,(j\omega) = a - \frac{b}{d_4\left(\frac{s}{\omega_c}\right)^4 + d_3\left(\frac{s}{\omega_c}\right)^3 + d_2\left(\frac{s}{\omega_c}\right)^2 + d_1\left(\frac{s}{\omega_c}\right) + 1} \tag{4}$$

Where coefficients d_i are Butterworth coefficients for a four order low pass filter.

Finally, the model uncertainty could be represented as radius frequency dependent discs on Nyquist response plot. A 3D Nyquist-frequency response plot (see Fig. 11) shows that uncertainty effect is more important between 10 and 30 rad/s

5 Conclusions and Further Work

Preliminary experiments have confirmed the time invariant and low pass filter properties of ZMP index. In consequence, the ZMP modeling using harmonic balance seems to be validated. Other experiments have tested effectiveness of the ZMP frequency response modeling (please see [2]). However, is necessary to extend the experiments for more degrees of freedom, mostly when a complete biped ZMP stabilization is required

Acknowledgments

The authors would like to acknowledge the funding from Consejería de Educación y Cultura of Comunidad de Madrid, under research project ROBICAM 07T/0022/2001.

References

1. Caballero R., Akinfiev T., Montes H., Armada M. (2001) *On the modelling of SMART non-linear actuator for walking robots.* International Conference of Climbing and Walking Robots 2001, September 24–26, 2001 Karlsruhe, Germany.
2. Caballero R., Akinfiev T., Armada M. (2002) *Robust cascade controller for ROBICAM biped robot: Preliminary experiments.* International Conference of Climbing and Walking Robots, Clawar 2002, Paris, France.
3. Vukobratovic V., Borovac B., D. Surla, Stokic D. (1990) *Biped Locomotion.* Springer-Verlag.
4. Furushu J., Masubushi M. (1986) *Control of a Dynamical Biped Locomotion System for Steady Walking.* Journal of Dynamic Systems, Measurement, and Control. Vol. 108. pp. 111–118.
5. Gienger M., Loffler K., Pfeiffer F. (1999) *Design and control of a biped walking and jogging robot.* International Conference of Climbing and Walking Robots, Clawar 99, pp. 48–58. Portsmouth, United Kingdom.
6. Hirai, K. Hirose, M. Haikawa, Y. Takenaka, T. (1998) *The Development of Honda Humanoid Robot.* Proceedings of the 1998 IEEE. International Conference on Robotics & Automation. Leuven, Belgium.
7. Mita, T. Yamagushi, T. Kashiwase T. and Kawase, T. (1984) *Realization of a high speed biped using modern control theory.* Int. J. Control. Vol. 40. No. 1. pp 107–119.
8. Goswami, A. (1999) *Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point.* The International Journal of Robotic Research. Vol. 18. No. 6. pp. 523–533.

9. Yamaguchi, J. Takanishi, A. Kato, I. (1993) *Development of a biped walking robot compensating for three-axis moment by trunk motion.* Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems. Yokohama, Japan.
10. Hemami, H. Weimer, F. Koozekanani, S. (1973) *Some Aspects of the inverted Pendulum Problem for Modeling of Locomotion Systems.* IEEE Transactions on Automatic Control.

An Introductory Revision to Humanoid Robot Hands

D. Alba, M. Armada, and R. Ponticelli

Instituto de Automática Industrial, Carretera de Campo Real Km 0.200;
La Poveda, Arganda del Rey, 28500 MADRID Apartado 56; ESPAÑA

Summary. In this paper is presented a collection of humanoid anthropomorphic robotic hands, which represent the state of the art at the best of our knowledge, at the date of this paper. It is revised the most important characteristics of the robotic hands and a comparative table is developed. The objective of this paper is try to give a general idea of the most well known robot hands by compiling and comparing principally what it thinks are the most relevant designs.

1 Introduction

The development of a universal manipulator has been the dream of many researchers in robotics, in the last twenty five years thanks to the new technologies (principally the new developments in electronics), the research of anthropomorphic robot hands has had important advances.

At the moment there are many different robot hands designs with 5 or fewer fingers in agreement with the criterion of its inventors, at the same way the actuation type differ in each model, with advantages in some aspects, and disadvantages in other aspects, for example sacrificing force for velocity or less flexible design for some kind of threats. Although this paper does not have all the models available for robot hands, the objective of it is try to give a general idea of the state of the art by compiling and comparing what it thinks are the most relevant designs.

As the complexity of the human hand is very high, the complexity of the different designs is high too, and the number of parameters that we have to deal to evaluate a particular design grows. In order to reach the objective of this paper, the information about the different designs has been put in statistics diagrams and tables that organize in a better way the quantity of information available for each hand, giving us in this way a better approach to compare between them.

2 The Development of Robot Hands in the Past Years

Between 1980 and 1990 there was as few robot hands developed, but in the next years the number robot hands grown, as we can see in the Fig. 1. This figure shows us that the robot hands in the 2000 year had much interest, with important advances as for example Gifu Hand or the DLR II hand. Again the new technologies have an important role in these new robot hands.

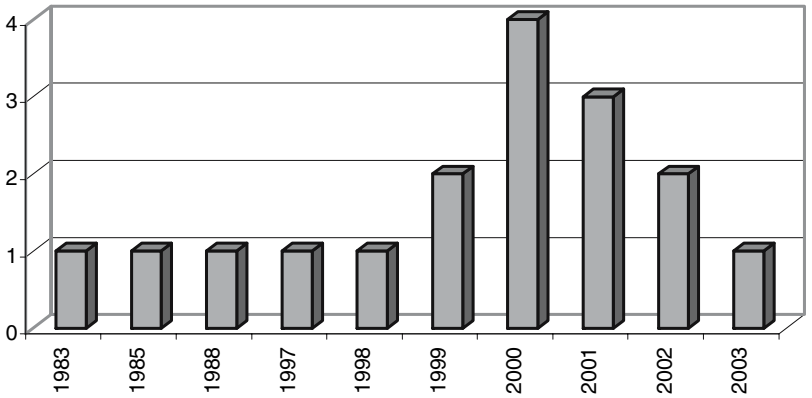


Fig. 1. Number of most well known developed robot hands in the past years

3 Design Parameters Preferred by the Designers

The decision of choosing a particular parameter depends in many cases on which is the field of research of the hand, for example the Stanford/JPL hand is meant to be used un research of pinch grasp, for this reason it has only three fingers and the finger tips have tactile sensors, for this compare robot hands with each other is very difficult due to the robot hands have different fields of applications.

The preference of the designers for the number of fingers is to use 5 fingers for a more anthropomorphic hand, as it can see in the Fig. 2 with the half of the hands incorporating this design, the use of 3 or 4 fingers have the same preference approximately between the designers.

Probably the number of degrees of freedom and the number of actuated DOF is one of the most difficult parameter to choose. Again the application of the robot hand is the main factor in this parameter. The different designs prefer more DOF and less actuators as it can see in Fig. 3, this simplifies the control, the needed actuators, the cable and tendons routing, and in general, but not necessarily, for example the NASA's Robonaut hand had a 4 inch diameter forearm, in contrast the Barret hand is very compact with

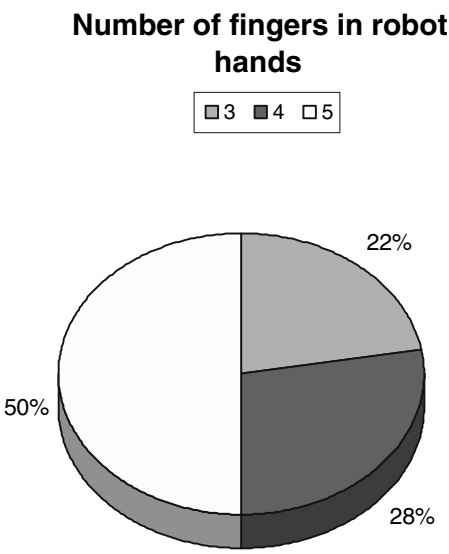


Fig. 2. The preference of the number of fingers in robot hands designs

less actuators as we can see in Fig. 3, an exception of this observation is the Gifu hand, with many actuators has a very compact design but the force of this hand is lower than the Robonaut and the Barret hand. The Variable Force Hand and the Dexterous Robot do not have actuators installed, but the tendons can be manipulated in order to give 10 and 4 active DOFs respectively.

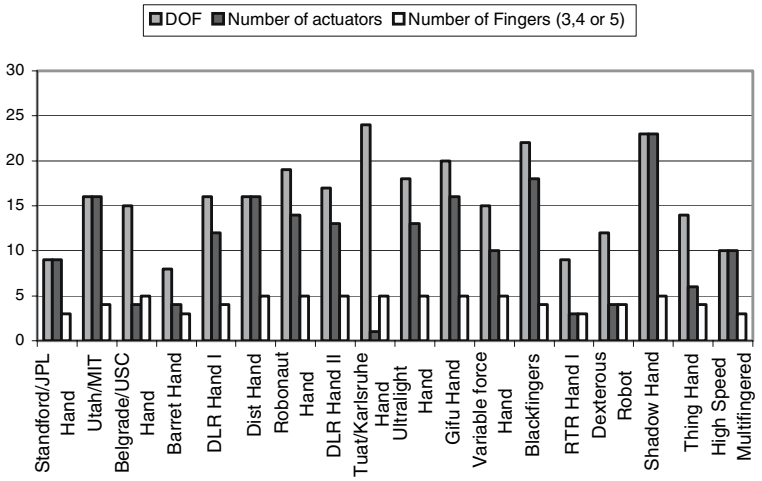


Fig. 3. Number of actuators and degrees of freedom

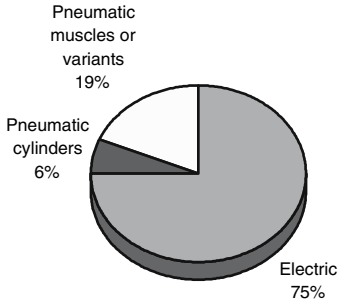


Fig. 4. Preferred actuation type

The preferred actuation type is electric, maybe due to it gives less problems than other actuators. The only hand that uses Pneumatic cylinders in this paper is the Utah/MIT hand, the use of pneumatic muscles has come being increased in the last years, this type of actuation have the advantage of high power in reduces spaces although, the space of the compressor is not taken into account. The Fig. 4 shows the distribution of the different actuation type used in the hands of this paper. The Hitachi robot hand (23) has different actuators based in SMA (Shape Memory Alloys) artificial muscles, with high force in little space, but it has the inconvenient the response is very slow.

The preferred transmission type as it can see in the Fig. 5 is tendons; tendons simplify the design and allow take the actuators off from the hand; however they have the problem that introduces many errors. The gears with zero backslash have more accuracy, but they have more friction too. The direct transmission could be the best, but the difficult on its design make it less used.

The rapidity of the hands is showing in the Fig. 6, the rapidity is expressed in terms of the time it takes from fully closed, to fully open and fully closed again. The graphic is in chronological order approximately, during the years

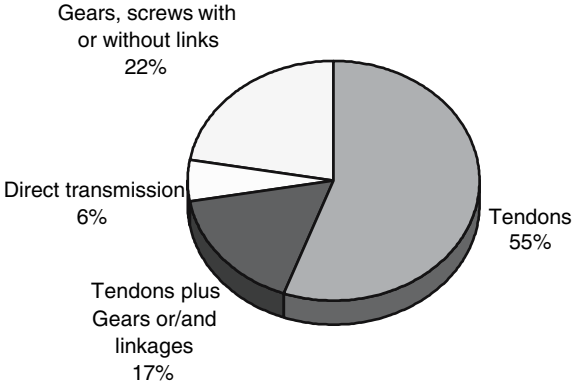


Fig. 5. Preferred transmissions types

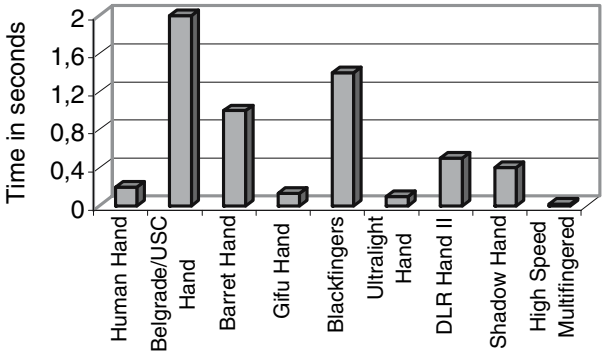


Fig. 6. Time to totally open to totally closed

faster hands have been designed. The Utah/MIT hand has fast actuators with a 20 Hz frequency response, when they working alone. The fastest hand is the High Speed Multifingered hand; this hand is used for catch rubber balls at free falling with artificial vision. In the figure it can see there are several hands with similar rapidity than the human hand.

The average force of the human hand is 15N at the finger tip; in the Fig. 7 we can see that many hands have more force than the human hand, for example the Barret Hand, this hand is used for industrial applications, in this way justifying the high force of this hand. In the case of the High Speed Multifingered hand the force showed is the dynamic force at top speed; this hand has high acceleration actuators, the actuators allow a large current flow for short time.

In the Fig. 8 shows the force-frequency ratio; this ratio has the same units than momentum or impulse. Despising the variation forces of the actuators during the entire movement, the force-frequency ratio can give us an idea of the impulse the fingers can make. The force-frequency ratio is just for

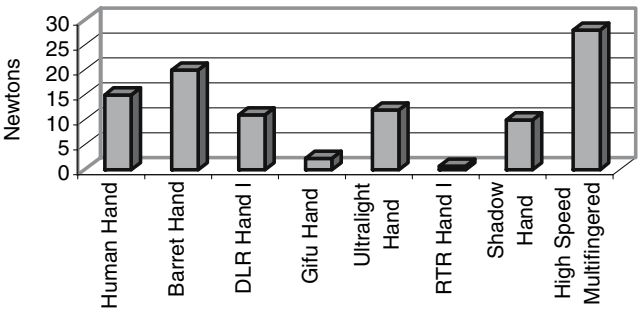


Fig. 7. Force of the hands

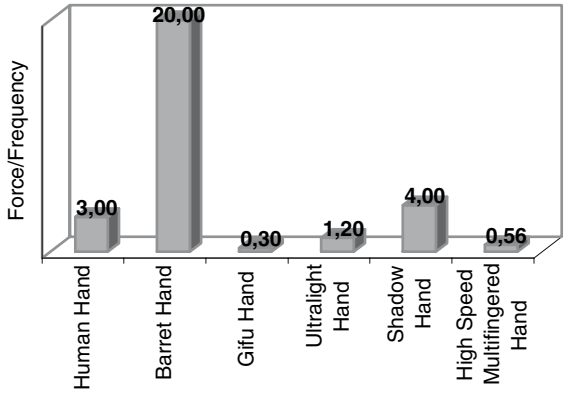


Fig. 8. Force-Frequency Ratio

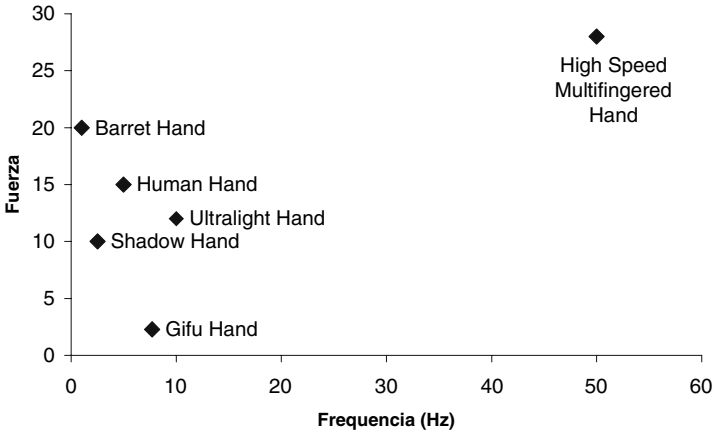


Fig. 9. Frequency-Force Space, a better comparison

comparison proposes, is not a measure of the real momentum of the fingers. The Fig. 9 shows the distribution of the hands in the Frequency-Force space; in this figure it can see in a better way the similarities with the human hand and the robot hands.

4 Characteristics of the Robot Hands

The following tables resume the different characteristics of the hands listed in this paper.

Table 1. Summary of the different Robot Hands examined in this paper

Name (References)	Research Institute	Year	Fingers	DOF	Number of Actuators	Actuation Type
Stanford/JPL Hand (1, 27)	Stanford University	1983	3	9	9	Electrical (DC)
Utah/MIT (20, 25)	Utah University	1985	4	16	16	Pneumatic Cylinders
Belgrade/USC Hand (21)	Belgrade University	1988	5	15	4	Electrical Motors
Barret Hand (2)	Barret Technology Inc.	1988	3	8	4	Electrical (Brushless)
DLR Hand I (3, 4)	DLR-German Aerospace Center	1997	4	16	12	Electrical
Dist Hand (22)	Génova University	1998	5	16	16	Electrical
Robonaut Hand (5, 6, 24)	NASA Johnson Space Center	1999	5	19	14	Electrical (Brushless)
Gifu Hand (10, 11, 12)	Gifu University	1999	5	20	16	DC Micromotors
Blackfingers (14, 30 31)	Politécnico di Milano	2000	5	22	36	Mckibben Actuators
Tuat/Karlsruhe Hand (8)	Tokyo and Karlsruhe Universities	2000	5	24	1	Electrical
Ultralight Hand (9)	Research Center of Karlsruhe	2000	5	18	13	Flexible fluidic actuators
Variable force Hand (13)	Hokkaido University	2000	5	10	5	None
DLR Hand II (7, 26)	DLR-German Aerospace Center	2001	4	17	13	Electrical
RTR Hand I (15)	Centro INAIL RTR, Scuola Superiore Satn' Anna	2001	3	9	3	DC Micromotors
Dexterous Robot (16)	Maryland University	2001	4	12	–	None
Shadow Hand (17)	Robot Shadow Company Ltd.	2002	5	23	23	Pneumatic muscles
Thing Hand (18)	Florida University	2002	4	14	6	Electrical
High Speed Multifingered Hand (19)	Tokyo and Hiroshima Universities	2003	3	10	10	Electrical

Table 2. Summary of the different Robot Hands examined in this paper (cont.)

Name	Sensors	Transmission	Observations
Stanford/JPL Hand	Tactile sensors, Tension tendons sensors	Tendons	One of the firs dexterous robot hands
Utah/MIT	Hall effect sensors at each joint, Tendon tension sensors, tactile distributed sensors	Tendons	One of the must famous hands, widely studied and used in several investigation centers. Have high speed actuators, with 20 Hz when they working alone
Belgrade/USC Hand	A potentiometer as a Position sensor, Interlink force resistor	Links	–
Barret Hand	Optical encoders, Tactile sensors	Tendons, worm gears, clutches	High commercial successful, with several options available, an innovate three finger design, it is able to real-time reconfiguring.
DLR Hand I	28 sensors in each finger, Hall sensors for motor position, Optical position sensor in joints, Torque sensors, X-Y Force Sensors at the Finger Tip, Tactile sensors, Temperature sensors, light barriers, 6 dimensional force sensor at the Wrist, stereo camera integrated in the palm, light protection laser diode	Tendons	1.5 Times the size of the human hand, spiral encoder for measure position, have a lot of sensors
Dist Hand	Rotation sensors in each joint,	Tendons	Low cost actuators
Robonaut Hand	42 sensors, and the research is adding tactile sensors	Tendons	The active DOF of this hand is a good simplification for a dexterous hand

Table 2. (*Cont.*)

Name	Sensors	Transmission	Observations
Gifu Hand	Magnetic encoder in the motors, tactile distributed sensor,	Gears, Links	Integrated MAXON motors in the phalanges
Blackfingers	Position and force sensors directly on the actuator	Tendons	The actuators works in pairs, for this there are only 16 active DOF, for flex or extend the fingers, due to this hand use Mckibben actuators the behavior it is similar to biological muscles, joints and control system biologically inspired.
Tuat/Karlsruhe Hand	–	Tendons, rigid links	–
Ultralight Hand	–	None	Dexterous hand with the actuators directly in the joints.
Variable force Hand	–	Tendons	Adjustable power transmission
DLR Hand II	Torque sensors in each joint	Tendons	More like human pinch grasp, fingers can be bent backwards, 3 DOF palm, Silicon Coated on the finger tips
RTR Hand I	Hall effect sensors, Strain gages as force sensors	Screws	–
Dexterous Robot	None	Tendons, linkages	Each finger is moved by a pair of tendons, the other joints are coupled with the proximal joint, an unusual design that can manipulate human tools
Shadow Hand	Hall effect sensors, Tactile sensors	Tendons	A hand with characteristics very close to the human hand
Thing Hand	Bend Sensors, Camera with simple image processing for the thumb pronation angle	Tendons	A simple design that demonstrate the thumb pronation is sufficient to be dexterous
High Speed Multifingered Hand	Camera for image processing, Strain gages,	Miniharmonic drive, bevel gears	Similar design to the Barret hand, Artificial vision sensors with parallel image processing capability

5 Conclusions

In this paper it has been presented to the best of our knowledge the comparison of different robot hands. The tendency of the different designers is to make fully anthropomorphic robot hands, where most of the hands in this paper have 5 fingers. The designers tend to reduce the number of actuated DOF, for a simple design, the distal phalange is coupled in the majority of the robot hands presented. The tendon driven phalanges with electrical actuators is predominant in almost all robot hands. In accordance of the research, the speed and the force of the hand are selected, not necessarily imitating the human hand.

From the different robot hands exposed in this paper, it can conclude that the development of robot hands still have a long way to go. Many of the robot hands look very advanced but the real capability is still limited, as for example in Robonaut Hand. Although it seems very advanced, there is not experiments with this hand working alone, always is controlled by telepresence. Other example of this is the shadow hand, with a lot of actuated degrees of freedom, an effective control system has been not present yet. The High Speed Multifingered Hand is think to be used in high speed precision grasping; although it can be reconfigured for other grasping the experiments was focused in this type of grasping. The hands exposed here are specialized tools and still do not have the flexibility of the human hand.

References

1. Salisbury JK, Craig JJ (1982) Articulated hands: Force control and kinematics issues. In: International Journal on Robotics Research. pp. 4–20
2. Townsend W (2000) The BarrettHand grasper – programmably flexible part handling and assembly. In: Industrial Robot: An International Journal. 27(3):181–188
3. Butterfass J, Hirzinger G, Knoch S, Liu H (1998) DLR's Multisensory Articulated Hand, Part I: Hard and Software Architecture. In: International Conference on Robotics and Automation, ICRA '98. pp. 2081–2086
4. Butterfass J, Hirzinger G, Meusel P, Liu H (1998) DLR's Multisensory Articulated Hand, Part II: Parallel Torque/Position Control System. In: International Conference on Robotics and Automation, ICRA '98. pp. 2087–2093
5. Lovchic C, et al. (2001) Compact Dexterous Robotic Hand. United States Patent, Patent Number 6,244,644
6. Lovchik C, et al. (2000) Robonaut: NASA's Space Humanoid. In: IEEE Intelligent Systems. 15(4): 57–63
7. Borst C, Fischer M, Hirzinger G (2002) Calculating hand configurations for precision and pinch grasps. In: Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1553–1559
8. Fukaya N, Shigeki T, Afour T, Dillmann R (2000) Design of the TUAT/Karlsruhe Humanoid Hand. In: IEEE/RSJ International Conference on Intelligent Robots and Systems.

9. Schulz S, Pylatiuk C, Bretthauer G (2001) A new ultralight anthropomorphic hand. In: Proceedings ICRA IEEE International Conference Robotics & Automation. Seoul, Korea, pp. 2437–2441
10. Kawasaki H, Shimomura H, Shimizu Y (2001) Educational-industrial complex development of an anthropomorphic robot hand 'Gifu hand'. In: Advanced Robotics, 15(3):357–363
11. Kawasaki H, Komatsu T, Uchiyama K (2002) Dexterous Anthropomorphic Robot Hand With Distributed Tactile Sensor: Gifu Hand II. In: IEEE/ASME Transactions On Mechatronics. 7(3):296–300
12. Mouri T, Kawasaki H, Yoshikawa K, Takai J, Ito S (2002) Anthropomorphic Robot Hand: Gifu Hand III. In: Proceedings of International Conference ICCAS2002. pp. 1288–1293
13. Ishikawa Y, Yu W, Yokoi H, Kakazu Y (2000) Development of Robot Hands with an Adjustable Power Transmitting Mechanism. In: Intelligent Engineering Systems Through Neural Networks. Asme Press, 10: 631–636
14. Folheraiter M, Gini G (2000) Blackfingers: Artificial Hand that Copies Human Hand in Structure, Size and Functions. In: Proceedings of Humanoids 2000
15. Carrozza MC, Massa B, Micera S, Zecca M, Dario P (2001) A "Wearable" Artificial Hand for Prosthetics and Humanoid Robotics Applications. In: Proceedings of the 2001 IEEE – RAS International Conference on Humanoid Robots
16. Akin D, Carignan C, Foster A (2002) Development of a Four-Fingered Dexterous Robot End Effector for Space Operations. In: Proceedings of the ICRA 2002, pp. 2302–2308
17. Shadow Robot Company (2003) Design of a dexterous hand for advanced CLAWAR applications. In: Proceedings of CLAWAR 2003, pp. 691–698
18. Grimm M, Arroyo A, Nechyba M (2002) Thing: A Robotic Hand with Realistic Thumb Pronation. In: Proceedings of the FCRAR 2002
19. Namiki Y, Imai M, Ishikawa M, Kaneko (2003) Development of a High-speed Multifingered Hand System and Its Application to Catching. In: 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. Las Vegas, pp. 2666–2671
20. McCammon I, Jacobsen S (1990) Tactile Sensing and Control for the Utah/MIT Hand. In: Dexterous Robot Hands (eds) Springer-Verlag, pp. 238–266
21. Bekey G, Tomovic R, Zeljkovic I (1990) Control Architecture for the Belgrade/USC Hand. In: Dexterous Robot Hands (eds) Springer-Verlag. pp. 136–149
22. Caffaz A, Cannata G (1998) The Design and Development of the DIST-Hand Dextrous Gripper. In: Proceedings of International Conference on Robotics an Automation (ICRA'98). Leuven, Belgium, pp. 2075–2080
23. Nakano Y, Fujie M, Hosada Y (1984) Hitachi's robot hand. In: Robotics Age. 6(7), pp. 18–20
24. Lovchik C, Diftler M (1999) The Robonaut Hand: A Dexterous Robot Hand For Space. In: Proceedings of the IEEE International Conference on Automation and Robotics. Detroit, Michigan, 2:907–912
Lovchik C, Aldridge H, Diftler M (1999) Design of the NASA Robonaut Hand. In: Proceedings of the ASME Dynamics and Control Division DSC. Nashville, Tennessee, 67:813–830
25. Jacobsen S C, et al. (1986) Design of the Utah/MIT dextrous hand. In: Proceedings of the IEEE International Conference of Robotics and Automation. pp. 1520–1528

26. Butterfass J, Grebenstein M, Liu H, Hirzinger (2001) DLR-hand II: Next generation of a dextrous robot hand. In: Proceedings of the IEEE International Conference of Robotics and Automation, pp. 109–114
27. Salisbury KS, Roth B (1983) Kinematics and Force Analysis of Articulated Mechanical Hands. In: Journal of Mechanism, Transmissions and Actuation in Design, p. 105
28. Lee K, Shimoyama I (1999) A Skeletal Framework Artificial Hand Actuated by Pneumatic Artificial Muscles. In: IEEE International Conference on Robotics and Automation ICRA99. Detroit, Michigan, pp. 926–931
29. Raparelli T, Mattiazzo G, Mauro S, Velardocchia M (2000) Design and development of a pneumatic anthropomorphic hand. In: Journal of Robotic Systems. 17:1–15
30. Folgheraiter M and Gini G (2003) Human-like reflex control for an artificial hand. In: Proceedings of the Fifth International Workshop on information processing in cells and tissues (IPCAT2003). Lausanne, Switzerland, pp. 223–238
31. Folgheraiter M, Gini G (2003) A bio-inspired control system and a VRML simulator for an autonomous humanoid arm. Proceedings of the IEEE Humanoids 2003, Karlsruhe, Germany

Control Architecture of LUCY, a Biped with Pneumatic Artificial Muscles

B. Vanderborght[1], B. Verrelst, R. Van Ham, J. Vermeulen, J. Naudet, and D. Lefeber

Vrije Universiteit Brussel
bram.vanderborght@vub.ac.be
Pleinlaan 2, B-1050 Brussels, Belgium
http://lucy.vub.ac.be

Summary. This paper describes the biped Lucy and it's control architecture that will be used. Lucy is actuated by Pleated Pneumatic Artificial Muscles, which have a very high power to weight ratio and an inherent adaptable compliance. These characteristics will be used to let Lucy walk in a dynamically stable manner while exploiting the adaptable passive behaviour of these muscles. A quasi-static global control has been implemented while using adapted PID techniques for the local feedback joint control. These initial control techniques resulted in the first movements of Lucy. This paper will discuss a future control architecture of Lucy to induce faster and smoother motion. The proposed control scheme is a combination of a global trajectory planner and a local low-level joint controller. The trajectory planner generates motion patterns based on two specific concepts, being the use of objective locomotion parameters, and exploiting the natural upper body dynamics by manipulating the angular momentum equation. The low-level controller can be divided in four parts: a computed torque module, an inverse delta-p unit, a local PI controller and a bang-bang controller. In order to evaluate the proposed control structure a hybrid simulator was created. Both the pneumatics and mechanics are put together in this hybrid dynamic simulation.

1 Introduction

Most of the legged robots nowadays use electrical drives. Well know robots are Asimo [1], Qrio [2], Johnnie [3] and HRP-2P [4]. Because the torquedensity of the drives is too low to actuate legs, gearboxes are used to deliver the required torque at low rotation speeds, thereby making the joint stiff and losing joint compliance. While the compliance characteristics actually can be beneficial for legged locomotion to reduce shocks and decrease energy consumption.

The research group Multibody Mechanics of the Vrije Universiteit Brussel has built the planar walking biped Lucy. This biped model is actuated by

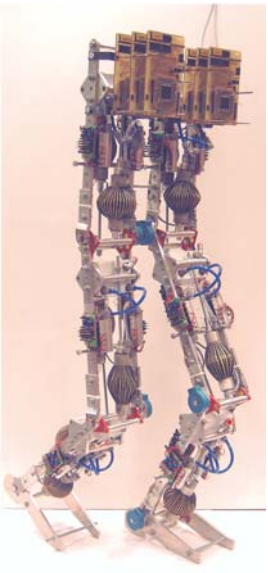


Fig. 1. Photograph of Lucy

pleated pneumatic artificial muscles (PPAM) [5]. These actuators are an alternative to the McKibben type muscle by trying to overcome some of the latter's shortcomings such as a high threshold of pressure and dry friction. The goal of the biped project is to achieve a lightweight bipedal robot able to walk in a dynamically stable way while exploiting the passive behaviour of the pleated pneumatic artificial muscles in order to reduce energy consumption and control effort. Presently Lucy has been assembled and tested. A picture of the complete set-up is given in Fig. 1. The movement of Lucy is restricted to the sagittal plane by a sliding mechanism. The structure is made of a high-grade aluminium alloy, AlSiMg1, and is composed of two legs and an upper body. The robot, all included, weighs about 30 kg and is 150 cm tall. The robot has 12 pneumatic actuators for 6 DOF's.

2 Control Architecture

Presently Lucy has been assembled and debugged, here basic control strategies were implemented. With basic PID techniques already satisfactory behaviour was attained [6]. The following step will be the implementation of a dynamic control scheme to induce faster and smoother motion. An overview of this control architecture is given in the next paragraphs. In order to evaluate the proposed control structure an hybrid simulator was created, which means that both the pneumatics and mechanics are put together in a real-time dynamic simulation. The pressure building inside the muscle is represented by first

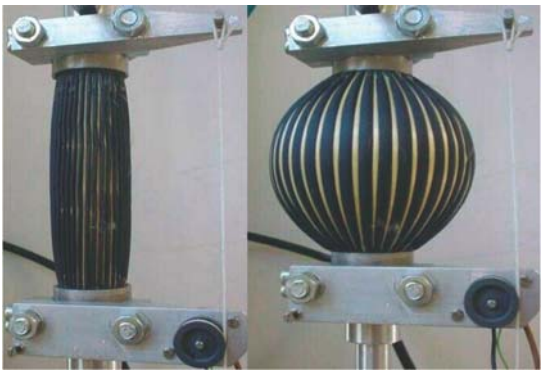


Fig. 2. Photograph of deflated and inflated state of the PPAM

order differential equations deduced from the first law of thermodynamics for an open system while assuming a perfect gas for the compressed air. The orifice valve flows are derived from the model presented by ISO635 [7]. The integration of these first order differential equations coupled with the mechanical differential equations gives the torques.

The considered controller is given in the schematic overview of Fig. 3 and is a combination of a global trajectory planner and a local low-level joint controller. The low-level controller can be divided in four parts: the computed torque module, the inverse Δp unit, the local PI controller and the bang-bang controller. The implementation of the trajectory planner, computed torque module and inverse Δp control is done in a central computer, working with a refresh rate of 500 Hz. Each joint is controlled with a micro-controller working at 2000 Hz and is used for the local PI controller and bang-bang controller. The communication system uses the USB 2.0 protocol.

2.1 Trajectory Planning

The trajectory planner generates motion patterns based on two specific concepts, being the use of objective locomotion parameters, and exploiting the natural upper body dynamics by manipulating the angular momentum equation [8]. The trajectories of the leg links, represented by polynomials, are planned in such a way that the upper body motion is naturally steered, meaning that in theory no ankle torque would be required. To overcome possible external disturbances, a polynomial reference trajectory is established for the upper body motion, which mimics a natural trajectory. Consequently the required ankle torque is low, meaning that it does not cause the Zero Moment Point [9] to move out of the predefined stability region. One of the most interesting aspects of this method is that they are based on fast converging iteration loops, requiring only a limited number of elementary calculations. The computation time needed for generating feasible trajectories is low, which makes this strategy useful for real-time applications.

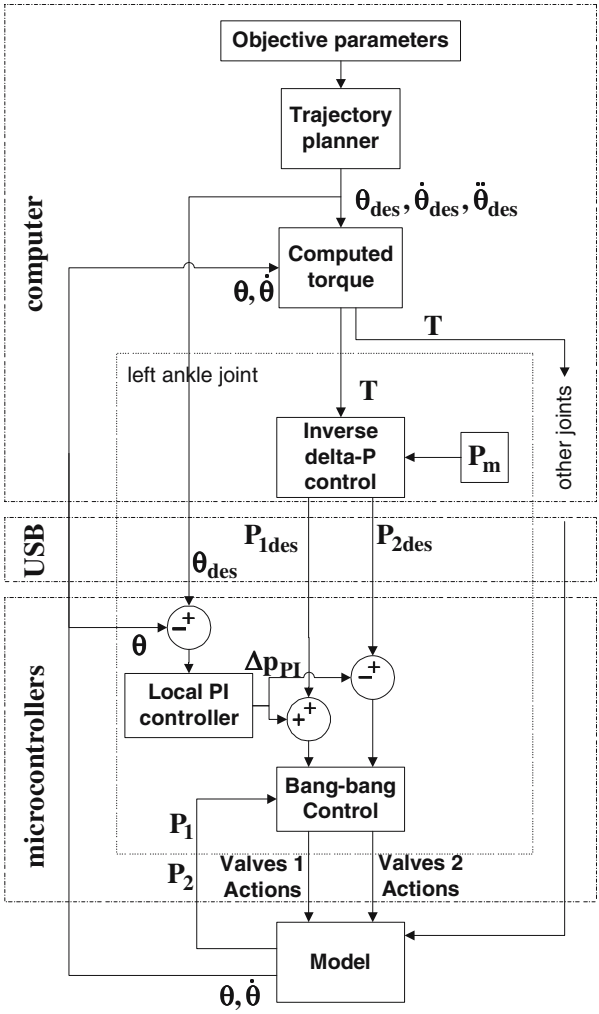


Fig. 3. The applied control architecture

2.2 Complete Low-level Joint Controller

Computed Torque Using the Lagrange equations of the dynamic model the equations of motion can be summarized in the following matrix form (during single support):

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = T$$

Where M is the inertia matrix, which is symmetric and positive definite, C is the centrifugal matrix which contains the centrifugal forces (involving $\dot{\theta}_i^2$) and the coriolis forces (involving $\dot{\theta}_i\dot{\theta}_j$ for $i \neq j$), G is the gravitational force vector. This is the feedforward calculation which is added with a proportional

and derivative feedback part for which the gains are tuned in order for the mechanical system to behave as critically damped.

During the double support phase, immediately after the impact of the swing leg, three geometrical constraints are enforced on the motion of the system. They include the stepheight, steplength and angular position of the foot. Due to these constraints, the robot's number of DOF is reduced to three.

The equations of motion are then written as

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = T + J^T \Lambda$$

where J is the Jacobian matrix and Λ is a column vector of Lagrange multipliers representing the generalized constraint forces. An extra equation is introduced to force the ankle torques to zero as proposed by the trajectory planner. This problem can be solved by dividing the 6 coordinates into a group of independent and dependent coordinates. Using the matrix pseudoinverse as described in [10], the torque vector can than be calculated. This feedforward term is added with a feedback part similar as in single support which gives the computed torque.

Inverse Δp Control For each joint a computed torque is available. The computed torque is then feeded into the inverse delta-p control unit, one for each joint, which calculates the required pressure values to be set in the muscles. The generated torque in an antagonistic setup with two muscles is:

$$\begin{aligned} T = T_1 - T_2 &= p_1 l_1^2 r_1 f_1 - p_2 l_2^2 r_2 f_2 \\ &= p_1 t_1(\theta) - p_2 t_2(\theta) \end{aligned} \tag{1}$$

with p_1 and p_2 the applied gauge pressures in extensor and flexor muscle respectively which have lengths l_1 and l_2. The dimensionless force functions of both muscles are given by f_1 and f_2. The kinematical transformation from forces to torques are represented by r_1 and r_2 which results, together with the muscle force characteristics, in the torque functions t_1 and t_2. These functions are determined by the choices made during the design phase and depend on the joint angle θ. Thus joint position is influenced by weighted differences in gauge pressures of both muscles.

The two desired pressures are generated from a mean pressure value p_m while adding and subtracting a Δp value:

$$p_{1des} = p_m + \Delta p \tag{2}$$

$$p_{2des} = p_m - \Delta p \tag{3}$$

The mean value p_m will determine the joint stiffness and will be controlled in order to influence the natural dynamics of the system. Feeding back the joint angle θ and using expression (1), Δp can be determined by:

$$\Delta p = \frac{T + p_m\left(t_2(\theta) - t_1(\theta)\right)}{t_2(\theta) + t_1(\theta)} \tag{4}$$

The delta-p unit is thus a feed-forward calculation from torque level to pressure level using the kinematic model of the muscle actuation system.

Local PI Controller Because the communicationspeed between PC and the micro-controllers is 2 ms, instabilities occur when the proportional and derivative feedback part of the computed torque are too high. To track the desired trajectory a local PI controller was needed to regulate the error introduced by lowering the feedback gains.

Bang-bang Controller In order to realize a lightweight, rapid and accurate pressure control, fast switching on-off valves are used. The pneumatic solenoid valve 821 2/2 NC made by Matrix weighs only 25 g. The opening time is about 1 ms and it has a flow rate of 180 Std.l/ min. A set of 2 inlet and 4 outlet valves are used per muscle. In the last control block the desired gauge pressures are compared with the measured gauge pressure values after which appropriate valve actions are taken by the bang-bang pressure controller (see Fig. 4).

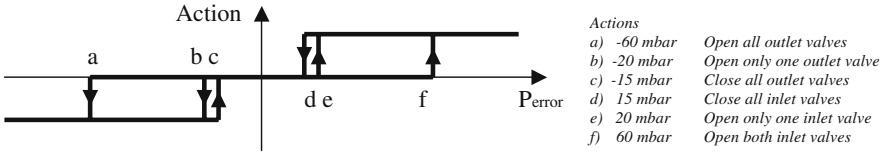


Fig. 4. Multi-level bang-bang control scheme

2.3 Results

The following values for the objective parameters characterize the walking pattern:

$\nu = 0.2\frac{\text{m}}{\text{s}} = 0,72\frac{\text{km}}{\text{h}}$	walking speed	$\delta = 0$ m	stepheight
$\lambda = 0.15$ m	steplength	$\gamma = 0.02$ m	footlift

The walking motion is considered to be a steady walking pattern, consisting of successive single support phases separated by a double support phase. The duration of the double support phase will be chosen as 20% of the total step duration, corresponding to its duration in human walking at low speeds [11]. So the duration of one step becomes 0.74 s.

The model parameter uncertainties are 5% on the mass and COG and 10% on the inertia. The simulations take a time delay of 1 ms for the closing and opening of the valves into account. The sampling time for the calculation of the desired pressures is 2 ms, which is restricted due to the communication between PC and micro-controller. The local PI controller and bang-bang controller, both implemented in the micro-controllers, work with a refresh rate of 0.5 ms.

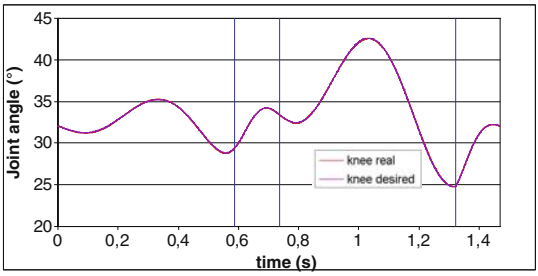


Fig. 5. Knee angle of left leg

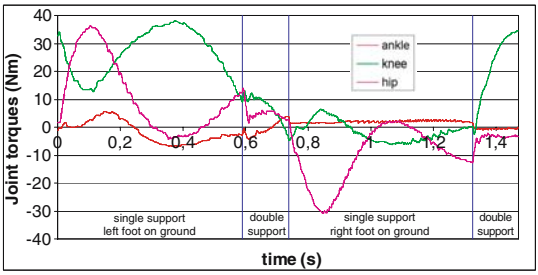


Fig. 6. Torques of left leg

Figures 7 and 8, representing respectively the pressures and valve actions of the front and back muscle of the knee of the left leg, clearly shows the control strategy of keeping the mean pressure constant, which in this case is set at a value of 2 *bar*. Also the valve action due to the bang-bang controller is shown. Note that in these figures a closed valve is represented by a horizontal line at 2 *bar* while a peak upwards represents one or more opened inlet valves, a peak downwards one or more opened exhaust valve. The selection of an appropriate mean pressure value is important regarding energy consumption and control activity. Future work will be the incorporation of this mean pressure value determination in a higher-level control strategy.

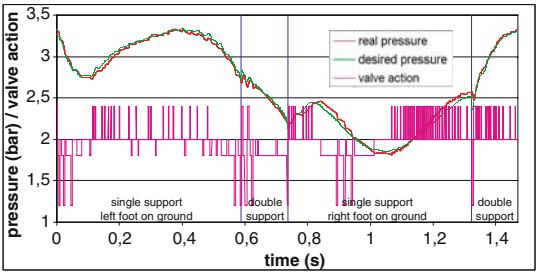


Fig. 7. Pressure and valve action of front knee muscle of left leg

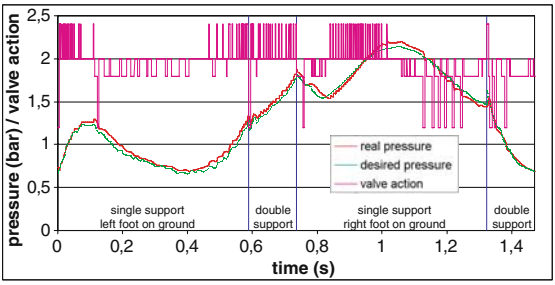


Fig. 8. pressure and valve action of back knee muscle of left leg

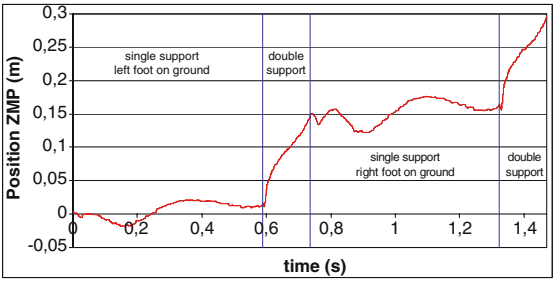


Fig. 9. Zero moment point position

The pressures of the front and rear muscles determines the torques (Fig. 6). Notify the very small ankle torques. The difference between desired and real angle (for example Fig. 5, giving the angle results for the ankle of the supporting leg) never exceeds the 0.1°. For biped locomotion this tracking error is not a problem if the overall stability of the robot is not threatened.

Indication of postural stability is given by the ZMP shown in Fig. 9. One can verify that during the single support phase the ZMP remains close to the ankle joint. During the double support phase the ZMP is transferred from the rear ankle joint to the front ankle joint. This can be seen in Fig. 10 where the weight shift from the rear foot to the front foot is clearly visible.

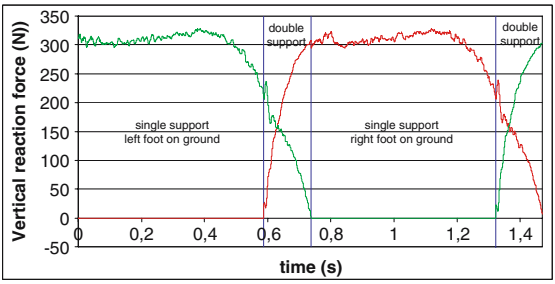


Fig. 10. Vertical ground reaction forces

3 Conclusion

A Pleated Pneumatic Artificial Muscle is very suitable to power a smooth walking bipedal robot. This actuator has a high power to weight ratio and an inherent adaptable passive behaviour. Two antagonistically coupled muscles can be implemented in a straightforward manner to power a rotative joint. The angular position of such a rotative joint depends on the difference in gauge pressures of both muscles and the stiffness of the joint is determined by the sum of pressures. Thus stiffness can be controlled while changing angular position. The biped Lucy is a robot actuated with these muscles.

A future control architecture, based on a global and local control, was discussed and tested in hybrid simulation. A global control is the trajectory planner for dynamically balanced bipeds, the local control can be divided in four parts: a computed torque module, an inverse delta-p unit, a local PI controller and a bang-bang controller.

The simulations showed already promising results. The next step will be the incorporation of this control architecture in the real bipedal robot Lucy.

Acknowledgments

This research has been supported by the Fund for Scientific Research-Flandres (Belgium) and by the Research Council (OZR) of the Vrije Universiteit Brussel (VUB).

References

1. K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka,, "The development of honda homanoid robot," *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp. 1321–1326, 1998.
2. Sony-PressRelease, "World's first running humanoid robot," December 18th 2003.
3. F. Pfeiffer, K. Löffler, and M. Gienger, "Sensor and control aspects of biped robot "Johnnie"," *IEEE International Conference on Humanoid Robots*, Munich, Germany, 2003.
4. K. Yokoi and al., "Humanoid robot's applications in HRP," in IEEE International Conference on Humanoid Robots" *IEEE International Conference on Humanoid Robots*, Karlsruhe, Germany, 2003.
5. F. Daerden and D. Lefeber, "The concept and design of pleated pneumatic artificial muscles," *International Journal of Fluid Power*, vol. 2, no. 3, pp. 41–50, 2001.
6. R. Van Ham,B. Verrelst,B. Vanderborght,F. Daerden and D. Lefeber, "Experimental results on the first movements of the pneumatic biped "Lucy"," *6th International conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, Catania, Italy, pp. 485–492, 2003.

7. ISO6358, "Pneumatic fluid power – method of test – determination of flow rate characteristics of components using compressible fluids," 1989.
8. J. Vermeulen, *Trajectory Generation for Planar Hopping and Walking Robots: An Objective Parameter and Angular Momentum Approach,* PhD Dissertation, Vrije Universiteit Brussel, April 2004.
9. M. Vukobratović and B. Borovac, "Zero-moment point – Thirty years of its life," *Int. Journal of Humanoid Robotics*,vol. 1, pp. 157–173, 2004.
10. C.L. Shih and W. Gruver, "Control of a Biped Robot in the Double-Support Phase," *IEEE Trans. on Systems, Man, and Cybernetics*,vol. 22(4), pp. 729–35, 1992.
11. M. Hardt, K. Kreutz-Delgado, J. Helton and O.V. Stryk, "Obtaining minimum energy biped walking gaits with symbolic models and numerical optimal control," *Workshop – Biomechanics Meets Robotics, Modelling and Simulation of Motion*, HeidelBerg, Germany, pp. 1–19, 1999

Part VII

Passive Walking Locomotion

Stable Walking and Running Robots Without Feedback

Katja D. Mombaur[1], H. Georg Bock[1], Johannes P. Schlöder[1], and Richard W. Longman[2]

[1] IWR, Universität Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany
Katja.Mombaur@IWR.Uni-Heidelberg.de
[2] Dept. of Mechanical Engineering, Columbia University, New York, 10027, USA
RWL4@Columbia.edu

Summary. In this paper we give an overview of our previous work on open-loop stable walking and running robots. The concept of open-loop control implies that all actuators of a robot receive purely periodic pre-calculated torque or force inputs that are never altered by any feedback interference. Passive-dynamic robots can be considered as a special case with zero actuation for which a stabilization is much easier due to the possibility of time shifts. We have developed a numerical method that optimizes the open-loop stability of solutions of periodic optimal control problems with discontinuities. Using this method, we were able to find a number of open-loop stable one- and two-legged robot configurations and motions.

1 Introduction

We investigate the stability of one- and two-legged robots performing periodic running and walking motions, i.e. motions with and without phases of free flight. Stability control of these robots is difficult since they have too few legs and too small feet to maintain static stability during their motions. A real-time closed-loop control of these robots requires sophisticated and expensive sensory systems and feedback-controllers. The computation of appropriate feedback is time critical and often a limitation for making motions faster, hence demanding for high computational capacities on-board.

As a different approach to the stability control issue, we determine, in a first step, what can be achieved without any active feedback, and search instead for purely open-loop controlled, self-stabilizing system configurations and running motions. This implies that there is no feedback and no sensors at all and that the robots response is only based on its natural dynamics. The system input (i.e. the joint torques etc.) is pre-calculated and never modified. It follows that the robot motion must always stay synchronized with the invariable external exciting frequency and that there is no possibility

of time shifts! These purely open-loop stable solutions are considered to be good starting points for an experimental phase in which the robot's motion can then be made more robust and adapted to irregular terrain by simple feedback measures.

The concept of passive-dynamic walking robots is well known. They are purely mechanical devices moving down inclined slopes which, besides lacking feedback, have no actuators at all and are only powered by gravity (see [1] for an overview). We consider them as a special case of open-loop stable robots with zero actuation. Passive-dynamic systems are described by autonomous differential equations such that perturbations of the original periodic trajectory are allowed to cause orbital shifts which nevertheless leads to orbitally asymptotically stable motions.

There only exist a few entirely open-loop stable actuated robots today [2], [3], and they rely mainly on simple stabilization measures like a large circular foot (i.e. the classical tinkertoy effect). There are, however, more complex robots that combine an exploitation of self-stabilizing effects and feedback control [4, 5].

The goal of our research was to supply a tool that can determine, for a given robot topology, those model parameters (i.e. masses, lengths, etc.), actuator inputs, initial values and cycle times, that characterize open-loop stable motions. We have developed a numerical stability optimization method that can perform this task for very general robots given as multibody system models for which an intuitive stabilization is not possible. We have used this method to determine open-loop stable motions for a wide range of one- and two-legged robots. The purpose of this paper is to give an overview of these results.

2 Mechanical Models of Walking and Running

Models of periodic gaits involve multiple phases, each with its own set of governing equations, and we assume that the order but not the duration of phases is known. The number of degrees of freedom as well as the number of free control variables $u(t)$ (i.e. actuator inputs in physical terms) can be different in each phase. Each phase is described by a set of highly nonlinear differential equations. These can be ordinary differential equations (ODEs)

$$\dot{q}(t) = v(t) \tag{1}$$

$$\dot{v}(t) = a(t) = M^{-1}(q(t), p) \cdot f(q(t), v(t), u(t), p) \tag{2}$$

where t is the physical time, q are the position coordinates and v and a the corresponding velocities and accelerations, and p the time-constant model parameters. The use of redundant coordinates leads to the alternative formulation with differential algebraic equations (DAEs)

$$\dot{q}(t) = v(t) \tag{3}$$

$$\dot{v}(t) = a(t) \tag{4}$$

$$\begin{pmatrix} M(q(t),p) & G^T(q(t),p) \\ G(q(t),p) & 0 \end{pmatrix} \begin{pmatrix} a \\ \lambda \end{pmatrix} = \begin{pmatrix} f(q(t),v(t),u(t),p) \\ \gamma(q(t),v(t),p) \end{pmatrix} \tag{5}$$

$$g_{pos} = g(q(t),p) = 0 \tag{6}$$

$$g_{vel} = G(q(t),p) \cdot \dot{q}(t) = 0\ . \tag{7}$$

In the case of passive dynamic robots, the dimension of u is zero. Phase boundaries τ_j (and thus also phase durations) are implicitly defined by roots of so called switching functions

$$s_j(t,q(t),v(t),u(t),p) = 0\ . \tag{8}$$

At the phase boundaries τ_j, there may be a discontinuity between the right hand side force terms f_j and f_{j+1} of the two adjacent phase models

$$f_{j+1}(x(\tau_j^+,u(\tau_j^+),p)) \neq f_j(x(\tau_j^-,u(\tau_j^-),p))\ . \tag{9}$$

$x^T = (q^T, v^T)$ are the state variables of the problem. Furthermore, discontinuities of the state variables may occur at these points:

$$x(\tau_j^+) \neq x(\tau_j^-) \text{ with } x(\tau_j^+) = h(x(\tau_j^-),p)\ . \tag{10}$$

Periodicity constraints of the form $q_{(red)}(T) = q_{(red)}(0)$, $v(T) = v(0)$ are applied to the full gait model cycle T. Position variables associated with the direction of travel are excluded from the periodicity constraints.

The model cycle considered is typically one step (which is a full physical cycle for a monopod and just half a physical cycle for a biped) plus possibly a leg shift to restore periodicity. Thus we ensure the generation of symmetric gaits with equal right and left steps.

A number of pointwise or continuous inequality constraints are imposed on the model in order to produce the desired gait pattern, e.g.

- forced clearance of swing foot
- ranges for positions and velocities of certain points
- minimum and maximum phase durations
- relation between positions or velocities at different points in the cycle.

3 Numerical Optimization of Open-loop Stability

We have developed and implemented a numerical method for the optimization of open-loop stability of a periodic system. A detailed description of the method can be found in [6] and [7]. The method relies on a two-level approach splitting the problems of periodic gait generation and stabilization of the periodic system.

3.1 Outer Loop: Stabilization of a Periodic Gait

In the outer loop of the optimization procedure, model parameters are determined according to stability aspects. Stability is defined in terms of the spectral radius of the Jacobian C of the Poincaré map associated with the periodic solution. It can be proven that this well-known criterion for smooth systems [8] can also be used to determine the stability of solutions of a nonlinear multiphase system with discontinuities [7, 9]. If the spectral radius is smaller than one, the solution is asymptotically stable, and if it is larger than one, the solution is unstable. In case of non-periodic variables, a projection of the monodromy matrix to the subspace of the periodic variables has to be used instead. In the case of passive-dynamic systems, the invariable eigenvalue of one associated with orbital perturbations needs to be eliminated.

We use the spectral radius of C or the relevant projection as objective function of our optimization

$$\min_p |\lambda_{max}(C(p))| \,, \tag{11}$$

in the intention to decrease it below one. This is a difficult optimization criterion for two different reasons: The maximum eigenvalue function of a non-symmetric matrix is non-differentiable and possibly even non-Lipschitz at points where multiple eigenvalues coalesce. Additionally, the determination of the matrix C involves the computation of first order sensitivities of the discontinuous trajectories. Any gradient-based optimization method would thus require second order derivatives of the trajectory which are extremely hard to compute, especially due to the discontinuities in the dynamics. For all reasons mentioned, a direct search method (i.e. a method that solely uses function information and does neither compute nor explicitly approximate derivatives) has proven to be a very good choice for the solution of this problem. We have implemented a modification of the classical Nelder-Mead algorithm [10].

3.2 Inner Loop: Generation of Periodic Gaits

The task of the inner loop is to find – for the set of parameters prescribed by the outer loop – actuator patterns, initial values and cycle time leading to a periodic trajectory. For the outer loop optimization it is essential that the solution of this inner loop problem is unique. We therefore do not solve a boundary value problem, but an optimal control problem with the additional objective of minimal actuator input:

$$\min_{x,u,T} \quad \int_0^T ||u||_2^2 dt \tag{12}$$

$$\text{s.t.} \qquad \text{multiphase ODE / DAE-model}$$

$$x(\tau_j^+) \quad = \quad h(x(\tau_j^-), p) \quad \text{for} \quad j = 1, \ldots, n_{ph} \tag{13}$$

$$g_j(t, x(t), u(t), p) \quad \geq \quad 0 \qquad \text{for} \quad t \in [\tau_{j-1}, \tau_j] \tag{14}$$

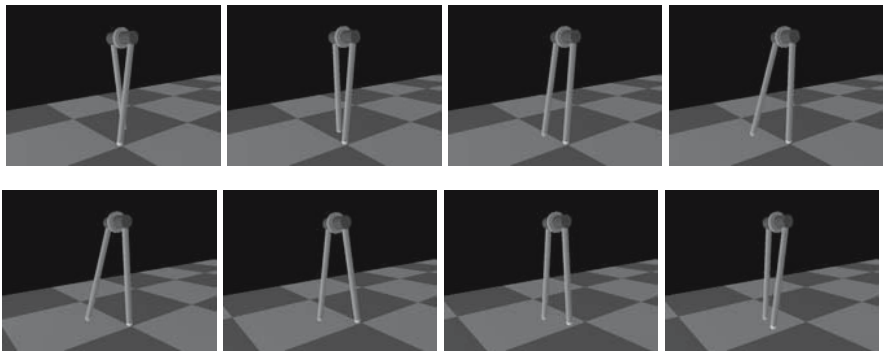
$$r_{eq/ineq}(x(0), \ldots, x(T), p) = / \geq 0 \tag{15}$$

For passive-dynamic systems a different objective function is used (e.g. minimal cycle time). We solve this problem using a variant of the optimal control code MUSCOD (Bock & Plitt [11], Leineweber [12]) suited for periodic gait problems. It is based on a direct method for the optimal control problem discretization and a multiple shooting state parameterization which transforms the original boundary value problem into a set of initial value problems with corresponding continuity and boundary conditions. The resulting structured non-linear programming problem is solved by an efficient tailored SQP algorithm.

4 Numerical Results: Open-Loop Stable Robots

In this section we present periodic solutions for several one- and two-legged robots, all of which are open-loop stable and have been found using the stability optimization method described above.

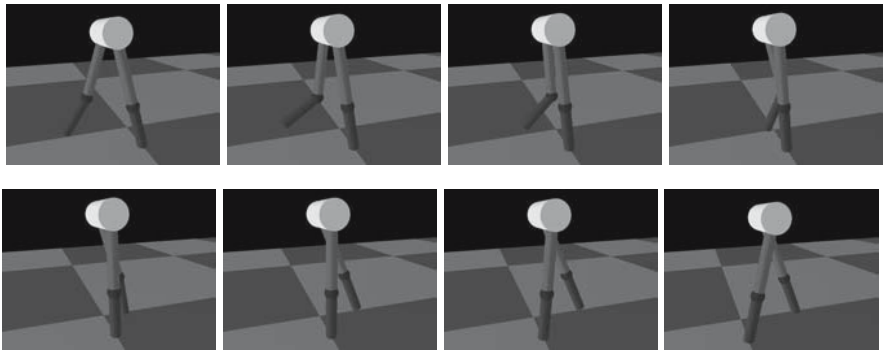
4.1 3D Passive-Dynamic Walking Robot with Point Feet



The first stable robot example is a passive-dynamic three-dimensional walker with two straight legs connected by a hinge and no torso, that moves on an inclined slope without any actuator help. Always only one foot is in non-sliding contact with the ground. The foot switching is assumed to be

instantaneous and collisional such that there is no double support phase. This model has its origin in the physical robot with toroidal feet built by Coleman (see [13] and [14]). In simulation, we have since then treated the model versions with disk feet [15] and with point and toroidal feet [6]. Here we present the most stable point foot solution. The system has got four DOFs. We use the set of angle coordinates $q^T = (\phi, \psi, \theta_{st}, \theta_{sw})$ which, speaking in aeronautical terms, are the robot's heading and rolling angle, the pitch angle of the stance leg, and the relative pitch angle of the swing leg. The model parameters of this robot are the six inertia parameters d_1, d_2, d_3, ρ, β, γ (see [13] for an explanation of this parameterization), the leg mass m, the slope angle α, the total leg length l, the leg center of mass location d_x, d_y, d_z in local leg coordinates and the hip spacing w. In the case of disk (or toroidal) feet there would be one (or two) additional parameters related to foot radii r_1 (and r_2). As objective function in the inner loop, we use a minimization of cycle time. The most stable solution for the Tinkertoy robot with point feet has a monodromy matrix with spectral radius 0.7958. The model parameters of this solution are $d_1 = 0.0252$, $d_2 = 0.3879$, $d_3 = 0.2858$, $\rho = -0.009$, $\beta = -0.2323$, $\gamma = -0.0294$, $m = 1.0$, $\alpha = 0.0757$, $d_x = 0.0024$, $d_y = 0.7901$, $d_z = 0.4323$, $l = 1.0$, $w = 0.3165$. The corresponding trajectory has initial values of $x_0^T = (0.0876, -0.0114, -0.1731, 3.465, -0.0885, -0.0256, 0.3952, -0.3248)$ and a cycle time of $T = 1.2314s$.

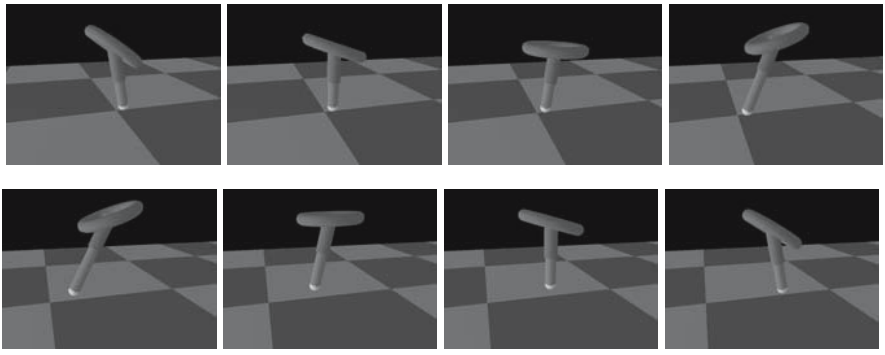
4.2 Actuated Biped Walking Robot with Knees



As the first actuated example of open-loop stable walking, we present a two-legged robot with knees and point feet and without a torso. The robot can be considered as a simplified model of human walking in the saggital plane. The robot consists of four bodies, two identical legs with a thigh and a shank each and has point feet. It is actuated by torques at hip and knees. The model parameters p are masses (m_T, m_S), lengths (l_T, l_S), and relative center of mass locations (c_T, w_S, c_S) of thigh T and shank S. The observed cycle – one step

– starts and ends right after heelstrike and involves two different phases with state discontinuities at the end of each phase. The stance leg is assumed to be straight all the time, whereas the swing leg is bent in the first phase and straight in the second phase after kneestrike, such that the robot has got three or two degrees of freedom, respectively. We use for all phases the uniform set of coordinates $q = (\phi_1, \phi_2, \phi_3)^T$ which are the angles of stance leg and of swing leg thigh and shank. The most stable solution for the actuated kneed walking robot has a spectral radius of 0.5667. The initial values of this most stable periodic trajectory are $x_0^T = (-0.3, -0.3, 0.3, 0.8385, -4.5056, -1.5686)$ and its period is $T = 0.7832$ with phase times $T_1 = 0.4905\,\mathrm{s}$ and $T_2 = 0.2926\,\mathrm{s}$. Its step length is $\Delta_{step} = 0.127\,\mathrm{m}$. The solution was obtained for a robot configuration with parameters values (in ISO units) $l_T = 0.2627$, $l_S = 0.1685$, $m_T = 1.2759$, $m_S = 0.6575$, $w_S = 0.0402$, $c_T = 0.0111$, and $c_S = 0.2259$. For more details on this robot model, compare [16] or [6].

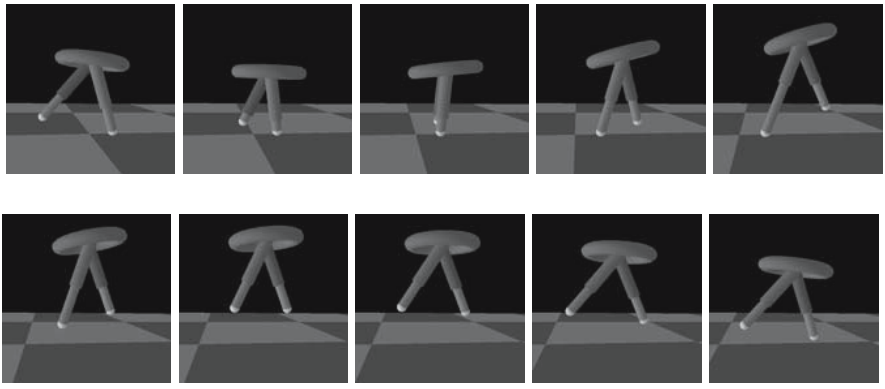
4.3 Actuated One-Legged Hopping Robot



Next, we are presenting an actuated one-legged hopper moving in the vertical plane. With its single leg, a small foot and a relatively high center of mass, the robot has obviously no statically stable standing position. However, it is capable of performing open-loop stable two-dimensional hopping motions including a non-sliding contact phase and a flight phase. The toroidal trunk and the leg are coupled by an actuated hinge. The two parts of the telescopic leg are connected by an actuated spring-damper element. We have studied circular as well as point shaped feet but we only present the point foot solution here. Model parameters of the robot are trunk mass m_b and inertia Θ_b, leg mass m_l and inertia Θ_l, distance between centers of mass of trunk and leg d, leg rest length l_0, torsional spring and damper constants k_{tors} and b_{tors}, rest location of torsional spring δ_ϕ, and translational spring and damper constants k and b. The foot is assumed to be massless. The leg length l is assumed to be a free variable only during the contact phase and is driven back to rest

position soon after lift-off and fixed for the flight phase. During the flight phase, the robot therefore has four degrees of freedom, and three during contact phase (2 DOF lost due to point contact, and 1 additional DOF due to free leg length). As state variables we choose the uniform set of coordinates $q = (x_b, y_b, \phi_b, \phi_l)^T$ and the corresponding velocities, where x_b and y_b are two-dimensional position coordinates of the trunk center of mass, and ϕ_b and ϕ_l are the orientations of trunk and leg. Using eigenvalue optimization in the outer loop and objective (12) in the inner loop, we were able to bring the spectral radius down as far as 0.1292. The model parameters of this solution are (in ISO units) $m_b = 2.0$, $\Theta_b = 0.3503$, $m_l = 0.5033$, $\Theta_l = 0.2391$, $d = 0.3663$, $l_0 = 0.5$, $r = 0$, $k_{tors} = 25.902$, $\delta_\phi = 0.2$, $b_{tors} = 3.457$, $k = 589.1$, and $b = 61.79$. The cycle time of this solution is $T = 0.471\,\mathrm{s}$ with phase times $T_{contact} = 0.305\,\mathrm{s}$ and $T_{flight} = 0.166\,\mathrm{s}$, and the initial values are $x^T = (0, 0.490, -0.1447, 0.20, 0.3326, 0.0011, -2.8399, -0.6524)$. For more extensive results on this robot and details about the mathematical model, we refer to [17].

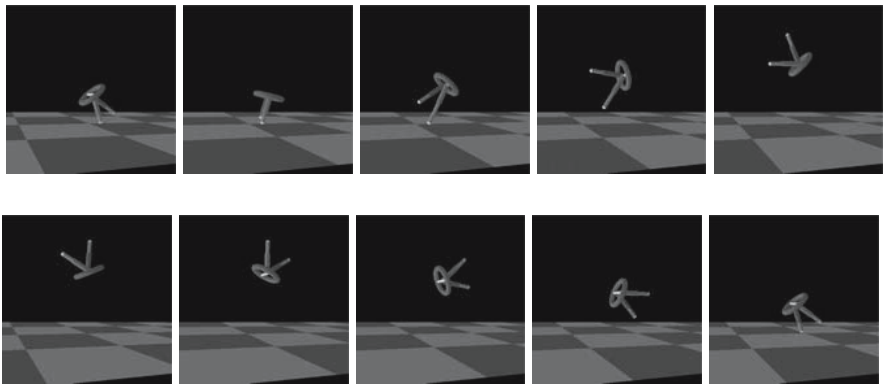
4.4 Actuated Biped Running Robot



The running biped is an extension of the monopod described above with a second identical leg. It is capable of stable two-dimensional running motions with alternating flight and non-sliding single-foot contact phases without any feedback support. The biped has the same model parameters as the monopod, but along with the leg also the actuators are duplicated. It has one degree of freedom more which results in five DOF during flight phase and four DOF during contact phase with the telescopic leg being unlocked. As state variables we choose the uniform set of coordinates $q = (x_b, y_b, \phi_b, \phi_{l,1}, \phi_{l,2})^T$ and the corresponding velocities. The most stable solution has a spectral radius of 0.8168. The model parameter values of this solution are $m_b = 2.0$, $\Theta_b = 0.3465$, $m_l = 0.2622$, $\Theta_l = 0.182$, $d = 0.11$, $l_0 = 0.5$, $k_{tors} = 11.08$, $\Delta\phi_l = 0.5$, $b_{tors} = 9.989$,

$k = 606.8$, and $b = 42.48$. The corresponding cycle time is $T = 0.5476\,\mathrm{s}$ for one step with $T_{contact} = 0.2533\,\mathrm{s}$ and $T_{flight} = 0.2943\,\mathrm{s}$ and the initial values $x_0^T = (0.0, 0.4777, -0.1, 0.3, -0.7, 1.240, -2.490, -0.3941, -0.8908, 2.032)$. For more information, see [17].

4.5 Actuated Biped Performing Somersaults



The same biped model as in 4.4 can perform another, very unusual type of forward motion: repetitive forward somersaults with alternating single leg contact phases. Even this motion can be made stable without any feedback! The spectral radius of the most stable solution we found is 0.7324. It is associated with the parameter values $m_b = 2.0$, $\Theta_b = 0.3997$, $m_l = 0.25$, $\Theta_l = 0.1973$, $d = 0.1$, $l_0 = 0.5$, $k_{tors} = 17.066$, $\Delta\phi = 0.655$, $b_{tors} = 12.74$, $k = 699.9$, and $b = 20.01$. and the initial values $x_0^T = (0.0, 0.468, 0.8, 0.359, 1.056, 3.093, -3.471, -8.1, -3.351, -9.838)$. The period is $T = 0.784\,\mathrm{s}$ with phase durations $T_1 = 0.20\,\mathrm{s}$ and $T_2 = 0.584\,\mathrm{s}$ for contact and flight phase, respectively. Its step length is $1.799m$. The result is presented in more detail in [18].

5 Conclusion

We have presented several simulated monopods and bipeds that are capable of stable walking and running motions without any feedback. Finding these solutions was only possible using a newly developed numerical stability optimization method. In the design and control of real walking robots these solutions can be used as a sound basis on top of which feedback control is applied. Furthermore, the results of this work encourage one to look for open-loop stable features in the gaits of more complex systems, like humans or animals.

Acknowledgement

The first author would like to acknowledge financial support by the state of Baden-Württemberg through a Margarete von Wrangell scholarship.

References

1. T. McGeer. Passive dynamic biped catalogue. In R. Chatila and G. Hirzinger, editors, *Proceedings of the 2nd International Symposium of Experimental Robotics, Toulouse*. Springer-Verlag, New York, 1991.
2. R. P. Ringrose. Self-stabilizing running. Technical report, Massachusetts Institute of Technology, 1997.
3. T. E. Wei, G. M. Nelson, R. D. Quinn, H. Verma, and S. L. Garverick. Design of a 5-cm monopod hopping robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2828–2823, 2000. http://www.fluggart.cwru.edu/icra2000/icra2000.html.
4. M. Buehler. Dynamic locomotion with one, four and six-legged robots. *Journal of the Robotics Society of Japan*, 20(3):15–20, April 2002.
5. J. E. Pratt. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. PhD thesis, Massachusetts Institute of Technology, 2000.
6. K. D. Mombaur. *Stability Optimization of Open-loop Controlled Walking Robots*. PhD thesis, University of Heidelberg, 2001. www.ub.uni-heidelberg.de/archiv/1796. VDI-Fortschrittbericht, Reihe 8, No. 922, ISBN 3-18-392208-8.
7. K. D. Mombaur, H. G. Bock, J. P. Schlöder, and R. W. Longman. Open-loop stable solution of periodic optimal control problems in robotics. *submitted to Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM)*, 2003.
8. J. C. Hsu and A. U. Meyer. *Modern Control Principles and Applications*. McGraw-Hill, 1968.
9. J. Grizzle, G. Abba, and F. Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *IEEE Transactions on Automatic Control*, 46(1):51–64, 2001.
10. J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.
11. H. G. Bock and K.-J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress, Budapest*, pp. 242–247. International Federation of Automatic Control, 1984.
12. D. B. Leineweber. *Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models*. PhD thesis, University of Heidelberg, 1999. VDI-Fortschrittbericht, Reihe 3, No. 613.
13. M. J. Coleman. *A stability study of a three-dimensional passive-dynamic model of human gait*. PhD thesis, Cornell University, February 1998.
14. M. J. Coleman and A. Ruina. An uncontrolled walking toy that cannot stand still. *Physical Review Letters*, 80(16):3658–3661, April 1998.
15. M. J. Coleman, M. Garcia, K. D. Mombaur, and A. Ruina. Prediction of stable walking for a toy that cannot stand. *Physical Review E*, 2, 2001.

16. K. D. Mombaur, H. G. Bock, J. P. Schlöder, and R. W. Longman. Human-like actuated walking that is asymptotically stable without feedback. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4128–4133, Seoul, Korea, May 2001.
17. K. D. Mombaur, R. W. Longman, H. G. Bock, and J. P. Schlöder. Open-loop stable running. *Robotica*, to appear 2004.
18. K. D. Mombaur, H. G. Bock, J. P. Schlöder, and R. W. Longman. Self-stabilizing somersaults. *submitted*, 2004.

From Passive to Active Dynamic 3D Bipedal Walking – An Evolutionary Approach

Steffen Wischmann and Frank Pasemann

Fraunhofer Institute for Autonomous Intelligent Systems (AiS),Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{steffen.wischmann, frank.pasemann}@ais.fraunhofer.de

Summary. Applying an evolutionary algorithm, we first develop the morphology of a simulated passive dynamic bipedal walking device, able to walk down a shallow slope. Using the resulting morphology and adding minimal motor and sensory equipment, a neural controller is evolved, enabling the walking device to walk on a flat surface with minimal energy consumption. The applied evolutionary algorithm fixes neither the size nor the structure of the artificial neural network. Especially, it is able to generate recurrent networks, small enough to be analyzed with respect to their behavior relevant inner dynamics. An example of such a controller is given which realizes also minimal energy consumption.

Key words: passive dynamic walking, evolutionary robotics, recurrent neural networks, bipedal walking

1 Introduction

A large amount of research activities has been spend on problems concerning the development and control of bipedal robots. Most of them mainly headed to mimic human motions. The most famous example is Honda's Asimo [4]. It is also a representative example for not attaching importance on efficient bipedal walking, concerning the energy consumption, as it can be found for example in human walking. Taking efficiency into account principles of bipedal walking in biological systems had to be understood first [7].

The efficency of human walking is mainly based on two aspects. On one hand, there is a very well coordinated use of muscles [1]. There is still a lack of new technology for efficient actuators to imitate such a motor system. On the other hand, the morphology is very well adapted to this kind of locomotion [11]. Considering the latter point seems to be very promising [3, 6]. A bipedal walking machine is shown, that is able to walk down a shallow slope without any energy consumption only powered by gravity (passive dynamic walking). As it is described in [6] the gravity force compensates the energy loss occurring due to the heel strike in bipedal walking. Therefore it should be

possible to create a motor system which just compensates this energy loss and therefore enables the machine to walk on level surface with minimal energy consumption. To reach this goal, and to avoid tremendous mathematical calculations, for determining an efficient morphology we use an evolutionary algorithm.

In considering the strong interrelationship of morphology and control [2, 5, 8] we first evolve the morphology for a passive dynamic walking device, and secondly we add a minimal sensorimotor system to that model and evolve a recurrent neural network (RNN) as control structure. Finally the behavior relevance of the controller dynamics is analyzed.

2 Methods

2.1 Evolutionary Algorithm

For evolution the ENS^3 (evolution of neural systems by stochastic synthesis) algorithm [9] was used. A simple form of this algorithm is as follows.

1. Create an initial population $P(0)$.
2. Create an offspring population $\hat{P}(t)$ by reproducing the parent population $P(t)$.
3. Vary the structure and the parameters of $\hat{P}(t)$ by stochastic operators.
4. Evaluate each individual of $\hat{P}(t)$ and $P(t)$ and assign a fitness value to it.
5. Create $P(t+1)$ by a rank based selection of individuals out of $\hat{P}(t)$ and $P(t)$ according to their fitness values.
6. Has some stop criteria fulfilled? If yes, exit, if no, go to 2.

A population either consists of a number of neural controllers (control optimization) or a number of parameter sets (morphology optimization). In the first case the evolutionary algorithm fixes neither the size nor the structure of the artificial neural network. Parameters of the evolutionary process such as the probability of including, deleting or changing synapses and neurons, average size of a population, the steepness of the selection function, costs of neurons and synapses etc. can be modified online.

The neurons of the controller are of the additive graded type with zero bias terms. The dynamics of the controller is then given by

$$a_i(t+1) = \sum_{j=1}^{j} w_{ij}\ f(a_j(t))\ , \tag{1}$$

where the activation a_i of neuron i at time $t+1$ is the weighted sum of the outputs $f(a_j(t))$ of the connected neurons at time t, and w_{ij} denotes the strength of the connection from neuron j to neuron i. The input neurons act as buffers with identity as transfer function. The transfer function for all hidden and output neurons is given by the hyperbolic tangent $f(x) = tanh(x)$.

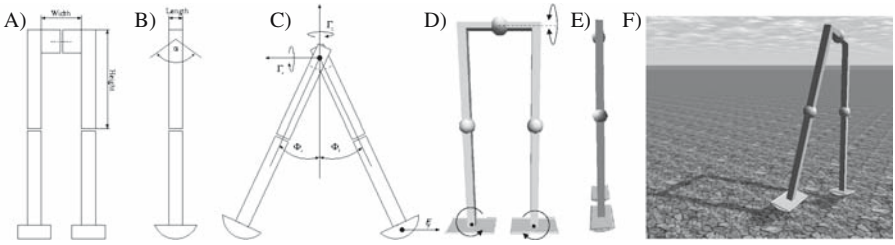


Fig. 1. A-C: Evolvable parameters of the passive dynamic walking device. **D-F**: The active dynamic walking device within the simulation environment

2.2 Evolution of Body and Brain

Hence we evolved the morphology and control in a sequential order, the parameters of the passive dynamic walking device has to be introduced first. But as far as the main concern of this paper is the use of the principles of passive dynamic walking for an active controlled bipedal walking machine, for information we want to mention that 10 morphology parameters (dimensions and mass distributions) and 5 initial conditions which were evolved (Fig. 1a–1c). A detailed description of the experimental setup and the results in pure passive dynamic walking can be found in [10]. The used model is sketched in Fig. 1.

The fitness value of each individual i was computed as follows.

$$F_{V_i} = (S_i + 1) \cdot W_i \ , \tag{2}$$

where S_i denotes the number of completed steps, and W_i the distance covered by the center of the hip. One is added to the number of steps to exclude a fitness value of zero if the walking device isn't able to make at least one step. For evaluation the model is implemented within the physical simulation environment *Open Dynamics Engine*[a] (ODE). The evaluation process was stopped when the walking device was falling over.

Once a suitable passive dynamic walking device was evolved, we applied a sensorimotor system to this model which should be as minimal as possible. Therefore, the model has just three motors. A servo motor model in each ankle is able to shorten or lengthen the leg respectively. The motor at the hip joint is controlled by torque. If no torque is applied to this motor the leg can swing freely around the joint axis. The rotation axis of all motors are sketched in Fig. 1d. As it can be seen the hip motor is the only one which is able produce forward motion.

The output of the control architecture is represented by the desired angles of the foot motors and the desired torque of the hip motor. The actual angles of these three joints are fed into the controller as internal sensors. Additionally

[a] ODE is an open source library with an C/C++ API for simulating rigid body dynamics. More information about ODE can be found at http://www.ode.org

a foot contact sensor is given as an external sensor, which provides information about which foot is in contact with the ground. This sensor can have three discrete states. The output of the according input neuron is +1, if the left foot is in contact with the ground, and −1 vice versa. If both feet are in contact with the ground, the signal is 0. Because we wanted to encourage walking, not running, the case that no feet is in contact is excluded. If no foot is contact with the ground the signal maintains its last value.

For evaluation the following fitness function was used.

$$F_V = k_1 \cdot (S_i + 1) \cdot W_i - k_2 \cdot \Delta E \tag{3}$$

It is an extended version of Formula (2) with an additional energy term ΔE to reduce the energy consumption. The energy term is given as follows:

$$\Delta E = \sum_{t=1}^{n} \sqrt{(\Delta T_H(t))^2 + (\Delta A_{Fr}(t))^2 + (\Delta A_{Fl}(t))^2}\,, \tag{4}$$

where $\Delta T_H(t) = T_H(t) - T_H(t-1)$ denotes the torque difference of the hip motor between two time steps and $\Delta A_F(t) = A_F(t) - A_F(t-1)$ denotes the difference of the desired angle of the foot motor of consecutive time steps and n denotes the maximum time steps. The two weights k_1 and k_2 can be varied online during the evolutionary process. The sum of these two terms was always 1 to normalize the fitness value. At the beginning of the evolution $k_1 \gg k_2$ was chosen to encourage the development of a gait pattern at all. Since a suitable gait pattern was obtained we set $k_1 = k_2$ to encourage control structures which additionally produce an energy efficient walking behavior. The neural network was updated with an frequency of 10 Hz, whereas the simulation was updated with 100 Hz.

The active dynamic walking device had to fulfill two tasks. First it has to produce a stable walking pattern on flat surface with minimal energy consumption, and second, it had to initialize the gait by itself. For the latter case all evolved starting conditions of the passive dynamic walking device were set to zero.

3 Results

The model taken for the active dynamic walking device was able to make about 9 steps and cover a mean distance about 3.4 meters on a descending slope of 4 degrees. This achievement seemed good enough to continue with the evolution of a neural network for an active dynamic walking device.

Therefore we applied the previously described sensorimotor system to the passive dynamic walking device. At the end of the evolutionary process the *RNN* displayed in Fig. 3a turned out to be the best result.

The walking device with this control structure was able to initialize a walking pattern and to maintain stable walking. It covers a mean distance

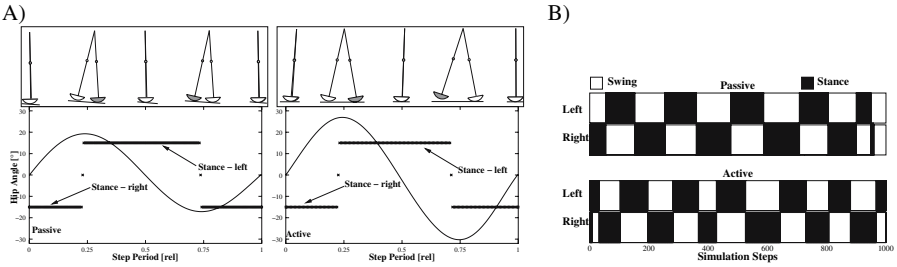


Fig. 2. Passive vs. active dynamic walking. (**A**): One step cycle. (**B**): Swing and stance phases

about 7.1 meters by making mean 12 steps. So we got an improved result in comparison to pure passive dynamic walking.

The small structure of the *RNN* has yet another advantage as we will see in the next section by analyzing the dynamics of the control architecture.

4 Analysis

In Fig. 2b the swing and stance phases of both devices are plotted. In the passive walking device a nearly regular pattern can be found. Here only the morphology and the environment (slope, gravity, friction etc.) defines the walking pattern. By way of contrast in the active walking device the pattern is additionally controlled by the *RNN*, thus it is less regular in comparison to the passive dynamic walking device.

A whole step cycle for both walking devices is shown in Fig. 2a. The pattern of the hip angle, the swing and stance phases and their switching points are very similar except of the amplitude of the hip angle. The reason for the higher amplitude is a little offset in the hip motor signal as it will be described later.

We will now see how the output activity of the sensor inputs and motor outputs (Fig. 3b and 3c) are related to the internal dynamics of the *RNN* (Fig. 3a). Therefore we first discuss the behavior of the foot motors on the basis of the left foot motor neuron N_6.

There are two strong excitatory connections w_{61} and w_{64} (describing the synapsis from N_1 to N_6 and accordingly from N_4 to N_6). As it can be seen in Fig. 3b the absolute output of N_1 is always lower than that of N_4 due to the fact that N_4, representing the foot contact sensor, can only have 3 states. So the influence of w_{64} is dominant. Whenever the left leg is in stance, the output of N_4 is $+1$ and so, because of the strong connection w_{64} and the strong excitatory self-connection w_{66} of N_6, the output of N_6 is also $+1$. If the left leg is in the swing phase, the output of N_4 is -1 and at the same time the output of N_6 is also -1. So, due to the strong self-connection of N_6 the output of this neuron switches between two states. These states correspond

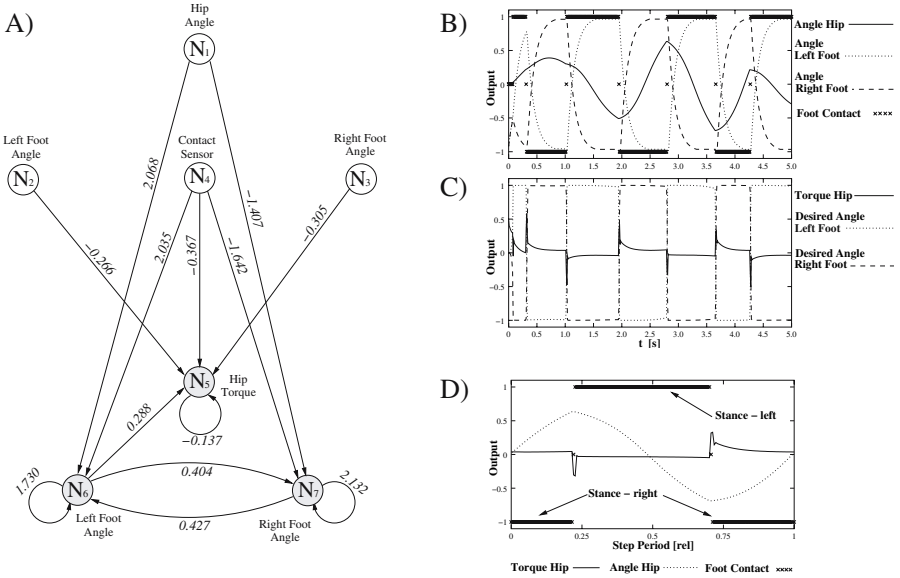


Fig. 3. (**A**): The *RNN* of the active dynamic walking device. (**B**): Output of the input neurons ($N_1 - N_4$). (**C**): Output of the output neurons ($N_5 - N_7$) (**D**): Hip motor output for one step cycle

to motor positions of 15 degrees or 0 degrees, meaning the leg is lengthen, if it is in stance and shorten if it is in the swing phase. The reaction of the servo motor in the left foot joint can be seen in the curve of the actual angle sensor (N_2) in Fig. 3b.

For the motor neuron N_7, controlling the right foot joint, we observe the same behavior. In contrast to N_6 there are two strong inhibitory connections w_{71} and w_{74}. The resulting behavior is the same for N_6, because here the output of N_4 is -1 whenever the right leg is in stance and $+1$ when it is the swing leg.

We will now discuss the behavior of the hip motor. Whenever the left leg is in the swing phase, we need positive torque at the hip motor to support forward swinging, and consequently we need negative torque if the right leg is in the swing phase. As it is shown in Fig. 3d, we have three characteristics for the produced hip torque (N_5). There are peaks always at the moment of changing the swing and stance leg (i), and a damped transition (ii) to a specific offset (iii). This is controlled by five connections including one inhibitory self-connection. The damped transition is the result of the negative self-connection w_{55}.

Additionally, there are two inhibitory connections w_{52} and w_{53} from the two foot angle sensor neurons. If we look at the outputs of the neurons N_2 and N_3 (Fig. 3b) we see that most of the time both have the same value with

different signs. Notice that there is a little difference of 0.039 between the weights w_{52} and w_{53}. We will later come back to this point.

Further on we see a inhibitory connection w_{54} from the foot contact sensor and a excitatory w_{56} from the motor output neuron of the left foot. We remember both outputs of the neurons are identical for most of the time, either -1 or $+1$. Notice the little difference of 0.079 between these weights.

Assume that the left leg is in stance. The output of N_2 is positive and of N_3 negative with the same absolute value. So we have a little excitatory influence because of the difference between the two weights w_{52} and w_{53}. The output of N_4 and N_6 is $+1$. So here we have a little inhibitory influence from these two connections because of the difference between w_{54} and w_{56} (compare with Fig. 3a). The difference of the described differences of the weights, in this case -0.04, is exactly the offset which can be observed in the output activity of N_5. Therefore, there is a little negative torque which supplies the swinging right leg. Correspondingly, if the left leg is in the swing phase we have an supporting offset output of $+0.04$.

Let us now consider the peaks of the produced hip torque. When both feet are in contact with the ground the inhibitory influence of w_{54} from the contact sensor is gone because the output of N_4 is zero. But the output of N_7 is either -1 or $+1$ due to the strong self-connection of N_7. So the balance between w_{54} and w_{56} is disturbed and a peak in the output of N_5 appears due to the excitatory connection w_{56}. The sign depends on which leg is in stance, as discussed before. Thus, the peak in the hip motor signal appears only at the switch between swing and stance leg when both feet are in contact with the ground. It is illustrated more clearly in Fig. 3c, where the produced output for the hip torque, the hip angle and the swing and stance phases are described for one step cycle. We can see the peak and the smooth transition to the offset, to which leg is in stance, respectively.

To summarize, we can say, that the walking pattern of the active dynamic walking device is mainly generated by the rigid body dynamics of the evolved morphology. Only at the moment of energy loss, at the heel strike, the *RNN* produce the required compensating hip torque. Only the motor at the hip joint is able to produce forward motion and so the energy consumption is minimal. We saw that the contact sensor is essential for controlling the foot motors. And it is also very essential for producing an appropriate torque at the hip joint.

5 Conclusions

We have presented a method to develop efficient control structures for generating a simple gait pattern in bipedal walking. We showed how important the optimization of the morphology is, how an evolved minimal sensorimotor system looks like, and how to interpret the dynamics of the evolved *RNN* as that of an efficient control structure. We minimized the number of parameters

of the walking device as well as its control architecture, its sensorimotor system, and the energy consumption in active dynamic walking.

We showed how to optimize the morphology of a bipedal walking device, so that it is adapted to a specific environment (walking down a slope only driven by gravity). Than we changed the environment (to flat surface) and evolved a control architecture which was able to adapt to this change by using the advantages of the models morphology.

There was only one type of information about the model-environment-interaction coming from the foot contact sensor having three possible discrete values. This was enough to enable the system, controlled by the evolved *RNN*, to adapted successfully to the changed environment.

In our opinion considering the optimization of morphology and control is an important aspect in developing and building bipedal walking devices, if the reduction of energy consumption is desired.

References

1. Alexander, R. McN. Energy-saving mechanisms in walking and running. *J. of Experimental Biology*, 160:55–69, 1991.
2. Bongard, J. C., and Paul, C. Making evolution an offer it can't refuse: Morphology and the extradimensional bypass. *Lecture Notes in Computer Science*, 2159:401–411, 2001.
3. Collins, S. H., Wisse, M., and Ruina, A. A three-dimensional passive-dynamic walking robot with two legs and knees. *The Int. J. of Robotics Research*, 20(7):607–615, 2001.
4. Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. The development of honda humanoid robot. *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1321–1326, 1998.
5. Mautner, C., and Belew, R. K. Evolving robot morphology and control. *Proc. of Articial Life and Robotics*, 1999.
6. McGeer, T. Passive dynamic walking. *Int. J. of Robotics Research*, 9(2):62–82, 1990.
7. McMahon, T. A. Mechanics of locomotion. *Int. J. of Robotics Research*, 3(2):4–28, 1984.
8. Nolfi, S., and Floreano, D. *Evolutionary Robotics: the biology, intelligence, and technology of self-organizing machines.* Cambridge, Massachusetts: The MIT Press, 2000.
9. Pasemann, F., Steinmetz, U., Hülse, M., and Lara, B. Robot control and the evolution of modular neurodynamics. *Theory in Biosciences*, 120:311–326, 2001.
10. Wischmann, S. *Entwicklung der Morphologie und Steuerung eines zweibeinigen Laufmodells.* Diplomarbeit, Naturwissenschaftlich-Technische Fakultät III, Universität des Saarlandes, Saarbrücken, Germany, 2003.
11. Witte, H. and Preuschoft, H., and Recknagel, S. Human body proportions explained on the basis of biomechanical principles. *Z. Morph. Anthrop.*, 78(3):407–423, 1991.

First Steps in Passive Dynamic Walking

Martijn Wisse[1] and Arend L. Schwab[2]

[1] Delft Biorobotics Laboratory, Delft University of Technology, The Netherlands
m.wisse@wbmt.tudelft.nl
[2] Laboratory for Engineering Mechanics, Delft University of Technology, The Netherlands
a.l.schwab@wbmt.tudelft.nl

Summary. Passive dynamic walking is a promising concept for the design of efficient, natural two-legged walking robots. Research on this topic requires an initial point of departure; a stability analysis can be executed only after the first successful walking motion has been found. Experience indicates that it is difficult to find this first successful walking motion. Therefore, this paper provides the basic tools to simulate a simple, two-dimensional walking model, to find its natural cyclic motion, to analyze the stability, and to investigate the effect of parameter changes on the walking motion and the stability. Especially in conjunction with the accompanying MATLAB[a] files, this paper can serve as a quick and effective start with the concept of passive dynamic walking.

1 Introduction

This text is written for prospective researchers of 'Passive Dynamic Walking'. Passive Dynamic Walking is an approach to investigate bipedal (two-legged) walking systems, be it humans or other bipedal animals, or bipedal walking robots that you want to build or control. Passive Dynamic Walking is a way to look at bipedal walking. Instead of seeing it as a continuous struggle to keep balance, bipedal walking is much better understood when regarding it as a continuous passive fall, only intermittently interrupted by a change of foot contact. A steady succession of steps can then be analyzed as a cyclic motion.

The approach of Passive Dynamic Walking as originally proposed by McGeer [4] has led to various insights regarding human walking [3], and has produced a number of natural and efficient walking machines [1, 4, 6], see Fig. 1. It is our opinion that huge progress can be made in both fields using the approach of Passive Dynamic Walking. However, experience indicates that it is rather difficult to get started with Passive Dynamic Walking, as one can

[a] For availability of the accompanying files, please try http://dbl.tudelft.nl or contact the first author.

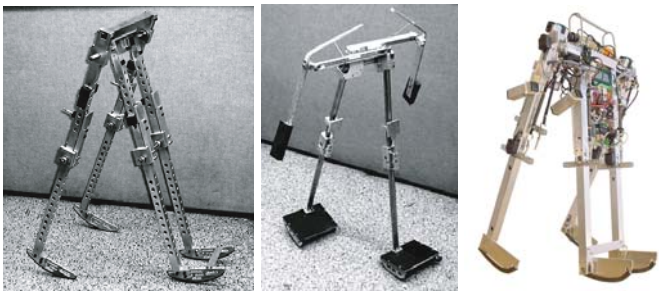


Fig. 1. Prototypes of passive dynamic walking bipeds that have been developed over the years. *Left*: Copy of Dynamite, McGeer [4], *middle*: 3D walker, Collins et al. [1], *right*: Mike, Wisse and Frankenhuyzen [6]

start analysis only after at least one successful walking motion has been found. This text serves as a guide to that first start.

We will present the complete simulation procedure for a simple, two-dimensional passive dynamic walker. The model is realistic enough to enable the construction of a physical prototype with corresponding behavior. Section II describes the required algorithms for a computer simulation that will predict a walking motion after a proper launch of the biped. This section includes the model description, the derivation of the equations of motion, numerical integration, heel strike detection and the derivation of the impact equations. Section III focuses on the analysis of the step-to-step progression of disturbances on the walking motion, encompassing the selection of a Poincaré section and a linearized stability analysis.

The text is accompanied by a set of MATLAB (version 5.2 or higher) files that will provide an operational programming example for a quick start. The following sections will guide you through the functions and background of each of the files.

2 Forward Dynamic Simulation

Model

The simplest system that can perform a Passive Dynamic Walking motion consists of two rigid legs interconnected through a passive hinge. We will study a two-dimensional model for the sake of simplicity. A real-world prototype can be made to behave (more-or-less) two-dimensional through the construction of two symmetric pairs of legs, see Fig. 2. The corresponding dynamic model is shown in Fig. 2.

We will make a number of assumptions to keep the simulation manageable. First, we assume that the legs suffer no flexible deformation and that the hip joint is free of damping or friction. Second, we idealize the contact between

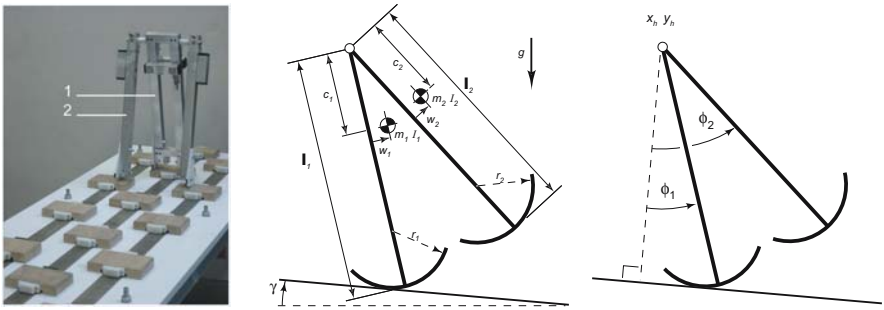


Fig. 2. (*Left*): Prototype Passive Dynamic Walking robot with four legs (two-dimensional walking behavior), walking on a checkerboard surface to prevent foot-scuffing at mid-stance. *Middle*: Parameters of the simulation model. (*Right*): four degrees of freedom of the simulation model; the position of the hip and the absolute leg angles

the foot and the floor, assuming perfectly circular feet that do not deform or slip, while the heel strike impact is modeled as an instantaneous, fully inelastic impact where no slip and no bounce occurs. Finally, the floor is assumed to be a rigid and flat slope with a small downhill angle.

There is one problem due to oversimplification of the model. Contrary to humans who have knees, the legs of the model cannot extend or retract, which inevitably leads to foot-scuffing at mid-stance. In a real-world prototype this problem is solved by covering the floor with a checkerboard pattern of tiles that provide foot clearance for the swing foot, see Fig. 2. In the computer simulation, we will simply assume that there is no interference between the floor and the swing foot under certain conditions, as described in the section 'Heel strike detection'.

Based on these assumptions, the model is defined with 14 parameters, which is done in the file wse_par.m. The world is parameterized with gravity g and slope angle γ. A leg must be parameterized as a single rigid body with a mass m, a moment of inertia I, the coordinates for its center of mass with respect to the hip in vertical direction c and in horizontal direction w, the leg length l and the foot radius r. An idealized model consists of two completely equal legs. However, we have noticed that a small difference in parameter values between the legs can strongly influence the walking behavior, so the model will be prepared for legs with different parameter values. All parameters are summarized in Table 1 in which we have also provided a set of default parameter values that should lead to a successfully walking model or prototype.

The number of degrees of freedom of this model requires some attention; although the two legs each have two position and one orientation coordinate in a two-dimensional world resulting in a total of six degrees of freedom (twelve states when including the velocities), only three states are independent at the

Table 1. Parameters for a simple passive dynamic walking model corresponding to Fig. 2. The given default parameter values were chosen to 1) comply with a realistic prototype, and 2) provide stable simulation results

World			
gravity	g	9.81	m/s^2
slope angle	γ	0.01	rad
Leg 1 and 2			
length	l	0.4	m
foot radius	r	0.1	m
CoM location	c	0.1	m
	w	0	m
mass	m	1	kg
mom. of inertia	I	0.01	kgm^2

start of a step. We get from twelve to three by successively considering the hip joint constraint, the foot contact, and the Poincaré section. First, the hip joint constrains two degrees of freedom (four states) so that the model has only four independent generalized coordinates, x_h, y_h, ϕ_1 and ϕ_2, see Fig. 2. Second, the foot contact constrains two more degrees of freedom (again four states), leaving only ϕ_1 and ϕ_2 as independent coordinates. The hip coordinates depend alternatingly on the one or the other foot contact, which is calculated in the file wse_dep.m. Third, we take a Poincaré Section of the cyclic walking motion. This means that we will focus our attention on the start of each step defined as the instant just after heel strike when both feet are in contact with the floor, which makes one more state dependent; only one leg angle is independent, the other is the same but opposite in sign. Together with the two independent velocities, there are three independent initial conditions that completely define the state at the start of a step, see Table 2. The definition of the initial conditions takes place in the file wse_ic.m. The values in Table 2 together with the default parameter values in Table 1 will result in a cyclic and stable walking motion.

Next to defining initial conditions for the model coordinates, we also need to define the foot contact coordinates. The actual foot contact point travels forward as the model 'rolls' forward over the sole of its circular feet. Therefore we appoint a single, fixed location as foot contact coordinate for the entire duration of a step. This location is defined as the actual point of contact if the leg angle is zero. The piecewise non-holonomic nature of walking systems requires that the foot contact coordinates are re-evaluated after each step. The initial values for the foot contact locations are set rather arbitrarily to zero in Table 2.

Table 2. Initial conditions for a simple passive dynamic walking model corresponding to Fig. 2. Leg 1 is chosen as the initial stance leg. The given default values will, in combination with the default parameter values in Table 1, result in a stable cyclic walking motion

Independent initial conditions			
Stance leg (leg 1) angle	ϕ_1	0.2015	rad
Stance leg (leg 1) angular velocity	$\dot{\phi}_1$	−1.4052	rad/s
Swing leg (leg 2) angular velocity	$\dot{\phi}_2$	−1.1205	rad/s
Dependent initial conditions (`wse_dep.m`)			
Swing leg (leg 2) angle	ϕ_2	−0.2015	rad
Hip horizontal displacement	x_h	0.0802	m
Hip vertical displacement	y_h	0.3939	m
Hip horizontal velocity	$\dot{x}_h$	0.5535	m/s
Hip vertical velocity	$\dot{y}_h$	0.0844	m/s
Initial foot contact coordinates			
Foothold location stance leg (leg 1)	xf_1	0	m
Foothold location swing leg (leg 2)	xf_2	0	m

Derivation of Equations of Motion

The equations of motion are the heart of the computer simulation. For our model we will first derive the generalized equations of motion for the two legs plus hip joint, then derive the algebraic equations that describe the alternating foot contact, and finally put these together in a system of DAE's – Differential Algebraic Equations. The equations in this section correspond to the file wse_eom.m.

Let's first consider the system of legs and hip without foot contact. As explained above, that system has four independent *generalized* coordinates $\mathbf{q}$. Their accelerations are calculated with the set of equations

$$\overline{\mathbf{M}}\ddot{\mathbf{q}} = \overline{\mathbf{f}} \tag{1}$$

with the generalized mass matrix $\overline{\mathbf{M}}$ and the generalized force vector $\overline{\mathbf{f}}$. They are constructed with the principle of virtual power and d'Alembert inertia forces (the so-called TMT-method) resulting in

$$\overline{\mathbf{M}} = \mathbf{T}^T\mathbf{M}\mathbf{T}, \qquad \overline{\mathbf{f}} = \mathbf{T}^T[\mathbf{f_g} - \mathbf{M}\mathbf{h}] \; . \tag{2}$$

In this $\mathbf{M}$ and $\mathbf{f_g}$ are the terms from 'normal' Newton-Euler equations of motion, i.e. without hip joint constraints and thus for six coordinates. The matrix T transfers the independent generalized coordinates $\dot{\mathbf{q}}$ into the velocities of the center of mass of the bodies $\dot{\mathbf{x}}$. The vector $\mathbf{h}$ holds the convective accelerations. $\mathbf{T}$ and $\mathbf{h}$ are generated by running wse_sde.m once, which creates the file wse_mat.m containing all necessary matrices.

With equation (2) we can calculate the accelerations for the two legs while ensuring that they remain connected at the hip. However, the system is in free

fall like this as we have not yet incorporated the contact between the feet and the floor. This contact is described with two equations per leg. First, the foot should be at floor level. Since we apply circular feet, the vertical constraint equations becomes

$$g_y = y_h - (l - r) * cos(\phi) - r \tag{3}$$

where g_y must be zero to fulfill the constraint. Second, the horizontal displacement of the foot must be related to the leg angle plus some initial offset (x_f) depending on where the foot has landed,

$$g_x = x_h + (l - r) * sin(\phi) + r * \phi - x_f \tag{4}$$

where g_x must be zero to prescribe pure rolling without slip.

To construct the complete set of DAE's we must first determine which foot is in contact, as only one set of foot contact constraints is active at a time. We will need the second derivative of these constraint equations (in the form of **D** and **D2**) to allow a combination with the equations of motion in the total system of equations

$$\begin{bmatrix} \overline{M} & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_c \end{bmatrix} = \begin{bmatrix} \overline{F} \\ D2^T \dot{q}^2 \end{bmatrix} \tag{5}$$

Solving these equations at any instant will provide the generalized accelerations $\ddot{\mathbf{q}}$ and the foot contact forces $\mathbf{f_c}$ at that instant.

Numerical Integration

The next step is to go from accelerations at any instant to a continuous motion. To obtain that motion numerical integration is needed, which is done in the file wse_rk4.m. We use the classical Runge-Kutta 4 method, which calculates in four intermediate steps the positions and velocities at time $t + \Delta t$.

One of the problems of numerical integration is the accumulation of numerical errors. The overall error can be checked by inspection of the energy content of the system, as the sum of kinetic and potential energy should be constant for a passive walker. An example of such energy checks is given in the file wse_ech.m. Otherwise, one could check stride characteristics such as stride time or stride length, and investigate how much these change by halving the integration step Δt.

Next to the overall error, there is the problem of non-satisfied constraint conditions. The accumulating numerical errors easily lead to a foot that sinks into the ground or flies away. The source of this type of errors is the fact that in equation (5) there are only second derivatives of the constraint equations, which only impose that the *acceleration* of the foot is zero. A small round-off error leads to huge position displacements after a while. Therefore, the file wse_rk4.m frequently calls the file wse_dep.m which recalculates the hip coordinates and velocities as a function of the independent leg angles and angular velocities so that the foot constraints are met.

Heel Strike Detection

During normal walking some events take place every step, whereas in the case of a fall a few other events could take place. To start with the latter, falling forward, falling backward, and losing ground contact (too high velocity) are three possible events. At every step, the file wse_evd.m checks for each of these terminal events and reacts by stopping the simulation.

During continuous locomotion, at every step a heel strike impact occurs, followed by a change of stance foot. This event is detected by monitoring the clearance of the swing foot (3). Zero clearance means that either a genuine heel strike has occurred or that the swing leg has briefly reached floor level during mid-stance. To distinguish between the two, the file wse_evd.m contains a four-level decision tree;

`IF`

- the vertical distance between the swing foot and the floor has changed sign, `AND`
- the stance leg has passed mid-stance (i.e. its direction of motion is away from the vertical position), `AND`
- the swing foot is currently below floor level, `AND`
- the legs are not parallel but in a spread configuration

`THEN` there must have been a valid heel-strike somewhere between the previous and the current integration time step.

If this is detected, an interpolation is necessary to determine the exact instant of heel contact. We approximate the motion between the timesteps t_{n-1} and t_n with a third order polynomial and determine when this polynomial passes through zero, see Fig. 3. This is done in wse_int.m. After this operation we know the precise instant of heel contact and the state of the model at that instant.

Derivation of Impact Equations

We assume that heel strike is a fully inelastic impact between the forward foot and the floor. During the heel-strike impact there are very high forces for a

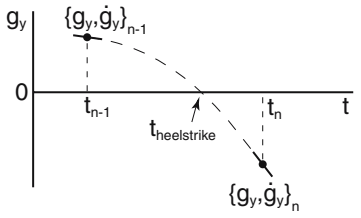


Fig. 3. Interpolation with third order polynomial to find the instant of heel strike between to integration steps. The clearance function g_y is given by (3)

very short time. This process can be interpreted as an impulsive motion, an instantaneous event in which the velocities change but not the positions of the model elements. To calculate this we can use the same equations of motion as (5) if we apply an integration over the impact duration and take the limit of this duration to zero. The result is a system of impact equations with much resemblance to (5):

$$\begin{bmatrix} \overline{M} & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ f_c \end{bmatrix} = \begin{bmatrix} \overline{M}\dot{q}^- \\ 0 \end{bmatrix} \tag{6}$$

The D matrix again represents the foot constraints, and is equal to the D used in (5). We must carefully decide which foot constraints are active during heel strike and which are not. At heel strike, both feet are at floor level, so both could possibly participate in the impact. However, the contacts are unilateral which means that only compressive forces can occur. We should only incorporate those constraint equations that result in a compressive impulse. For our model during normal walking, it turns out that only the forward foot does participate whereas the hind foot does not. Presumably the hind foot will obtain an upward velocity as a result from the impact calculation. If it doesn't, the assumption was wrong and we should have incorporated *both* feet in (6), which would have resulted in a full stop. The file wse_evd.m checks for this.

Walking Cycle

Now we have sufficient tools and algorithms to simulate a continuous walking motion. Let's give the model some initial conditions and see how many steps it will take or how it might fall. The file wse_scw.m ties all previously mentioned files together. Use it by first setting the appropriate parameter values and initial conditions in wse_par.m and wse_ic.m and then running wse_scw.m. The simulation results are then stored in the large matrices `t_t`, `q_t`, `qd_t`, `f_t`, and `g_t`, accessible from MATLAB's base workspace as global variables. To visualize the results, one can use and modify wse_fig.m which plots some basic graphs (Fig. 4), or wse_ani.m which displays a simple animation of the resulting motion, see Fig. 5.

3 Step-to-Step Stability Analysis

Stability

The most important characteristic of a walking machine is its stability; it should not fall down. According to the classical interpretation, this requires postural control at every instant of the motion, aimed to keep the center of gravity above the support polygon. We believe that this static approach (and related approaches such as 'ZMP' control [5]) are suitable for *standing* but

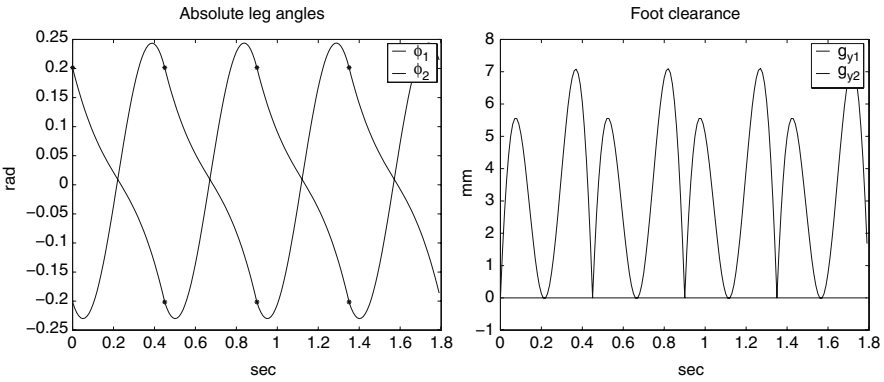


Fig. 4. Result figures produced by `wse_fig.m`.

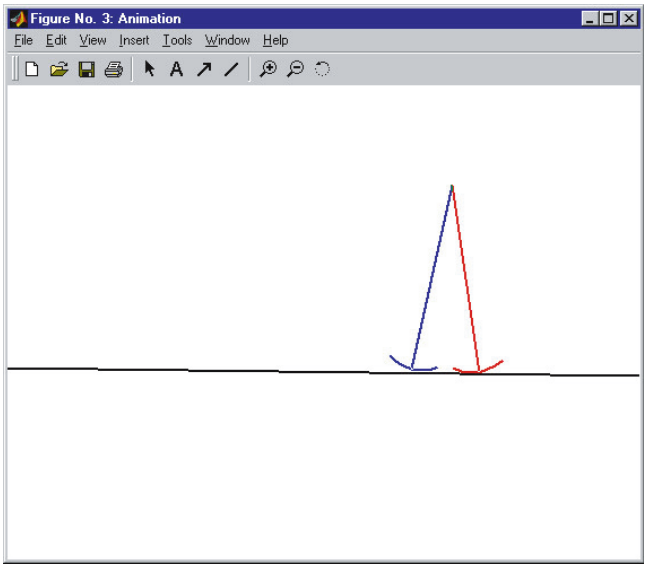


Fig. 5. Animation screen produced by `wse_ani.m`.

not for *walking*. As said before, walking should be regarded as a continuous passive fall with intermittent changes of foot contact. Instead of analyzing the balance at every instant we should analyze the stability of the entire cyclic motion in a step-to-step analysis.

A step-to-step analysis allows us to concentrate on the initial conditions only; the rest of the step is then a predictable passive motion. We can present the initial conditions in a phase-space graph (a plot of ϕ versus $\dot{\phi}$), where any point in the graph represents one specific combination of values for the three independent initial conditions. All points lead to a subsequent motion, but only some of them are successful steps. The end of a successful step is the

start of a new one, and so some points in the graph map to some others. This is called 'Poincaré mapping'.

With a little bit of luck (depending on the model parameter values) there are one or two points in the graph that map onto themselves. These are called *fixed points*. They represent a continuous walking motion with all identical steps, which is called a *limit cycle*. With some more luck, one of the fixed points is stable; if the initial conditions are a small deviation away from the fixed point, this deviation disappears over a number of steps until the walker is back in its limit cycle. This stability for small errors is analyzed in the next section. A question that remains is 'how small is small'; for what initial conditions will we still find convergence to the limit cycle? That question is answered with an analysis of the basin of attraction, but that is outside the scope of this paper.

Linearized Stability

We need to *find* a fixed point and to *analyze its linear (small-error) stability*. This is easiest understood in reverse order, so for now let's assume that we already know a fixed point. Actually we do, see Table 2. The three independent initial conditions are represented with $\mathbf{v} = [\phi_1, \dot{\phi}_1, \dot{\phi}_2]^T$, whereas we'll call the fixed point $\mathbf{v}_{fp}$. The Poincaré mapping is represented with the nonlinear function S, so that

$$\mathbf{v}_{n+1} = \mathbf{S}(\mathbf{v}_n) \tag{7}$$

where S is a short notation for the complete simulation of one walking step including the heel strike impact and a mirroring of the legs to compare $\mathbf{v}_{n+1}$ with $\mathbf{v}_n$. All initial conditions can be written as a sum of the fixed point plus some deviation:

$$\mathbf{v}_n = \mathbf{v}_{fp} + \Delta\mathbf{v}_n \tag{8}$$

Although S is highly nonlinear, for small deviations from $\mathbf{v}_{fp}$ we can approximate the mapping with a linearization according to

$$\begin{aligned} \mathbf{v}_{fp} + \Delta\mathbf{v}_{n+1} &= \mathbf{S}(\mathbf{v}_{fp} + \Delta\mathbf{v}_n) \approx \mathbf{S}(\mathbf{v}_{fp}) + \mathbf{J}\Delta\mathbf{v}_n \\ \textit{with} \quad \mathbf{J} &= \frac{\partial \mathbf{S}}{\partial \mathbf{v}} \end{aligned} \tag{9}$$

This equation simplifies to $\Delta\mathbf{v}_{n+1} = \mathbf{J}\Delta\mathbf{v}_n$. The Jacobian (matrix of partial derivatives) $\mathbf{J}$ here is the key to our linearized stability analysis. Basically it multiplies the errors at step n to produce those at step $n + 1$. If the multiplication factor is between -1 and 1, errors decrease step after step and the walker is stable. The multiplication factors are found in the eigenvalues of $\mathbf{J}$ that should all three have a modulus smaller than 1. In the case of the parameter values of Table 2 the eigenvalues are 0.65, 0.22 + 0.30i, and 0.22 − 0.30i, so the model is linearly stable.

Unfortunately, the Jacobian $\mathbf{J}$ is not readily available. It must be obtained by numeric differentiation by means of four full-step simulations; once for the

initial conditions of the fixed point and three times to monitor the effect of a small perturbation on each of the initial conditions. This is done in the file wse_lca.m. The resulting eigenvalues of $\mathbf{J}$ tell us whether a fixed point is stable or not. However, more than a simple 'yes' or 'no' cannot be expected, as the actual eigenvalues and eigenvectors do not provide much more insight. To determine which model is 'more stable', one should investigate the maximally allowable disturbance size, which can be found by analysis of the basin of attraction (not in this paper).

Finding a Fixed Point

Now we know how to analyze a fixed point, but how do we *find* it? The approximation of (9) can also be applied to a set of initial conditions *close to* the fixed point (which we need to guess). This will provide an estimate for $\mathbf{J}$. With that estimate and with the difference between the beginning ($\mathbf{v}$) and the end ($\mathbf{S}(\mathbf{v})$) of a step, a Newton-Raphson iteration can be performed that will quickly converge to the fixed point. The iteration procedure as used in wse_lca.m is as follows:

$$\begin{aligned} &\texttt{repeat} \\ &\quad \Delta\mathbf{v} = [\mathbf{I} - \mathbf{J}]^{-1}(\mathbf{S}(\mathbf{v}) - \mathbf{v}) \\ &\quad \mathbf{v} = \mathbf{v} + \Delta\mathbf{v} \\ &\texttt{until } |\Delta\mathbf{v}| < \epsilon \end{aligned} \tag{10}$$

If this procedure is started for example with $\{\phi_1, \dot{\phi}_1, \dot{\phi}_2\} = \{0.15, -1, -1\}$, it takes 7 iteration steps ($\pm$ 20 seconds on a 2GHz PC) to arrive at the fixed point with $\epsilon < 10^{-12}$.

Note that the file wse_lca.m always simulates only a single step. In order to compare the end state with the begin state, the end state must be mirrored. This is the standard procedure used in most passive dynamic walking researches, although it is not entirely realistic. In case the model has two legs with different mass properties or in some other special situations [2], the limit cycle analysis should be performed on two subsequent steps which eliminates the necessity for mirroring. The drawback of this is that there is more chance of a fall and thus more difficulty in finding (unstable) cycles with a bad initial guess for the initial conditions.

4 Conclusion

This paper provides the basic tools to simulate a simple, two-dimensional walking model, to find its natural cyclic motion, to analyze the stability, and to investigate the effect of parameter changes on the walking motion and the stability. Especially in conjunction with the accompanying MATLAB files, this paper can serve as a quick and effective start with the concept of passive dynamic walking.

References

1. S. H. Collins, M. Wisse, and A. Ruina. A who legged kneed passive dynamic walking robot. *International Journal of Robotics Research*, 20(7)-:607–615, July 2001.
2. M. Garcia, A. Chatterjee, A. Ruina, and M. J. Coleman. The simplest walking model: Stability, complexity, and scaling. *ASME J. Biomech. Eng.*, 120(2):281–288, April 1998.
3. A. D. Kuo, Energetics of actively powered locomotion using the simplest walking model. *Journal of Biomechanical Engineering*, 124-:113–120, February 2002.
4. T. McGeer. Passive dynamic walking. *Intern. J. Robot. Res.*, 9(2):62–82, April 1990.
5. M. Vukobratovic, B. Borovic, D. Surla, and D. Stokic. *Biped Locomotion: Dynamics, Stability, Control and Applications*. Springer-Verlag, Berlin, 1990.
6. M. Wisse and J. van Frankenhuyzen. Design and construction and mike a 2d autonomous biped based on passive dynamic walking. In Proc., *Conference on Adaptive Motion of Animals and Machines, AMAM*, Kyoto, Japan, 2003. Paper number WeP-I-1 (8 pages).

Controlling Walking Period of a Pneumatic Muscle Walker

Takashi Takuma[1], Koh Hosoda[1,2], Masaki Ogino[1], and Minoru Asada[1,2]

[1] Dept. of Adaptive Machine Systems, Graduate School of Engineering
[2] HANDAI Frontier Research Center
{takuma, ogino}@er.ams.eng.osaka-u.ac.jp
{hosoda, asada}@ams.eng.osaka-u.ac.jp

Summary. A passive dynamic walker can walk down a shallow slope without any actuation. If the dynamics of the biped is effectively utilized to realize passive walking, the robot walks with energy-less locomotion. An antagonistic pair of pneumatic actuators is one candidate to realize such *quasi*-passive walking using both passive and active motions like some animals do. This paper presents a simple control of to enlarge the stability of biped walking utilizing passive dynamics. First, we introduce the design of the biped robot used for experiments with antagonistic pairs of pneumatic actuators. Then, we propose to regulate opening duration of the valves based on the observed walking cycle. Finally, experimental results are shown to demonstrate the effectiveness of the proposed control scheme.

1 Introduction

A passive dynamic walker can walk down a shallow slope without any actuation [1], and is regarded to be one key technology for realizing human-like biped walking not only on slope but on various terrain. If the dynamics of the biped is effectively utilized, we can decrease the effort to design the desired trajectories for joints, eliminate several actuators, and realize less energy consumption with biped walking. To realize such *quasi*-passive dynamic walkers, the actuators should realize both active and passive motions. Although several researchers adopted electrical motors with reduction gears that can provide large torque [4, 5], the back-drivability cannot be preserved because of its friction, and therefore, it is difficult to swing the leg in a passive manner. Sugimoto and Osuka adopted direct-drive motors which made the weight of the robot heavy [3].

An antagonistic pair of pneumatic actuators should be one candidate to realize both passive and active motions like some animals do. Wisse et al. built a biped robot with such pairs of pneumatic actuators, and realized less energy consumption biped walking by utilizing the passive dynamics [2]. They adjusted opening duration of air-supply valves by a trial and error manner.

The stability of the walking almost depended on the robot dynamics since the duration is fixed, and as a result, the negotiable terrain is relatively limited.

If the pairs of actuators are controlled to track a given trajectory, they suffer from their friction and hysteresis. Although several researchers proposed control schemes to compensate such non-linearity by utilizing the dynamic models [7, 8], or by using fast switching on/off valves [9], the performance was so far not satisfactory. If the robot dynamics is properly utilized, the trajectory tracking control is not necessary for walking stability. In this paper, a simple control of the duration is proposed to enlarge the stability of the biped walking utilizing passive dynamics. First, we introduce the design of the biped robot used for experiments with antagonistic pairs of pneumatic actuators. Then, we propose to regulate opening duration of the valves based on the observed walking cycle. Finally, the experimental results are shown to demonstrate the effectiveness of the proposed control scheme.

2 Development of a Biped Walker with Antagonistic Pairs of Pneumatic Actuators

We have developed a biped robot driven by antagonistic pairs of pneumatic actuators shown in Fig. 1. We adopted McKibben artificial muscles [6] made by HITACHI Medical Corporation [10]. Its length and radius are 0.2[m] and 0.020[m] (when it contracts), respectively. It generates approx. 800[N] when the pressure in the inner tube is 0.7[MPa]. Figure 1 shows an overview of a control system for one joint driven by an antagonistic pair of actuators. Each actuator has supply and exhaust valves (SMC Corporation [11]) that are controlled by a D/A converter through an amplifier. Pressure inside the actuator is measured by a pressure sensor and the sensory datas are obtained through an A/D converter.

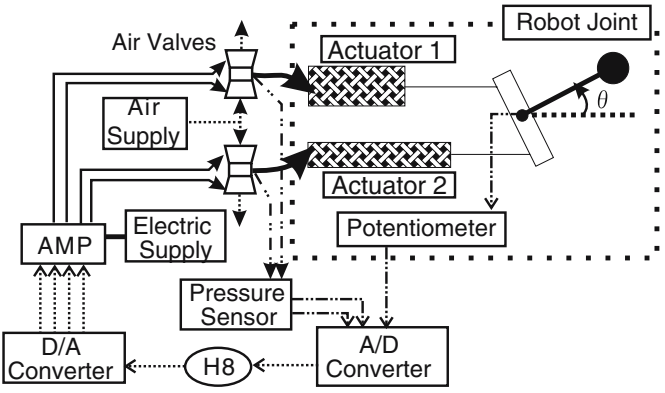


Fig. 1. Overview of the system

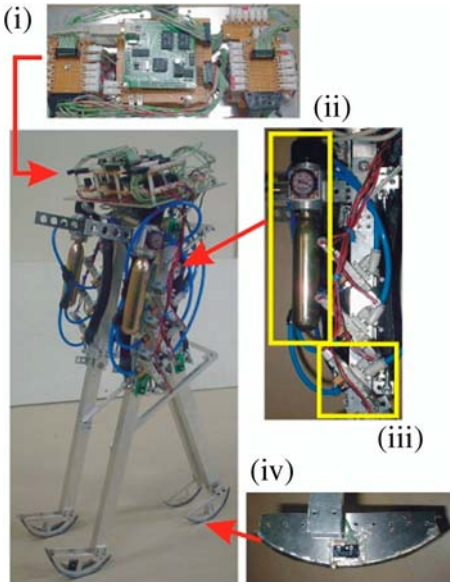


Fig. 2. Real biped walker with McKibben muscle actuators

Figure 2 shows the structure of the developed planar walking robot. The height, width, and weight of the robot are 0.750[m], 0.350[m], and 5[kg], respectively. It has four legs: two connected pairs of legs (outer and inner). Lengths of thigh and shank are 0.3[m] and 0.35[m], whose weights are 2.16[kg] and 0.48[kg], respectively. It has a micro computer(H8/3067) with D/A and A/D converters (i). As it is designed to be self-sufficient, it has two CO_2 bottles on it as air sources each of which weighs 0.7[kg](ii). The pressure is 1.2[MPa] and is regulated by a pressure regulator into 0.7 [MPa]. It has round-shape soles whose curvature and length are 0.125[m] and 0.16[m], respectively, determined by trial and error (iv). The sole has an ON/OFF switch that detects collision with the ground.

3 Regulation of Valve-Opening Duration for Stabilizing Biped Walking

3.1 Relation Between Robot Design and Control

If we can apply model-based control, we can ensure strict stability of the robot. However, we cannot apply such model-based control since it is hard to model the complicated interaction between the robot and the terrain. If the biped is properly designed to realize passive dynamic walking, on the other hand, it is expected that it can negotiate various terrains with little control. Simple control should be, then, enough for increasing the walking stability.

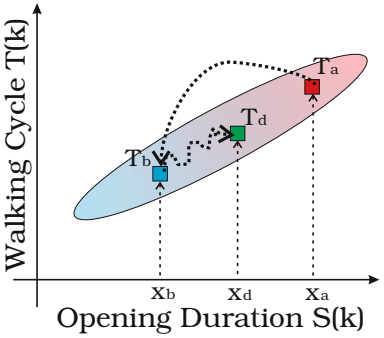


Fig. 3. Correlation between walking cycle and manipulated variable. Following the control law (1), the duration of opening valves is set x_b when the walking cycle is slower than desired cycle T_d. It is expected that the walking cycle changes faster from T_a to T_b, and then converges to desired cycle T_d

In this paper, we focus on the relation between the walking cycle and the opening duration of the valves and propose to regulate opening duration of the valves based on the observed walking cycle. Since the biped is properly designed to realize quasi-passive dynamic walking, we can let the robot walk even with small perturbation in the control, that is, perturbation in the duration. We can plot, then, the relation between the duration and the walking cycle that is observed by the touch sensors. Although the strict stability is not proved, we may find an appropriate controller.

If we can find a weak-linear relation between the duration and the walking cycle like depicted in Fig. 3, we can apply feedback control based on the relation, which may increase the stability of the biped walking utilizing passive dynamics:

$$S(k) = S(k-1) - K(T(k-1) - T_d)\ , \tag{1}$$

where $S(k)$, $T(k)$, T_d, K are the opening duration of the valve, the walking cycle at the k-th step, the desired walking cycle, and the feedback gain that is determined by the weak-linear relation, respectively.

3.2 Stabilizing Walking by Regulating Opening Duration of Valves

We operate the valves to generate a walking pattern shown in Fig. 4. The sequence is:

(i) at the beginning of k-th step, all valves are closed until T_0[s](a fixed value determined by trial and error),
(ii) At T_0[s], the supply valve of the hip flexor and exhaust valve of the hip extensor are opened for $S(k)$[s]. During this period, the knee of the free leg first bends, and then, stretch to be ready for the coming impact.
(iii) all valves are closed so that the robot keeps the posture until the free leg touches the ground.

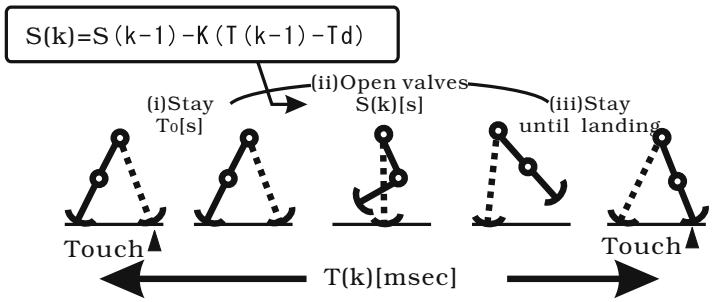


Fig. 4. Walking pattern and the controller

In the operations (i) and (iii), the collision with the ground is detected by the switch on the sole shown in Fig. 2(iv). Operations (i)–(iii) correspond to items (i)–(iii) in Fig. 4, respectively. Let $T(k)$ be a walking cycle from the heel strike to the other heel strike at k-th step.

4 Experimental Results

4.1 Relationship Between the Duration of Opening Valves and Walking Cycle

We observe the relationship between the input durations of opening valves to swing leg $S(k)$ and the output walking cycles $T(k)$.

In the experiment, the robot walks on the plane with the fixed duration of opening valves $S(k) = 320$[ms], and on the pre-determined step, the duration is changed randomly between 200[ms] to 440[ms]. The result shown in Fig. 5 indicates the relationship between the duration of opening valves and walking cycle. The duration and the cycle have a positive correlation in each case. The correlation rate is 0.72 when the duration of opening valves to swing a pair of inner legs is changed. As a result of using a real robot, it is confirmed that the walking cycle is influenced by the duration of opening valves, and (1) is an effective law to control walking cycle.

The cycle of the steps is different when outer and inner legs swings because the dynamics of each pair of legs is different. In subsequent experiments, the desired walking cycle when the pair of outer and inner legs swings is set as $T_d = 640$[ms] and $T_d = 560$[ms], respectively.

4.2 Testing Negotiation Ability with Terrain Changes

Descending One Difference in Level

In order to evaluate the proposed control, we have an experiment to walk over one small difference in level of 4[mm] using the biped walker.

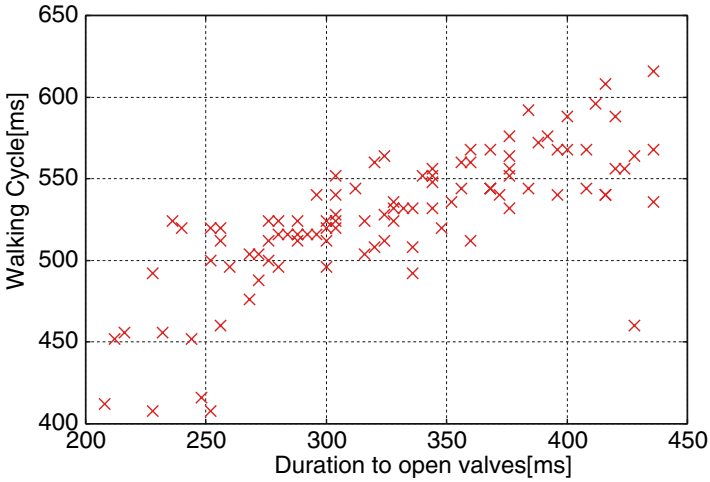
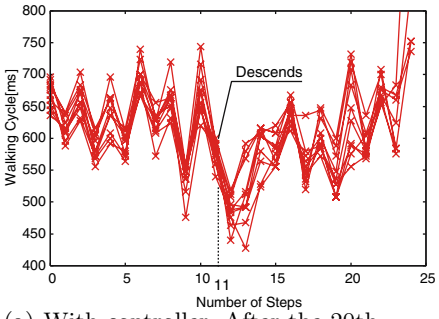
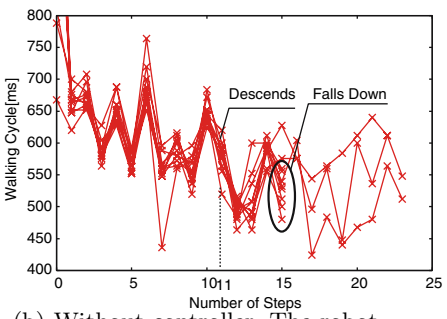


Fig. 5. Relation between the walking cycles and the durations of opening valves to rotate hip joint when a pair of inner legs swings. The cycle and the duration have positive correlation

Figure 6 shows variations of the cycles (a)with / (b)without cycle control by 10 trials in each case. In each figure, the dots "×" on lines express the walking cycles. The walker descends at 11th step, and the walking cycle becomes shorter in both case. As shown in Fig. 6(a), the walking cycle remains the same as before descending when the robot walks with the control. As shown in the Fig. 6(b), the robot often falls down at 15th step if it walks without control. Sometimes the robot without controller walks over the difference, but the cycles are kept shorter and do not indicate fixed patterns.



(a) With controller. After the 20th step, the cycle remains the same as before descending.



(b) Without controller. The robot often falls down at the 15th step or continues to walk getting its cycle faster

Fig. 6. The result of walking with/without the control when the robot walks over the one differences in level

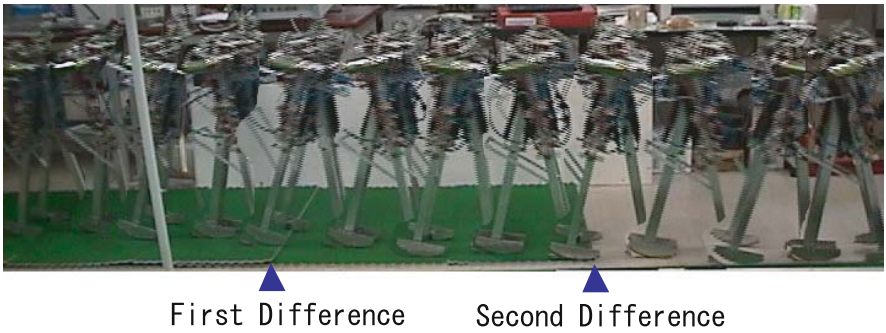


Fig. 7. The result of walking with the control when the robot walks over two differences in level

Descending Two Differences in Level

In the previous experiment, the walking cycle gets faster than before descending when the robot walks without control. We then have a experiment to walk over two differences in level shown in Fig. 7. In results, the robot with control can walk over two differences as shown in the Fig. 7 72 times of 100 trials, and the robot without control can walk over 27 times of 100 trials. The walking cycle of the robot without control gets more faster so that swing leg hits on the ground before swinging fully. The results of descending differences show that proposed control is effective to control walking cycle, and it can be said that the controller stabilize the walking against the terrain changes.

5 Conclusion

This paper introduces the developed biped walker that has McKibben artificial muscles. The height, width, and weight of the robot are 0.750[m], 0.350[m], and 5[kg], respectively. It has three degrees of freedoms, and it is designed to be self-contained.

Although the antagonistic pair of McKibben muscles is suitable for the quasi-passive dynamic walking, it is difficult to control the actuator because of its complex nonlinearity. We design the robot to realize walking with simple operation of opening/closing the air valves that drive the actuators to make the locomotion. We propose a simple control law for stable walking by changing the duration to swing the leg.

In the experiments, we observed first the relation between a duration of opening valves and a walking cycle using the real robot. In results, the duration and the walking cycle had positive correlation, which means that the walking cycle could be controlled by changing the duration of opening valves. Then the simple control law is proposed and we let the robot walk over one difference in level with and without control. When the robot walked without control, the

walking cycle was kept shorter after descending the difference, but the cycle with the control resumes to be as long as before descending the difference. The robot with control could often walk over two differences in level, but the robot without control could not so often walk over.

Our final goal is to build a robot that can walk against larger perturbation, and perform more dynamic motions such as running and jumping. In order to realize such motions, we are now planning to build up the mechanism of the robot which can change its dynamics easily, such as adding upper body or ankle joints. This is a future work.

References

1. T. McGeer, *Passive dynamic walking*, International Journal of Robotic Research, Vol. 9, No. 2, pp. 62–82, 1990
2. M. Wisse and J. van Frankenhuyzen, *Design and Construction of MIKE; 2D autonomous biped based on passive dynamic walking*, 2nd International Symposium on Adaptive Motion of Animals and Machines, WeP-I-1, October, 2002
3. Y. Sugimoto, K. Osuka, *Walking Control of Quansi-Passive-Dynamics-Walking Robot "Quartet III" based on Delayed Feedback Control*, Proceedings of 5th International Conference on Climbing and Walking Robots (CLAWAR 2002), Paris, France, 2002, September
4. Fumihiko Asano, Masaki Yamakita, and Katsuhisa Furuta *Virtual Passive Dynamic Walking and Energy-based Control Laws*, Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1149–1154, 2000
5. Fumihiko Asano, Masaki Yamakita, Norihiko Kawamichi, and Zhi-Wei LUO, *A Novel Gait Generation for Biped Walking Robots Based on Mechanical Energy Constraint*, Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland, 2002, October
6. Richard Q. van der Linde, *Design, Analysis, and Control of a Low Power Joint for Walking Robots, by Phasic Activation of McKibben Muscles*, IEEE Transactions on Robotics and Automation, Vol. 15, No. 4, August 1999
7. Darwin G. Caldwell, Gustavo A. Medrano-Cerda, and Mike Goodwin, *Control of pneumatic muscle actuators*, IEEE Control Systems, pp. 40–47, February 1995
8. Joachim Schröder, Kazuhiko Kawamura, Tilo Gockel, and Rüdiger Dillmann, *Improved control of humanoid arm driven by pneumatic actuators*, International Conference on Humanoid Robots 2003, CD-ROM, 2003
9. R.V. Ham, B. Verrelst, F. Daerden, and D. Lefeber, *Pressure control with on-off valves of Pleated Pneumatic Artificial Muscles in a modular one-dimensional rotational joint*, International Conference on Humanoid Robots, CD-ROM, October 2003
10. HITACHI Medical corp. web site: http://www.hitachi-medical.co.jp/english/index.htm
11. SMC Corporation web site: http://www.smcworld.com/

Evolutionary Design of an Adaptive Dynamic Walker

Joachim Haß, J. Michael Herrmann, and Theo Geisel

Max-Planck-Institut für Strömungsforschung, Bunsenstr. 10, D-37073 Göttingen
joachim@chaos.gwdg.de,
michael@chaos.gwdg.de,
geisel@chaos.gwdg.de

Summary. This work is concerned with passive dynamic walking on downhill slopes where gaits are at best of fragile stability. A hybrid Darwin-Boltzmann optimization technique is used to search for regions in parameter space where unstable gait cycles are likely and can subsequently be improved by a Newton method. Periodic gaits are present for parameters along a submanifold of the product space of parameters and initial conditions. These unstable gaits can be stabilized by small control actions. Reinforcement learning controllers, though being successful, turn out to be too slow for a real walking machines. A alternative control method has been derived from the homeokinetic principle. It implements a dynamics of the parameters of a quasi-linear controller and seems to provide an efficient and flexible control to support the intrinsic dynamic of a passive walker.

1 Introduction

Passive dynamic walking [1, 2] represents an interesting alternative to complex and energy-consuming control schemes to induce anthropomorphic bipedal locomotion. However, very small viable regions in parameter space have been reported from experiments and computational studies of passive walkers [2, 3]. Furthermore, stable passive walking can only be performed on down-hill ramps that are nearly flat. This tightly restricts the possible speed of the walker, which is determined by the slope.

In this work, we explore walking at larger slopes with and without stabilizing control. Although no stable fixed points may exist for higher slopes [2, 3], controlled behavior is known to be particularly efficient and flexible in the vicinity of unstable solutions of the passive dynamics [4, 5]. In order to identify unstable limit cycles of the passive dynamics, a numerical optimization algorithm is used, which allows us to identify parameters sets with good performance which subsequently can be improved and verified by Newton's method. In this way, conditions on the parameters are obtained, cf. Sect. 3.

Periodic gaits form a limit cycle in the space of the dynamic variables and can be characterized by fixed points of the stride map, which is given by the difference of dynamic variables after exactly one step. The stability of these fixed points is improved or achieved using reinforcement learning (Sect. 4), but this approach is too slow for real walkers. We favor instead an algorithm based on the principle of homeokinesis [10], which is implemented as an adaptive quasi-linear feedback controller and appears suitable for noisy and dynamic environments.

2 Dynamical Model of the Walker

A passive walker can be modeled as an inverted multi-link pendulum in two dimensions [2]. The simplest version of such a model is the inverted double pendulum which resembles a walker with stiff knees. The walker is supposed to be fixed to the ground with its standing leg, so the dynamic variables are the angles between the links and the ground.

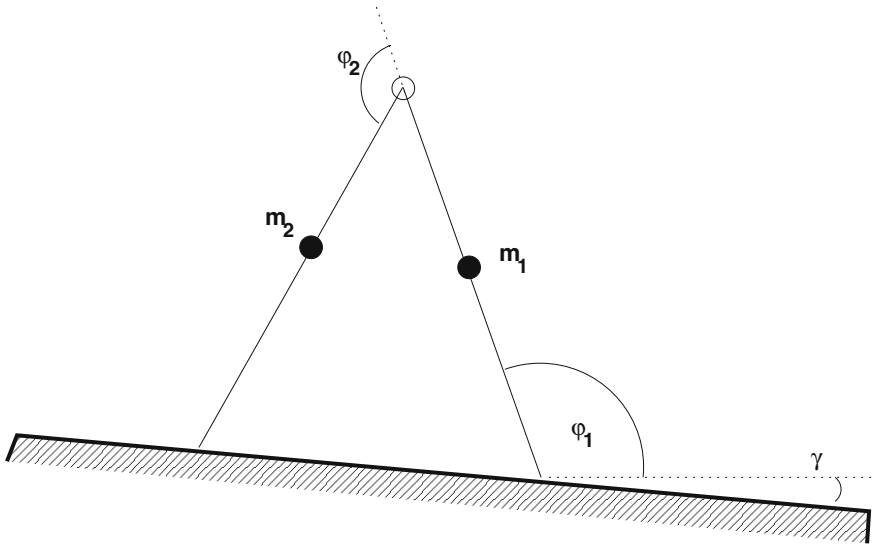


Fig. 1. Sketch of the walker with variable and parameter conventions. The position of the center of mass is measured from the free end of the legs

Collisions with the ground are regarded completely inelastic, instantly exchanging standing and swinging leg. Friction within the joints and sliding motion of the swing leg when passing the stance leg are also ignored. The pendulum dynamics is obtained using the Lagrange formalism and embedded in a program architecture that can be easily extended to more complex walkers.

The parameter space consists of the masses m and lengths l of the links, the position of the centers of mass c and radii of gyration r_{gyr} that determine the inertia by $I = m \times r_{gyr}$. Scaling of mass or length of the legs does not qualitatively change the dynamics, because all terms of the Lagrange function scale with the same order of m and l. We consider the two legs to be symmetric and ignore the masses and lengths in the following discussion.

3 Parameter Optimization

Our first objective is to obtain parameter combinations that lead to a periodic gait cycle. Since walking behavior crucially depends not only on the parameters but also on the initial conditions [3], they must be tuned simultaneously.

Here the only hardware parameters are the position of the center of mass c and the radius of gyration r_{gyr} for both legs. In the initial state, the walker is set in the double stance position, i.e. for initial conditions the link orientations can be expressed by the step distance between the legs as a single parameter σ. Together with the angular velocities ω_1, ω_2 and the two hardware parameters, this forms a five-dimensional parameter space. The following optimization scheme is well suited also for parameter spaces of higher dimensions as occurring in more complex walkers.

3.1 Optimization Technique and Fitness Function

In order to explore the parameter space in an efficient way, the problem is formulated as a numerical optimization task. The performance of each set of parameters is assessed by the fitness function

$$f \propto S \times exp((\Delta/\tau)^2) \times P(\sigma) \,, \tag{1}$$

where S marks the number of steps the walker takes until failure. Δ is the largest deviation of the angles and angular velocities between two successive steps. The squared exponential dependence with a small scale constant τ stresses periodicity as an important fitness factor. $P(\sigma)$ is a penalty function that reduces the fitness if the minimal step length σ of the walker falls below some threshold. This is necessary because the walker tends to prefer gaits with step lengths that are far too small to be realistic (cf. [6]).

The fitness function has a complex and for some parts even discontinuous dependence of the parameters, because small viable regions near periodic limit cycles are sharply separated from the rest of the parameter space. This excludes some algorithms, like gradient search from the outset. Instead, we use an optimization technique called NPOSA (new population-oriented simulated annealing) [7], which combines the paradigms of simulated annealing and genetic algorithms: Each parameter set of a randomly generated initial

population evolves itself by small random displacements. An *offspring* with fitness f which descents from an *ancestor* with fitness f_a is selected for reproduction in proportion to a Boltzmann factor $\exp(\beta(f_a - f))$.

Instead of proposing an explicit annealing scheme for the temperature parameter β, a *local temperature* β_i is assigned to each individual according to its fitness rank. Thus, selection pressure increases as the solutions get better while inferior individual get the chance to explore the parameter space more freely. By this method, we obtain a family of parameter sets with high fitness values, which allows us to analyze the diversity of successful solutions. Due to the stochastic nature of NPOSA no exact values yield, but regularly parameter configurations arise that are in a neighborhood of fixed points of the stride map. In this way good initial values for Newton's method are available which yields exact fixed-point parameters in the simulation.

3.2 Diversity of Successful Parameter Sets

Parameters are optimized independently for different slopes. For each slope, the optimization algorithm identifies a broad range of successful parameters with clear correlations between center of mass and radius of gyration that become more definite with increasing slope, cf. Fig. 2. Along the submanifolds defined by these dependencies, there are many sets of parameters that produce fixed points, including some pseudo-stable points which enable persistent walking even in presence of weak noise.

3.3 Fixed Points at Different Slopes

For parameters corresponding to a fixed point of the stride map, there is always a second set of initial conditions which gives rise to a second fixed

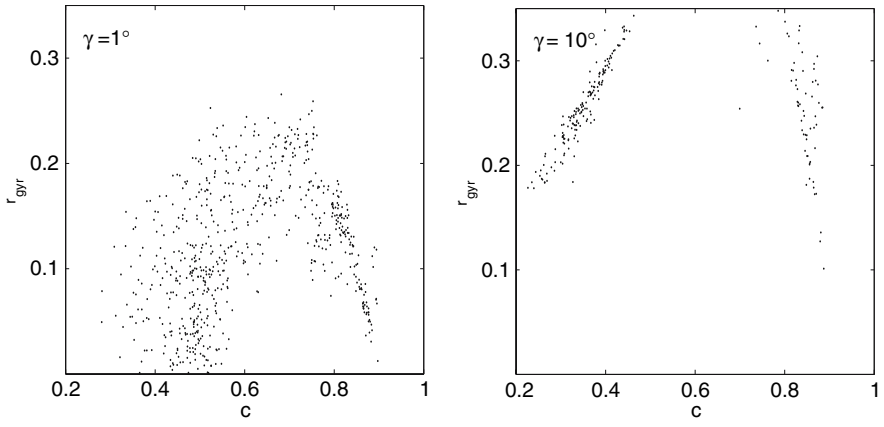


Fig. 2. Center of mass and radius of gyration for successful walkers at different slopes. The total length of a leg is set to 0.9

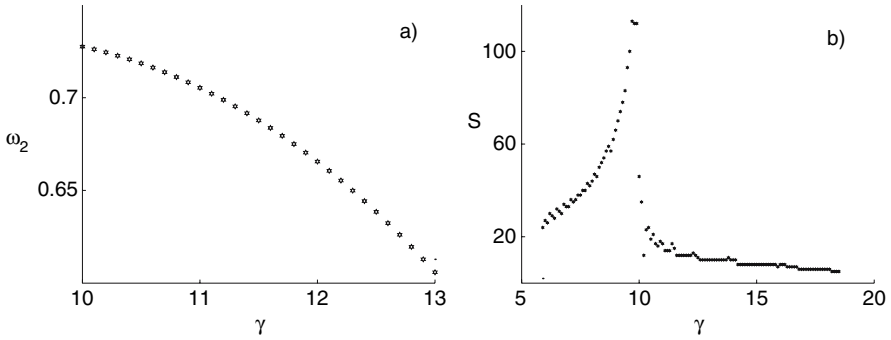


Fig. 3. Fixed points at different slopes. (**a**) Two fixed points coexist along smooth curves slightly separated from each other. Similar curves exist for the other parameters (**b**) Number of steps possible as a function of the slope. The short period solution (*stars*) is more stable than the long period solution (*dots*), cf. [2]

point close to original one. The two differ concerning their step lengths, corresponding to Garcia's "long period" and "short period" solutions [2]. For $\gamma = 0.1°$, the ratio of these periods is comparable to those in [2], which is not the case for $\gamma = 10°$. However, the distinction between long and short period fixed points seems to hold even if the assumption of small slopes can not be made.

Another interesting question is whether the fixed points preserve at changing slopes. For a wide range of slopes, fixed points show a continuous dependence on the slope, cf. Fig. 4a. The number of steps until failure,

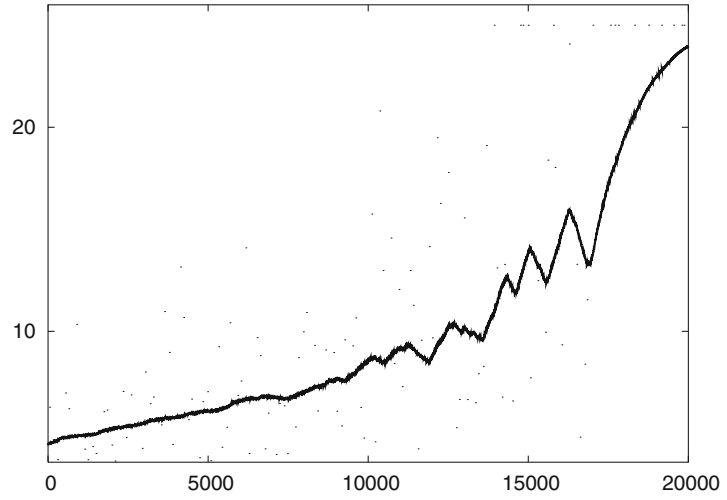


Fig. 4. Increase in walking time during reinforcement learning, expressed in terms of single trials (*dots*) and their time average (*line*)

however, exhibits an less smooth dependence, peaking at slopes slightly different from the target slope, cf. Fig. 4b. Although stability properties vary considerably with the slope, the number of steps is still large enough to allow for learning of an efficient adaptive controller, even at slopes distant from the original one.

4 Control

Preferentially, control forces should stay small and the walker should remain controllable even if slight changes in the environment occur, thus ensuring efficiency and flexibility. We tested two different control schemes accordant to these demands and their overall suitability for this task.

4.1 Reinforcement Learning

$\mathcal{Q}$-learning [8], a variant of the standard reinforcement learning, is based on dividing the state space into a number of discrete categories. To each category one out of a finite number of actions is assigned. The control action a^* to be applied at state category c is determined by

$$a^* = \arg\max_a \mathcal{Q}(c, a) \ , \tag{2}$$

the $\mathcal{Q}$-function being obtained by the iterative learning rule

$$\mathcal{Q}_{i+1}(c, a^*) = \mathcal{Q}_i(c, a^*) + \eta_{\mathcal{Q}} \left(r(c) + \gamma \max_a \mathcal{Q}_i(c', a) - \mathcal{Q}_i(c, a^*) \right) \tag{3}$$

where i is the iteration index, r is the immediate reward/punishment signal, and c' is the category reached from c by applying a^*. The $\mathcal{Q}$-value of a state-action pair indicates whether the application of an action promises transitions to states which are rewarded. Not only immediate rewards are considered, but also those that are expected at later states. The time horizon within with rewards are considered as relevant is controlled by the parameter $\gamma \leq 1$. In order to prevent the walker from excessive repeating of actions with only moderately high $\mathcal{Q}$-values, the actions are chosen according to (2) only with a certain probability and randomly otherwise, insuring in this way the exploration of the whole set of state-action pairs. The fraction of random actions is governed by an exploration rate which decays quadratically to zero during the learning process, while the learning rate $\eta_{\mathcal{Q}}$ in (3) decays linearly.

In the experiments (cf. Fig. 4) the reward signal is given by the time the walker is running. The state-space categories are formed by an adaptive vector quantizer, which is switched off after 10% of the total learning time, while the actions are implemented as direct variations of the angular velocities. The $\mathcal{Q}$-function is updated and a new action is selected only immediately after heel strikes. Although the conditions for convergence are not satisfied in any

strict sense, Fig. 4 demonstrates that learning is successful in the sense that the walking times increase up to the maximum of 25 time steps after the exploration rate has decayed to a very small value ($\gamma = 0.9$, $\eta = 0.1$ initially).

However, the training times are prohibitively long to be of relevance for any hardware walker. Because of the random actions at the start, this can not even accelerated by tuning the walker more precisely.

4.2 Homeokinesis

The second control method arises from the principle of homeokinesis [9]. It also depends on a quality measure, which is now derived from the deviation of the dynamics of the walker from an internal "world model" stating how a step should look like. This error function

$$E = \frac{1}{2}\left((\Delta\varphi_1)^2 + (\Delta\varphi_2)^2 + (\Delta\omega_1)^2 + (\Delta\omega_2)^2\right) \tag{4}$$

is minimized using the linear controller

$$\Delta K = p_1\Delta\varphi_1 + p_2\Delta\varphi_2 + p_3\Delta\omega_1 + p_4\Delta\omega_2 \; . \tag{5}$$

K is the material constant of a spring implemented at the end of the swing leg, enabling some energy conservation before heelstrike which is released as soon as the swing and stand leg interchange. For a fixed K, this does not interfere much with the dynamics, while changing K can have a regulative effect. Formally, this actuation scheme is equivalent to the abstract control actions used above in $\mathcal{Q}$-learning.

The parameters p_i are adapted by a gradient descent with respect to E. This gradient information can be obtained applying periodic perturbations ϕ_i to the system for each parameter p_i. Integrating over one full period of the perturbations yields the update rule [9]

$$\Delta p_i = \epsilon\phi_i\bar{E} \; . \tag{6}$$

The internal representation from which E is derived is not predefined but generated by the system itself. As a world model, we simply used the entire trajectory of the first step. This yields a continuous control, with E closely related to the deviations Δ. This kind of control is applied to a fixed point specified with a precision that about 40 steps are possible without control. E increases quickly until it gets large enough to induce a noticeable reaction of the controller. After this short critical phase which is used to initiate the parameters p_i, E returns to a very low level, allowing the walker to keep moving for seemingly unlimited times. Noise can be tolerated with strength up to the order of 10^3 E. Note that this adaptation is achieved during the walk without the need for large numbers of trials as with reinforcement learning.

While this shows that homeokinetic control works in principle, the allowed values for deviations and noise appear to small for real walkers. At another test

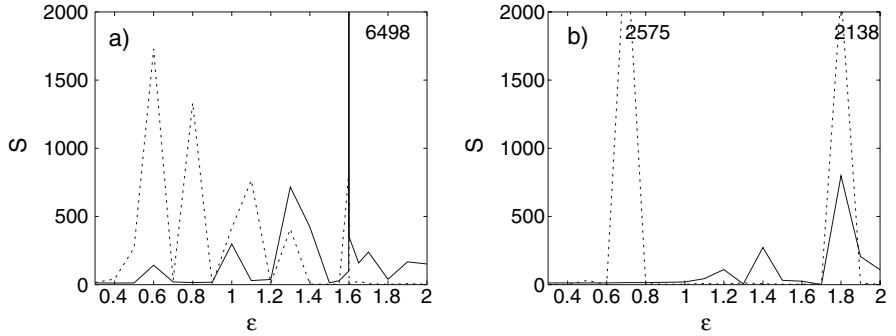


Fig. 5. (**a**) Number of steps as a function of the control parameter ϵ after one trial (*solid line*) and after two trials (*dashed line*) (**b**) The same under influence of gaussian noise of the order of $10^{-6}\,E$

using a parameter set with $\Delta = 10^{-2}$ control turned out to be more difficult. Here, no persistent walking can be achieved in a single run, although the number of steps until failure increases up to several thousands. This increase in performance critically depends on the control parameter ϵ (Fig. 5a). If the initialization of the parameters p_i is performed in a preparatory run, other values of ϵ become prominent, as well as in the presence of noise (Fig. 5b).

If ϵ is set to much smaller values and the walker is reinitialized several times, the parameter are gradually adjusted and stable walking is eventually reached. However, if the environment is noisy, the number of necessary trials is increased and stable walking may again depend on ϵ. So in any case, the control parameter has to chosen carefully and with respect to the expected level of noise in the environment.

In all these examples, the control forces stayed relatively small (1–2% of the total force at maximum). Thus, although its flexibility need to be explored further, the homeokinetic control can be rated as an efficient one. The walker can still rely on its intrinsic dynamics for the most part and control is only meant to be of some assistance in this context.

References

1. McGeer T (1993) Dynamics and control of bipedal locomotion. Journal of Theoretical Biology 163:277–314.
2. Garcia M, Chatterjee A, Ruina A and Coleman M (1998) The simplest walking model: Stability, complexity and scaling. ASME Journal of Biomechanical Engineering, 120(2):281–288.
3. Schwab A, Wisse M (2001) Basin of attraction for the simplest walking model. DETC 2001/VIB-21363.
4. Ott E, Grebogi C, Yorke J (1990) Controlling Chaos. Phys. Rev. Lett. 64:1196–1199.

5. Der R, Herrmann M (1994) Nonlinear Chaos Control by Neural Nets. In: Marinaro M, Morasso P (eds.), Proc. ICANN'94 Springer, London, 1227–1230.
6. Mayer N, Herrmann M, Forough-Nassiraei A, Christaller T (2004) A 2-D Passive Dynamic Walker in Theory and Experiment. Proc. AROB.
7. Cho H-J, Oh S-Y, and Choi D-H (1998) A new population oriented simulated annealing based on local temperature concept. ICEC, Proc. IEEE World Congress on Computational Intelligence, 598–602.
8. Watkins C (1992) $\mathcal{Q}$-learning. Machine Learning 8:279–292.
9. Der R, Steinmetz U, Pasemann F (1999) Homeokinesis – A new principle to back up evolution with learning. In: Mohammadian M (ed.) Computational Intelligence for Modelling, Control, and Automation. IOS Press, Concurrent Systems Engineering Series 55:43–47.
10. Der R, Herrmann M, Liebscher R (2002) Homeokinetic approach to autonomous learning in mobile robots. In: R Dillmann, R D Schraft, H. Wörn: Robotik 2002. VDI-Berichte 1679:301–306.

The Passivity Paradigm in the Control of Bipedal Robots

Mark W. Spong*

Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W. Main St., Urbana, IL 61801
mspong@uiuc.edu

Summary. The concepts of *Passivity* and *Passivity-Based Control* are well established in control theory and provide powerful design tools for control. In this paper we give an overview of a remarkable relation between Passivity-Based Control and the notion of *Passive Walking* in bipedal locomotion. We show how passivity-based control theory can be used to design nonlinear control laws for bipedal robots that reproduce passive limit cycles independent of the ground slope while overcoming some of the limitations of passive gaits, such as extreme sensitivity to ground slope, small basins of attraction, and poor disturbance rejection.

1 Introduction

Several researchers have studied passive walking in planar and 3-D mechanisms, with and without knees, and analyzed their passive gaits [3, 5, 6, 11]. The stable passive gaits found in these mechanisms typically exist only for very shallow slopes and exhibit extreme sensitivity to slope magnitude. The first results in active feedback control in this context appeared in [5, 18]. In [18] it was shown, for a fully actuated compass gait biped, that the passive limit cycle of [5] can be made slope invariant using a so-called potential shaping nonlinear control. These results were extended to the general case of an n-DOF biped in 3-D in [19]. These results rely on certain symmetry properties of the Lagrangian dynamics and impact dynamics under the action of $\mathcal{SO}(3)$.

This papers continues our previous work and provides a unification of the ideas of passive walking, passivity based control, and energy shaping in hybrid systems. We will show how the notion of passivity-based nonlinear control can be used not only to remove the sensitivity to ground slope in the passive limit cycles but also increase the basin of attraction, improve robustness to

*The results in this paper are based on earlier joint work with Francesco Bullo and Gagandeep Bhatia. This research was partially supported by NSF Grants ECS-0128656, CCR-0209202, and CSM-0100162

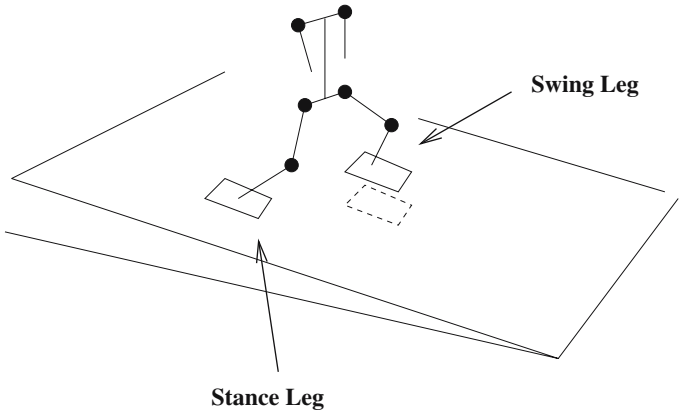


Fig. 1. A General 3-D Biped

disturbance rejection, and improve the transient performance in the sense of increasing speed of convergence to the limit cycle.

2 Background

We consider a general n-degree-of-freedom biped in 3-dimensions. The act of walking involves both a swing phase and a stance phase for each leg, impacts between the swing leg and ground, and possibly internal impacts, such as knee strikes. We make the standard assumptions, namely,

(i) impacts are perfectly inelastic (no bounce),
(ii) transfer of support between swing and stance legs is instantaneous,
(iii) there is no slipping at the stance leg ground contact.

Under these assumptions each impact results in an instantaneous jump in velocities, hence a discontinuity in kinetic energy. The position variables are continuous through the impact and so, if the kinetic energy dissipated during the impact is somehow compensated so that the joint angles and velocities after impact are restored to their original values at the beginning of the step, then a periodic gait (limit cycle) results. In passive walking this is achieved by starting the biped on a constant downhill slope so that that loss of kinetic energy is compensated by the change in potential energy during the step. The loss of kinetic energy can also be compensated by active control of actuators at the joints so that walking can be achieved on level ground and/or uphill.

Let $h\colon Q \to \mathbb{R}^\nu$ be a smooth function defining the foot/ground contact constraint. For bipeds with point foot contact, the dimension, ν, is two in the planar $2D$ case and three in the general $3D$ case. For bipeds with extended feet, ν is three for planar bipeds and six in the most general case. The function

h can be computed using the forward kinematic equations of the robot. We assume that the foot/ground impact takes place precisely when $h = 0$.

The change in velocity at impact is found by integrating the Euler-Lagrange equations over the (infinitesimally small) duration of the impact:

$$\left.\frac{\partial \mathcal{L}}{\partial \dot{q}}\right|_{t^-}^{t^+} = \int_{t^-}^{t^+} F(q,t)dt \tag{1}$$

where $F(q,t)$ represents the contact force over the impact event $[t^-, t^+]$. Because

$$\frac{\partial \mathcal{L}}{\partial \dot{q}} = \frac{\partial \mathcal{K}}{\partial \dot{q}} = M(q)\dot{q}$$

we conclude $\dot{q}(t^+) - \dot{q}(t^-) = M(q)^{-1}\int_{t^-}^{t^+} F(q,t)dt$. Second, we note that the contact force F is aligned with the constraint directions $dh_1, \ldots, dh_\nu$ so that there exist a function $f(t) = (f_1(t), \ldots, f_\nu(t))$ such that $F(q,t) = \sum_{i=1}^{\nu} f_i(t)dh_i(q)$.[1] Thus

$$\dot{q}(t^-) = \dot{q}(t^+) - M(q)^{-1}\int_{t^-}^{t^+} \left(\sum_{i=1}^{\nu} f_i(t)dh_i(q)\right) dt\ . \tag{2}$$

Third, since the evolution during the impact interval $[t^-, t^+]$ of the ideal plastic impact satisfies $h(q) = 0$, each component $h_i(q)$ satisfies

$$0 = \dot{h}_i(q) = dh_i \cdot \dot{q}(t^+) \quad \text{for } i = 1, \ldots, \nu \tag{3}$$

which means geometrically that $\dot{q}(t^+)$ is perpendicular with respect to the M-inner product to the vectors $M(q)^{-1}dh_i$. This fact show that the right-hand side of (2) is an orthogonal sum and therefore $\dot{q}(t^+)$ equals the $M(q)$-orthogonal projection of $\dot{q}(t^-)$ onto the feasible space $\{v \in T_qQ |\ dh_i(q) \cdot v = 0\ i = 1, \ldots, \nu\}$. In summary, the impact dynamics may be represented as

$$\dot{q}(t^+) = P_q(\dot{q}(t^-))\ , \tag{4}$$

where the *plastic projection* P_q for the impact occurring at $h(q) = 0$ is the $M(q)$-orthogonal projection of $\dot{q}(t^-)$ onto $\{v \in T_qQ |\ dh_i(q) \cdot v = 0\ i = 1, \ldots, \nu\}$. The dynamics of the controlled biped are therefore described as a hybrid dynamical system consisting of the usual Euler-Lagrange equations together with impact equations

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) &= u, && \text{for } h(q) \neq 0 \\ q(t^+) &= q(t^-) \\ \dot{q}(t^+) &= P_q(\dot{q}(t^-)) && \text{for } h(q) = 0\ . \end{aligned} \tag{5}$$

[1] This is a standard fact from constrained Lagrangian systems [10].

2.1 Group Actions and Symmetry

We first review some basic background from differential geometry.

Definition 1. *Let Q be a smooth manifold and G be a Lie group. A* left action *of G on Q is a map $\Phi\colon G \times Q \to Q$ taking a pair (g, q) to $\Phi(g, q) = \Phi_g(q) \in Q$ and satisfying for all $q \in Q$*

(i) $\Phi_e(q) = q$, where e is the identity element of G, and
(ii) $\Phi_{g_1}(\Phi_{g_2}(q)) = \Phi_{g_1 g_2}(q)$.

Definition 2. *Let T_qQ be the linear space of tangent vectors at $q \in Q$, and let $TQ = \bigcup_q T_qQ$ be the tangent bundle of Q. If Φ is a group action on Q, we let $T_q\Phi_g$ denote the tangent function to Φ_g mapping T_qQ onto $T_{\Phi_g(q)}Q$. This defines a mapping $T\Phi : G \times TQ \to TQ$ which is called the* Lifted Action.

A group action on Q thus induces, in a natural way, a corresponding action on TQ. Likewise a group action induces corresponding maps on other quantities, such as scalar functions over Q, one-forms, and covector fields. For example, if $h : Q \to \mathbb{R}$ is a scalar function on Q, then the group action induces a map via composition, $(h \circ \Phi)(q) := h(\Phi(q))$. These induced maps are important to determine how vector fields and their associated flows are affected by the group action.

2.2 Invariance and Equivariance

The notions of *Invariance* and *Equivariance* under group actions are crucial for what follows.

Definition 3. *Let $F : M \to N$ be a smooth mapping between manifolds M and N and let $\Phi : G \times M \to M$ be an action of the Lie Group G on M. Then we say that*

(i) F is Invariant *under the group action if $F \circ \Phi = F$, i.e., if, for all $g \in G$ and $m \in M$*

$$(F \circ \Phi_g)(m) = F(m) \, .$$

(ii) F is Equivariant *with respect to G if there exists an associated Lie Group $\tilde{G}$ and a group action $\tilde{\Phi} : \tilde{G} \times N \to N$ such that $F \circ \Phi = \tilde{\Phi} \circ F$ in the sense that for all $g \in G$ there exists $\tilde{g} \in \tilde{G}$ such that*

$$(F \circ \Phi_g)(m) = (\tilde{\Phi}_{\tilde{g}} \circ F)(m) \text{ for all } m \in M$$

Invariance is then seen to be a special case of equivariance corresponding to the choice $\tilde{\Phi} = I$, the identity transformation. For a vector field X, considered as a mapping from $Q \to TQ$, equivariance means that for all $g \in G$ and $q \in Q$

$$X(\Phi_g(q)) = T_q\Phi_g(X(q)) \, .$$

Similarly, a covector field α on Q is *equivariant* if, for all $g \in G$ and $q \in Q$

$$\alpha(\Phi_g(q)) = T_q^* \Phi_g(\alpha(q)) \ .$$

In addition, one can show that if a function $f\colon Q \to \mathbb{R}$ is invariant (respectively, equivariant), then so is its differential df. The importance of these definitions for us is the following result.

Lemma 1. *Let the vector field X be Φ-equivariant, and let $\gamma : [0,T] \to Q$ be an integral curve of X, i.e., the solution of the differential equation defined by X with initial condition $\gamma(0)$. Then, for all $g \in G$, the map $\Phi_g \circ \gamma : [0,T] \to Q$ is an integral curve for X.*

Proof: See, for example, [15]

2.3 Lagrangian Symmetry

The relevance of the above notions of group action invariance to the problem of bipedal locomotion was outlined in [19]. It was shown that changing the ground slope defines a group action on the configuration manifold of the biped. This group action is induced by the usual action of $\mathcal{SO}(3)$ on R^n and is denoted by Φ_A, where $A \in \mathcal{SO}(3)$.

It was shown in [19] that both the kinetic energy and impact equations are invariant under this group action. As a consequence, let

$$g_{\Phi_A}(q) = \frac{\partial}{\partial q} \mathcal{V} \circ \Phi_A(q) \tag{6}$$

where $\mathcal{V}(q)$ is the biped potential energy. Then it follows from Lemma 1 that

Proposition 1. *Solution trajectories of the system*

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) &= 0, && \text{for } h(q) \neq 0 \\ q(t^+) &= q(t^-) \\ \dot{q}(t^+) &= P_q(\dot{q}(t^-)) && \text{for } h(q) = 0 \ . \end{aligned} \tag{7}$$

are bijectively mapped to solution trajectories of the system

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g_{\Phi_A}(q) &= 0 \ , && \text{for } h(q) \neq 0 \\ q(t^+) &= q(t^-) \\ \dot{q}(t^+) &= P_q(\dot{q}(t^-)) && \text{for } h(q) = 0 \ . \end{aligned} \tag{8}$$

through the group action Φ.

Moreover, as we show in the next section, in the case of an actuated biped these two systems are feedback equivalent through passivity based nonlinear control showing that passive gaits can be preserved on arbitrary ground slopes.

3 Passivity Based Control

Passivity in control systems is related to energy and Lyapunov methods. Consider a nonlinear system of the form

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i = f(x) + G(x)u \tag{9}$$

$$y = h(x) \tag{10}$$

where $x \in R^n$ is the state vector, $u \in R^m$ is the control input, and $y \in R^m$ is the output. The vector fields f, $g_1, \ldots g_m$, and h are smooth and we assume that the system has the same number of inputs and outputs. We assume that $f(0) = 0$ so that the origin in R^n is an equilibrium of the open loop system.

Definition 4. *The system (9) is said to be (Input/Output) Passive if there is a nonnegative scalar function $S : R^n \mapsto R$, called a Storage Function, such that*

$$\dot{S} \leq y^T u \tag{11}$$

An important property of a passive system is the following:

Lemma 2. *Suppose that the system Σ is passive with nonnegative storage function S. Choose an output feedback control*

$$u = \gamma(y(t)) \tag{12}$$

where γ is any continuous function satisfying $y^T\gamma(y) \leq 0$. Then the origin $x = 0$ is stable in the sense of Lyapunov. In addition, if the system satisfies a uniform observability condition with respect to the output y, then $x = 0$ is asymptotically stable.

In the case of bipedal locomotion, one is concerned with stability relative to limit cycles rather than isolated equilibrium points. This is known as *Orbital Stability* [8]. Due to space restrictions we can only mention in passing that the above definition and lemma can be extended to the case where the storage function S is zero on the limit cycle and positive in a neighborhood of the limit cycle [8]. In addition, similar notions can be stated for hybrid systems of the form (5).

The relationship between passive walking and Passivity-Based Nonlinear Control is the following. where $\mathcal{K} = \frac{1}{2}\dot{q}^T M(q)\dot{q}$, $\mathcal{V}$ are the kinetic and potential energies of the biped. We define a so-called *Storage Function* $\mathcal{S}$ as

$$\mathcal{S} = \frac{1}{2}(E_{\Phi_A}(q, \dot{q}) - E_{ref})^2 \tag{13}$$

where $E_{\Phi_A}(q, \dot{q})$ is the total energy

$$E_{\Phi_A}(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \mathcal{V} \circ \Phi_A(q) \tag{14}$$

and E_{ref} is a reference value for the energy. We assume here that E_{ref} is constant. The case of nonconstant reference energy can also be treated [1]. Then it follows from the usual skew symmetry property of robot dynamics [20] that, along trajectories of the system

$$\dot{E}_{\phi_A} = \dot{q}^T \left(u - \frac{\partial \mathcal{V}}{\partial q} + \frac{\partial(\mathcal{V} \circ \Phi_A)}{\partial q} \right) \tag{15}$$

If we set

$$u = \frac{\partial \mathcal{V}}{\partial q} - \frac{\partial(\mathcal{V} \circ \Phi_A)}{\partial q} + \bar{u} \tag{16}$$

where $\bar{u}$ is an additional control to be designed, then the derivative of the storage function satisfies

$$\dot{\mathcal{S}} = (E_{\Phi_A} - E_{ref})\dot{E}_{\Phi_A} \tag{17}$$

$$= (E_{\Phi_A} - E_{ref})\dot{q}^T \bar{u} \tag{18}$$

Note that for $\bar{u} = 0$, the control u transforms the system (7) into the system (8 and hence can be used to render any passive limit cycle slope invariant. If we now choose $\bar{u}$ according to

$$\bar{u} = -k(E_{\Phi_A} - E_{ref})\dot{q} \tag{19}$$

then we obtain

$$\dot{S} = -2k||\dot{q}||^2 S \leq 0 \tag{20}$$

Note that $\bar{u}$ vanishes on the limit cycle and hence does not affect the slope invariance property derived with $\bar{u} = 0$. The affect of the term $\bar{u}$ is to influence the dynamics transverse to the limit cycle and, as shown below, drive the energy exponentially to the reference energy.

Proposition 2. *Suppose there is a constant $\alpha > 0$ such that $||\dot{q}(t)||^2 \geq \alpha$ for all $0 \leq t < T_1$, where T_1 is the time of first impact of the swing foot with the ground. Then*

$$S(t) \leq S(0)e^{-2k\alpha t} \quad \text{for } 0 \leq t < T_1 \tag{21}$$

Proof: Under the above assumption

$$S(t) \leq -2k||\dot{q}||^2 S \leq -2\alpha S \tag{22}$$

and so the result follows from the Gronwall-Bellman Lemma [8].

Under these conditions the total energy of the biped will thus converge exponentially toward the reference energy between impacts. At the impact,

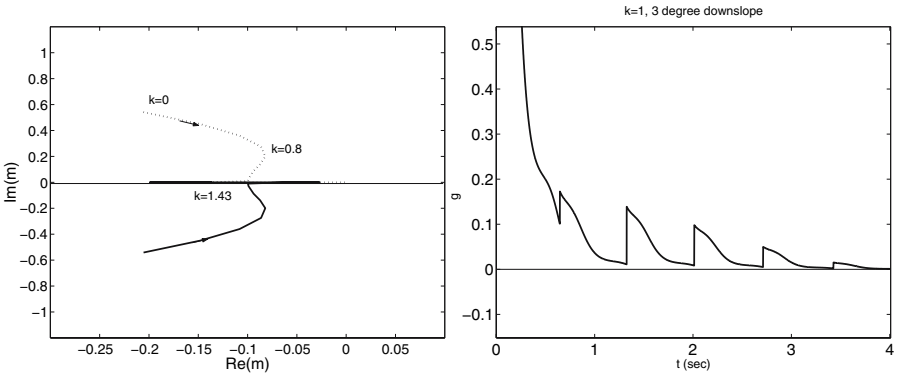


Fig. 2. Eigenvalues of Linearized Poincaré map as a function of the gain k (*left*) and value of the Storage Function S (*right*)

the storage function will exhibit a jump discontinuity. It follows from standard results in hybrid system theory [2] that, if the value of $S(t)$ at a jump discontinuity is less than its value at the previous jump, then the total energy will converge asymptotically to the reference energy as $t \to \infty$. A sufficient condition for local convergence to the origin of the storage function S is that the eigenvalues of the Poincaré map lie inside the unit circle (see Fig. 2).

4 Simulations

We have simulated these results on several planar systems, including a compass gait biped, a biped with knees, and a biped with knees and torso [1]. Here we present simulations for the compass gait for illustrative purposes. Figure 2 shows a root locus of the eigenvalues of the linearized Poincaré map as a function of the gain k in the control law (19). As the gain increases from zero, the eigenvalues, which are inside the unit disk, move toward the origin. The implication of this is that the trajectory after each step moves closer to the limit cycle on which the energy equals the reference energy, E_r. Therefore the value of the storage function, S, will decrease at each step. This is indeed confirmed by Fig. 2 showing the storage function for $k = 1$. We also notice that for large gain, the locus begins to move toward the boundary of the unit disk. This suggests that there is an "optimal" choice for k beyond which the stability is degraded. This is born out by simulations [1].

4.1 Basin of Attraction and Rate of Convergence

The *Basin of Atraction* of the limit cycle is the set of initial conditions in state space such that trajectories starting at these initial conditions converge asymptotically to the limit cycle. Since the state space is four dimensional and

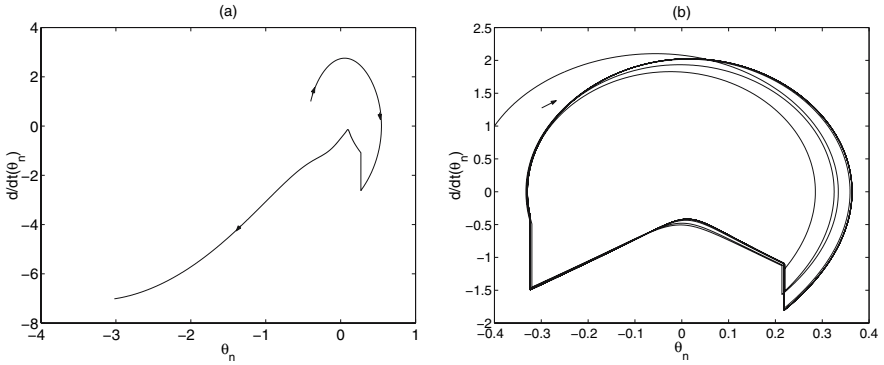


Fig. 3. Illustration of increased basin of attraction (**a**) without total energy control (**b**) with total energy control

the equations of motion highly nonlinear, the basin of attraction is difficult to characterize precisely. One generally finds through numerical techniques that the basin of attraction is small, consisting of a narrow band of initial conditions around the limit cycle where trajectories converge. One may consider the relative stability of the limit cycle by looking at the eigenvalues of the linearized Poincaré map. The further away from the unit circle the eigenvalues are, the more robust the limit cycle is to disturbances and unmodeled dynamics and the quicker the trajectory will converge to the limit cycle. Convergence to the limit cycle can also be characterized by the number of steps required to converge to within a given error band around the limit cycle.

Figures 3 and 4 show trajectories with and without the total energy control. In Fig. 3 a point previously outside the basin of attraction is captured by the addition of the total energy control. While the precise change in the

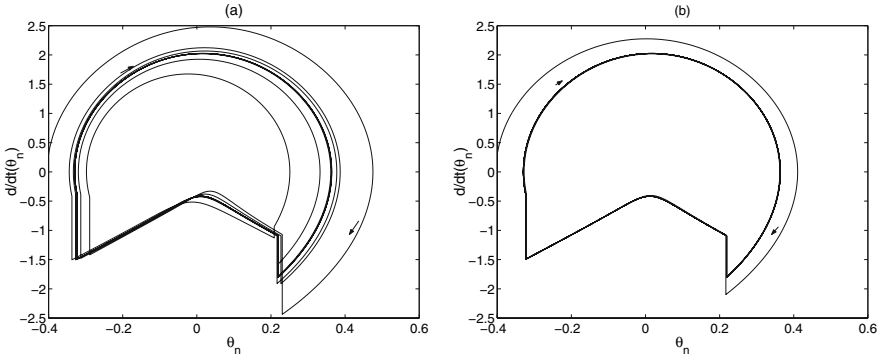


Fig. 4. Convergence to the limit cycle (**a**) without total energy control (**b**) with total energy control

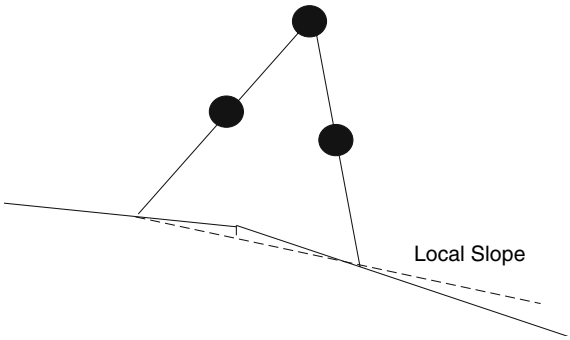


Fig. 5. Slope change from 3 degree down slope to 8 degree down slope

basin of attraction is again difficult to characterize, the total energy control has increased the size of the basin for a range of gains, k. Figure 4 shows that, even for initial conditions inside the basin of attraction of the system without total energy control, the addition of total energy shaping can increase convergence to the limit cycle. We have found in our simulations that, with total energy control, the biped trajectory converges to the limit cycle in one to three steps depending on the initial conditions whereas without the total energy control convergence is much slower, on the order of ten to twelve steps.

4.2 Slope Variation and External Disturbances

As a further illustration of the robustness enhancement of the total energy shaping, we show the performance of the system when the slope exhibits a sudden change and when the robot is subject to a constant disturbance torque. The control input is always determined by the so-called *local slope*, which is the ground slope at the stance leg. The local slope can be determined by the two-point contact condition which occurs at the moment of contact of the swing foot with the ground as shown in Fig. 5. Thus for a discrete slope change there are generically two steps during which an error in the perceived ground slope occurs. Figure 6(a) shows that, without the total energy control, the robot is not able to maintain a stable gait. Figure 6(b) shows that, with the addition of the total energy based control $\bar{u}$, the biped successfully makes the transition between slopes.

Figure 7 shows the effect of a 1-Nm disturbance torque added to each joint of the biped. This disturbance is sufficient to cause the biped to fall without the total energy shaping control whereas stability is maintained when the total energy shaping control is used

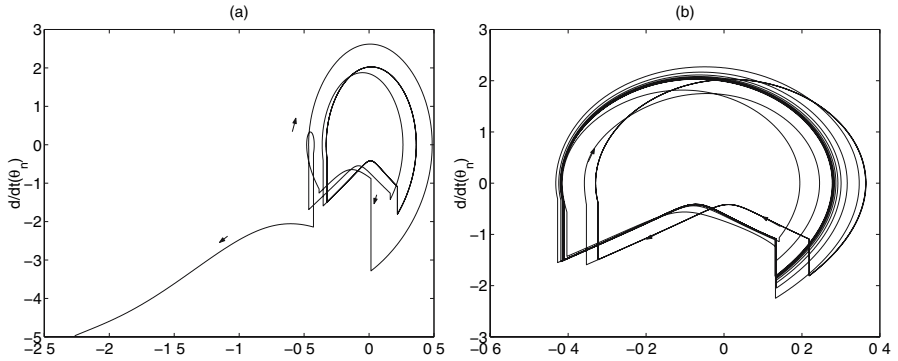


Fig. 6. Closed loop trajectory on terrain with slope change from 3 degree down slope to 8 degree down slope. (**a**) without total energy shaping; (**b**) with total energy shaping

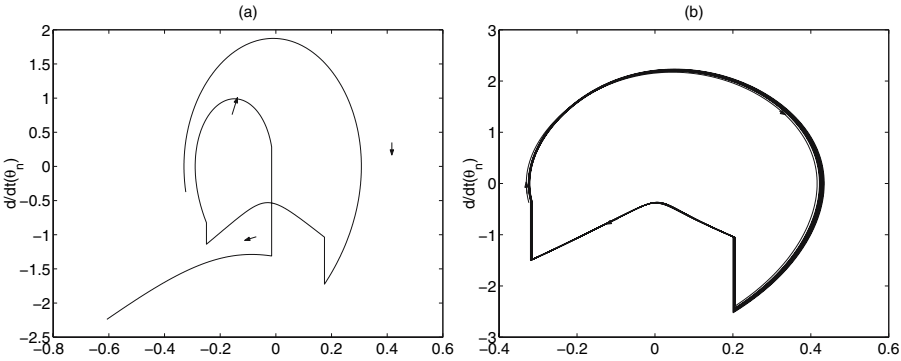


Fig. 7. Trajectory with a 1-Nm disturbance torque at each joint (**a**) without total energy shaping (**b**) with total energy shaping

References

1. Bhatia, G., *Passivity Based Control of Biped Robots*, Thesis, M.S. Department of Nuclear, Plasma, and Radiological Engineering, University of Illinois, December, 2002.
2. Branicky, M.S., "Stability of Hybrid Systems: State of the Art," *Proc. IEEE Conf. on Decision and Control*, San Diego, CA, December, 1997.
3. Collins, S.H., Wisse, M. and Ruina, A. "A 3-d Passive-Dynamic Walking Robot with Two Legs and Knees," *Int. J. Robotics Research*, Vol. 20, No. 7, July, 2001.
4. Garcia, M., et al., "The Simplest Walking Model: Stability, Complexity, and Scaling," *ASME Journal of Biomechanical Engineering*, April, 1998.
5. Goswami, A., Espiau, B., and Keramane, A. (1997), Limit cycles in a passive compass gait biped and passivity-mimicking control laws. *Journal of Autonomous Robots*, Vol. 4, No. 3.

6. Goswami, A., Thuilot, B., and B. Espiau, B. (1998), A Study of the Passive Gait of a Compass-like Biped Robot: Symmetry and Chaos, *International Journal of Robotics Research*, Vol. 17, No. 15.
7. Hiskens, I.A. "Stability of hybrid system limit cycles: application to the compass gait biped robot," *Proceedings of the 40th IEEE Conference on Decision and Control*, pp. 774–779, Orlando, Fl., Dec. 2001.
8. Khalil, H.K., *Nonlinear Systems*, Second Edition, Prentice Hall, Englewood Cliffs, NJ, 1995.
9. Kuo, A.D., "Stabilization of Lateral Motion in Passive Dynamic Walking," *International Journal of Robotics Research*, Vol. 18, No. 9, pp. 917–30, 1999.
10. Marsden, J.E. and T.S. Ratiu, *Introduction to Mechanics and Symmetry*. second ed. Springer Verlag. New York, NY, 1999.
11. McGeer, T., "Passive Dynamic Walking," *International Journal of Robotics Research*, 1990.
12. *Webster's New World Dictionary*, Warner Books, New York, NY, 1990.
13. McGeer, T. Dynamics and control of bipedal locomotion. *Journal of Theoretical Biology*, 163(3):277, August 1993.
14. McMahon, T.A. *Muscles, Reflexes, and Locomotion*. Princeton University Press, Princeton, New Jersey, 1984.
15. Olver, P.J., "Application of Lie Groups to Differential Equations," Graduate Texts in Mathematics, Vol. 107, Springer-Verlag, New York, 1993.
16. Parker T.S. and Chua, L.O. *Practical Numerical Algorithms for Chaotic Systems*, Springer-Verlag, New York, NY, 1989.
17. Spong M.W. and Praly, L. Energy based control of underactuated mechanical systems. In A.S. Morse, editor, *Control Using Logic-Based Switching*, pp. 162–172. Springer-Verlag, Berlin, 1996.
18. Spong, M.W. "Passivity Based Control of the Compass Gait Biped," In: *IFAC World Congress*, Vol. B., pp. 19–24, Beijing, China, July, 5–9, 1999.
19. Spong, M.W. and Bullo, F. "Controlled Symmetries and Passive Walking," *IFAC World Congress*, Barcelona, Spain, July, 2002.
20. Spong, M.W. and Vidyasagar, M. *Robot Dynamics and Control*, John Wiley & Sons, Inc., New York, 1989.

Ankle Joints and Flat Feet in Dynamic Walking

Daan G.E. Hobbelen and Martijn Wisse

Delft Bio-robotics Laboratory, Delft University of Technology, The Netherlands
d.g.e.hobbelen@wbmt.tudelft.nl

1 Introduction

One of the main concerns in current bipedal robot design is the high system complexity in most state of the art walking robots [4, 6, 8]. This high complexity is caused by the high amount of degrees of freedom in these systems, the multitude of actuators and sensors and the high level of control being applied to these robots. In contrast, the concept of 'passive dynamic walking', as introduced by McGeer [7], can well be used to make efficient and robust bipedal robots with a much simpler mechanical design. The concept shows that a walker with two legs can perform stable walking when it is placed on a shallow slope, without the need for actuation or control. A cyclic motion is obtained by utilizing the natural dynamics of the system, being the passive pendulum motion of the swing leg and the inverted pendulum motion of the stance leg.

In recent years the concept of passive dynamic walking has been used as a starting point for designing actuated dynamic walkers that are able to walk on a flat surface, one of which is the robot 'Max' [10] (shown in Fig. 1a.) that was developed at the Delft Biorobotics Laboratory. Max is equipped with arced feet, similar to most of the current passive dynamic walkers in the world [1, 2, 7], as well as to the actuated dynamic walkers based on this concept [9, 10]. This foot design is widely used because of its simplicity and the fact that it limits the vertical excursions of the walkers' center of gravity and thus is beneficial to both the efficiency and the robustness of the walkers [12].

In this paper, we replace the arced feet by flat feet that are connected to the lower legs of the robot through an ankle joint. The energy fluctuations due to the increasing vertical excursions of the center of gravity can be compensated by storing energy in a spring in the ankle joint. This new design gives the opportunity to make further improvements with beneficial effects on the robot's performance. The main four potential advantages of this new design are:

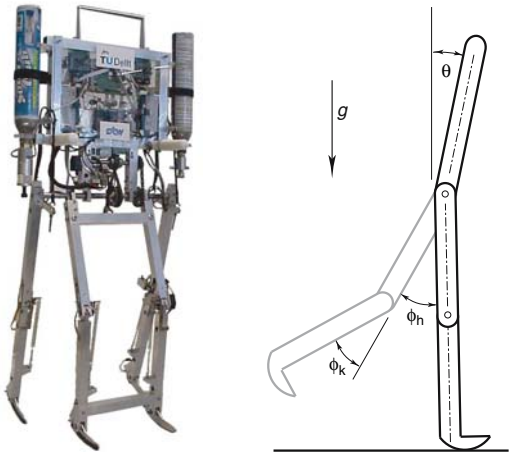


Fig. 1. (**a**) Walking robot Max; 2D robot based on the principle of passive dynamic walking. (**b**) 5-link simulation model of Max with arced feet

Increasing energy efficiency. Kuo [5] showed that an impulsive ankle push-off just before heel strike can significantly improve the walking energetics, because the collision loss at heel strike is decreased. The ankle joints allow the addition of this type of actuation.

Increasing versatility. The flat feet create the possibility for a dynamic walker to stand still with both legs next to each other and to switch between walking and standing still. This can significantly increase the applicability of this type of walkers.

Increasing resistance to yaw in 3D. One of the challenges in dynamic walking robot research is to obtain stability in 3-D on only two legs. Collins et al. [1] experienced that the counter-swinging of the two legs induces yaw (rotation about the vertical axis), an undesirable effect. Flat feet have a larger contact area with the floor than arced feet, which will increase the resistance to yaw.

Increasing robustness to variable floor properties Since a flat foot will not be moving relative to the floor for most of the stance phase, it is expected that the walker will be less sensitive to for instance floor irregularities or hardness.

The first two potential advantages require actuated ankle joints, the last two can already be obtained with passive ankle joints. This paper focuses on the introduction of passive ankle joints to the concept of (passive) dynamic walking. The main objective is to find how the design parameters affect the efficiency and robustness of the walker and which parameters are most influential. Section 2 presents the performance criteria used to assess the performance of the new design. Section 3 describes the simulation model

used in this research. The results are presented in Sect. 4, followed by the conclusions in Sect. 5.

2 Performance Criteria

To evaluate the robot's performance while varying the design parameters, it is essential to define specific criteria that sufficiently describe whether the robot's operation is satisfactory. The most important factors to judge the robot's performance are efficiency and robustness. In addition, to achieve resistance to yaw and increase the robustness to variable floor properties it is also important to have a long period of full foot-ground contact. Three main performance criteria have been specified for this study, as listed below:

Efficiency To judge the efficiency of the robot design a new measure will be introduced, which we will call the efficiency criterion H. The efficiency criterion H builds on the 'specific resistance' η (energy consumption per weight per distance traveled) which is commonly used to assess the efficiency of a bipedal robot, as well as of human beings. However, the specific resistance is highly dependent on the walker's speed and thus identical designs will be judged differently when walking at different speeds. To find an efficiency measure that is independent of speed, a fundamental relation between walking speed and energy loss has to be found. Arguing that foot impact causes the only fundamental (and inevitable) form of energy loss in dynamic walking, we have found such a relation and based a new efficiency measure upon it. This measure has been defined using the simplest walking model [3], which solely exhibits an energy loss at foot impact. The new efficiency criterion H is defined as:

$$H = \frac{2 \cdot \eta}{Fr^2 \cdot l_{norm}} \cdot \left(\frac{\tanh\left(l_{norm}/2Fr\right)}{l_{norm}/2Fr} \right)^2 \tag{1}$$

where:
η is the specific resistance [-]
Fr is the Froude number ($Fr = v/\sqrt{gl}$) [-]
l_{norm} is the walker's normalized step length ($l_{norm} = l_{step}/l$) [-]
v is the walker's speed [m/s]
g is the gravitational acceleration [m/s^2]
l is the maximal height of the walker's center of gravity [m]
l_{step} is the walker's step length [m]

The second term in (1), including the tanh function, follows from an approximation of the nonlinear inverted pendulum motion that characterizes a dynamic walker. The efficiency criterion H is always equal to one (within 5% due to approximations) in case of the simplest walking model. A walker with a lower value of H will be judged as having a more energy efficient

design and a higher value of H means a less efficient design. Max has an H-value of 7.4.

Robustness The robustness of a walker is the ability to continue walking under the influence of various perturbations. The robustness can be quantified by finding the maximum size of the perturbation for which the walker can maintain its stability. The Basin of Attraction (BoA) is often used as an indication of the robot's robustness. The BoA is the set of initial conditions for a step (starts just after heel strike) from which the robot can converge to a limit cycle walking motion and thus maintain stability. The size of the Basin of Attraction is hard to quantify, since it has a complex shape in multiple dimensions (in the particular case of this paper five initial conditions at the start of a step have to be defined). Calculating the total shape takes much computing time which is undesired given the iterative character of the design process. Because of this, in this research an approximation of the size of the Basin of Attraction is used. Firstly the search for the boundaries of the BoA was limited to a search along the principal axes of the search space, examining the variation of one separate initial condition at a time. Secondly it was established which of the initial conditions allowed the least amount of variation and thus define the smallest cross section of the Basin of Attraction. These initial conditions will be the ones that are most likely to cause instability and falling and thus are the best measure for the robustness of the robot. Figure 2 illustrates an example of this robustness measure in practice.

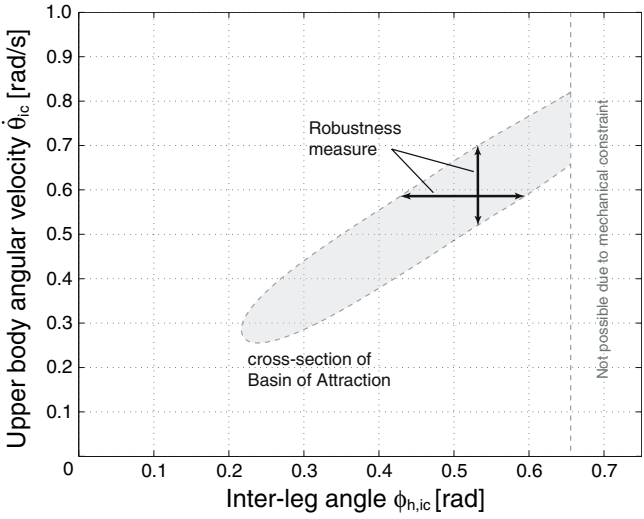


Fig. 2. Illustration of how the robustness measure gives an indication of the size of the Basin of Attraction

Period of full foot contact To get resistance to yaw and increase the robustness to variable floor properties it is essential to have the foot firmly placed on the ground. This criterium checks for what percentage of the time one of the robot's feet is in full contact with the floor and thus is not moving relative to the ground. In case of Max' arced feet, there is no moment in time during walking in which the feet are not moving relative to the ground and therefore there is no full foot contact period. This criterium has a value of zero for Max.

3 Simulation

By means of dynamic simulation, we studied how the foot design parameters influence the performance, before building a physical prototype. The simulations are done in MATLAB, based on an example simulation by Wisse [11]. The procedure contains numerical simulation of the continuous dynamic motion of the robot alternated with discrete impact calculations at the instants when events like heel, knee and toe strike are detected.

3.1 Main Model Assumptions

The main assumptions that were made for the simulations are the following:

- All body links are assumed infinitely stiff.
- Joints are free of damping and friction.
- Heel, knee and toe impacts are fully inelastic.
- Contact between foot and floor is without slip.
- Heel and toe are released from ground when the contact force becomes tensile.
- Floor is rigid and perfectly flat.
- No rolling damping is present between foot and floor.

3.2 Preceding Model of Max

The basis of the simulations is a two-dimensional 5-link model of the existing prototype robot Max, as depicted in Fig. 1b. The model consists of an upper body, two upper legs and two lower legs. There are three joints present in the model; the hip joint and two knee joints. In the hip joint a kinematic constraint is present that keeps the upper body at the intermediate angle of the two upper legs [10]. In the knee joint a constraint is present which fixes the lower leg to the upper leg when they are aligned. This models a latch mechanism that is present in the prototype.

The actual prototype is actuated in the hip joint by McKibben air muscles. In the simulation model these muscles are approximated by linear spring-dampers with an alternating equilibrium position. Whenever the front foot

hits the floor the equilibrium position of the hip spring is switched such that the hind foot is pulled forward. The energy consumption of the robot is measured by the potential energy that is introduced to the hip spring when changing this equilibrium position.

In recent research this simulation model has been validated to the real prototype [10]. The comparison of the motion of the real prototype and the simulation model is depicted in Fig. 3.

3.3 The New Simulation Model with Flat Feet

As a result of the satisfactory validation of the 5-link model to the prototype, we decided to use this model in predicting the motion of a similar robot

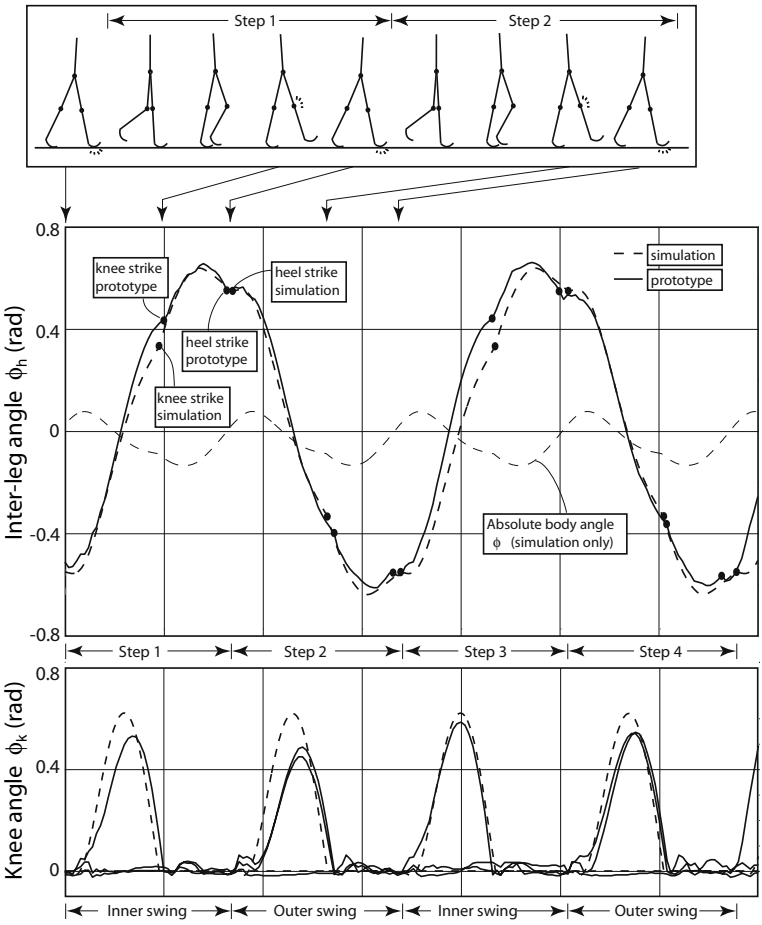


Fig. 3. Comparison of the walking motion of the real prototype and the 5-link simulation model, indicating that this simulation model gives a satisfactory description of the robot's dynamics. This figure is taken from Wisse et al. 2004 [10]

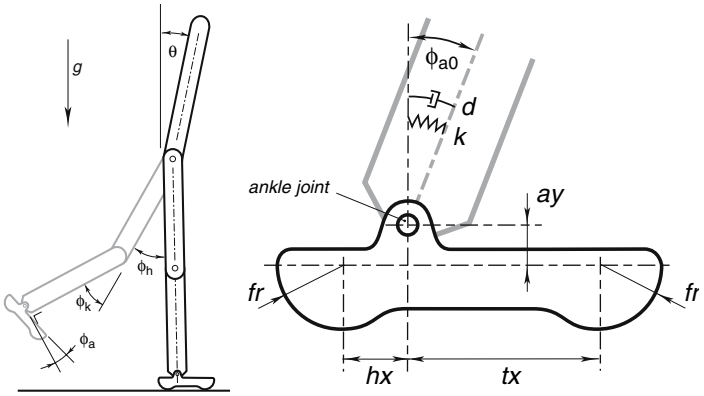


Fig. 4. (**a**) 7-link simulation model with ankle joints and flat feet. (**b**) Foot geometry indicating design parameters that are investigated in this research

with flat feet and ankle joints. The arced feet in the model were replaced by ankle joints and flat feet, while all other model parameters (including actuation) remained the same. The resulting 7-link model is depicted in Fig. 4a. Figure 4b. gives a detailed view of the foot and the design parameters that are investigated in this paper. The ankle joint is aligned with the hip and knee joint when the leg is fully stretched. The heel and toe radius were chosen to be equal. There is a spring-damper combination in the ankle joint with variable stiffness k, damping d and equilibrium position ϕ_{a0}.

4 Results

The simulations have resulted in knowledge on how the performance of the walker changes due to variations in the new foot design parameters. This relation is presented in the next section by showing how the performance measures change when the design parameters are varied around a typical working point in the parameter space.

4.1 Typical Working Point

Table 1 lists the nominal parameter settings of the working point in the parameter space. The effect of changing these parameters on the three main performance criteria is discussed separately.

Efficiency

The efficiency criterion H at the nominal working point has a value of 9.9. Figure 5 gives the deviation from this nominal H-value as a function of

Table 1. Nominal parameter settings for representative working point

Design Parameters		Nominal Parameter Value
Vertical ankle position ay	[mm]	10
Horizontal heel position hx	[mm]	15
Horizontal toe position tx	[mm]	45
Heel and toe radius fr	[mm]	20
Ankle stiffness k	[Nm/rad]	6
Ankle damping d	[Nms/rad]	0.05
Ankle equilibrium position ϕ_{a0}	[deg]	5

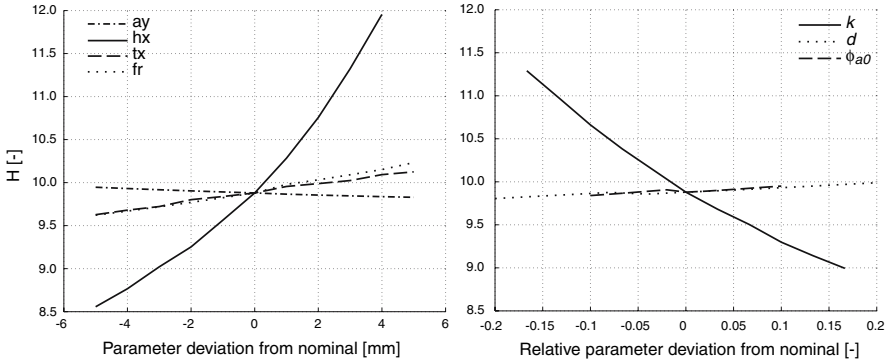


Fig. 5. Variation of efficiency criterion H as a result of varying design parameters, both geometrical parameters (*left*) and ankle parameters (*right*, given in relative parameter changes because of differing units)

parameter variations around the working point. From these graphs it can be concluded that two parameters have a major influence on the efficiency of the robot, being the distance from the ankle joint to the heel hx and the ankle stiffness k. The walker's efficiency is improved by decreasing hx and by increasing the ankle stiffness k.

The reason that increasing hx decreases the efficiency can most probably be found in the fact that increasing hx results in a premature interruption of the forward falling (inverted pendulum) motion of the robot. It is this part of the robot's motion that most effectively creates a forward speed in the robot from the potential energy that is available. Increasing ankle stiffness k increases the potential for storing energy and thus smoothes the robot's motion. This smooth motion results in a smaller impact with the ground and thus increases efficiency.

Table 2. Five initial conditions at the beginning of a step, their nominal value at the working point limit cycle and the allowed variation on these conditions while maintaining stability. The shaded rows give the initial conditions that are most sensitive; those were used to give an indication of the total size of the Basin of Attraction

Initial Condition	Limit Cycle Value	Allowed Variation
Inter-leg angle $\phi_{h,ic}$ [rad]	0.53	0.17
Front foot ankle angle $\phi_{a,ic}$ [rad]	0.083	1.29
Inter-leg angular velocity $\dot{\phi}_{h,ic}$ [rad/s]	0.55	1.22
Upper body angular velocity $\dot{\theta}_{ic}$ [rad/s]	0.59	0.17
Fill percentage air muscle [%]	98	100

Robustness

Table 2 gives the five initial conditions defined in our simulations, their value for the limit cycle in the working point and the range of allowed variation on these conditions. From these numbers it was decided that the inter-leg angle and the upper body angular velocity are the most crucial and sensitive initial conditions for the robot and their allowed variation will be used as an indication of the size of the Basin of Attraction.

Figure 6 shows the sensitivity of these two indicators to change in the design parameters. These graphs show some remarkable jumps, which are caused by the fact that new failure modes arise, which appear to allow less variation on the initial conditions of a step. An example of such a case is the transition from a situation where the robot trips over its toe during the first step to a trip during the fourth step.

Nonetheless some trends can be derived from the four graphs in Fig. 6 and a few can easily be explained. First of all it seems that, similar to the case of efficiency, the heel position hx and the ankle stiffness k are two important factors. Increasing the ankle stiffness has a definite positive effect on robustness. This effect seems similar to the effect of increasing the radius of the rigidly connected arced feet [12]. Further research on this effect will be done in the near future. Although the effect of increasing hx cannot be generalized easily, it shows that a high value for hx drastically reduces the size of the BoA in at least one dimension. Increasing the distance from the ankle joint to the toe (tx) decreases the robustness, simply by the fact that the walker will trip over its toes more easily. A similar effect comes from decreasing the equilibrium angle of the ankle stiffness ϕ_{a0}. Finally, parameters fr, ay and d show little effect on robustness and the small effects that are noticeable cannot easily be explained.

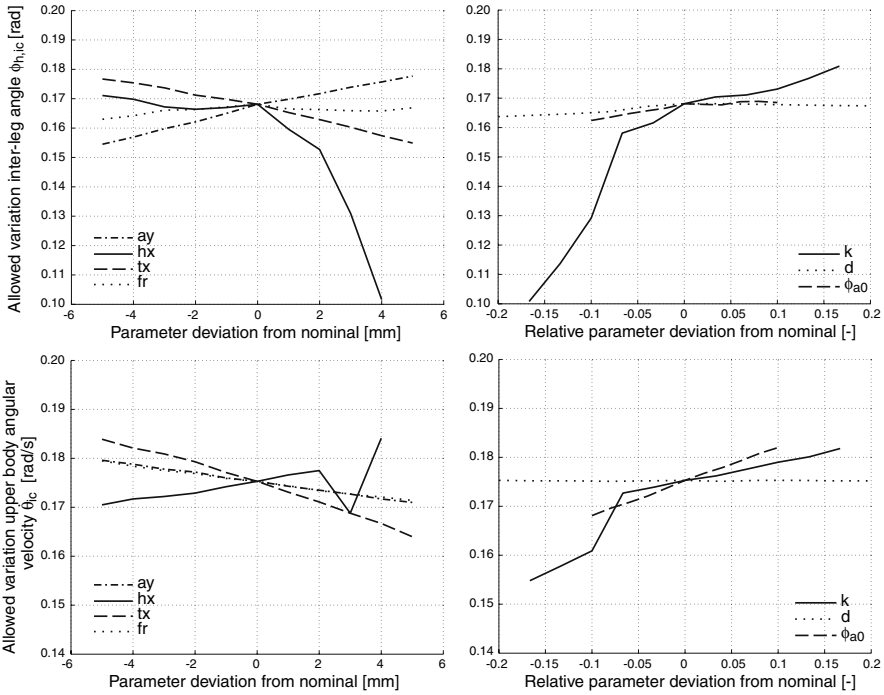


Fig. 6. Variation of two Basin of Attraction size indicators due to varying design parameters, both geometrical parameters (*left*) and ankle parameters (*right*)

Period of Full Foot Contact

At the nominal working point there is full contact between a foot and the floor for 89.5 percent of the time. Figure 7 gives the deviation from this nominal full foot contact period as a function of parameter variations around the working point.

Also for this criterium it goes that the heel location and the ankle stiffness are the most important parameters. Placing the heel further to the back (increasing hx) ensures a longer foot contact period. This is because just after heel strike there is a larger torque acting around the ankle joint bringing the foot to a horizontal position at a faster rate. An increase of the ankle stiffness k counteracts this same torque and therefore results in a shorter foot contact period.

4.2 General Trends

Table 3 summarizes the major trends that were observed in the parameter sensitivities of the performance criteria.

These general trends were found throughout the parameter space, except for one difference. Although the general trend is that increasing hx decreases

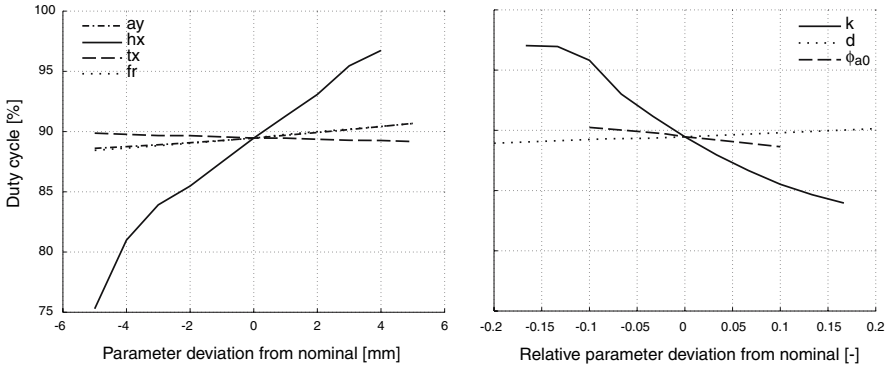


Fig. 7. Variation of the period of full foot contact with the floor under varying design parameters, both geometrical parameters (*left*) and ankle parameters (*right*)

Table 3. Major trends observed in the parameter sensitivities of the three design criteria. The effects of the two most sensitive design parameters are marked

Design Parameters	Efficiency	Robustness	Full Foot Contact Period
Vertical ankle position ay	○	○	○
Horizontal heel position hx	− −	− −	++
Horizontal toe position tx	−	−	○
Heel and toe radius fr	−	○	○
Ankle stiffness k	++	++	− −
Ankle damping d	○	○	○
Ankle equilibrium position ϕ_{a0}	○	+	○

the efficiency, an optimal value of the parameter hx was found when the ankle stiffness was high (around 10 Nm/rad). This optimum is shown in Fig. 8; it arises because there are two opposite effects present in this case. Increasing hx still has a negative effect on efficiency due to the reasoning found above, but on the other hand the positive effect on the foot contact period is beneficial to the efficiency in case of a high ankle stiffness. For the high ankle stiffness only contributes to the major robot's motion and thus the efficiency of the robot when the foot is fully placed on the ground.

4.3 Final Design

A final prototype design of the flat feet and ankle joints has been established, using the simulated influences of the design parameters. Table 4 lists the final design parameters that are going to be used. The most sensitive design parameters, hx and k, are marked. The high sensitivity of these parameters is taken into account in the prototype design, by allowing them to be mechanically adjusted. The tolerances on the other parameters are low, so

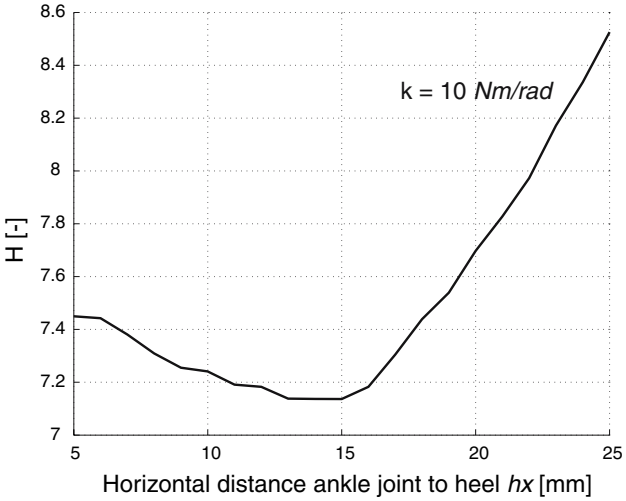


Fig. 8. The efficiency criterion H for varying hx with an ankle stiffness k of 10 Nm/rad. It shows there is a value for hx for which the efficiency is optimal

Table 4. Final design parameter settings. The shaded rows give the most sensitive parameters; this high sensitivity is taken into account in the prototype design

Design Parameters		Final Parameter Value
Vertical ankle position ay	[mm]	0
Horizontal heel position hx	[mm]	15
Horizontal toe position tx	[mm]	45
Heel and toe radius fr	[mm]	20
Ankle stiffness k	[Nm/rad]	10
Ankle damping d	[Nms/rad]	0.05
Ankle equilibrium position ϕ_{a0}	[deg]	5

small variations on these parameters are allowed if this has beneficial effects on the ease of manufacturing or weight for instance.

An initial indication to whether the final design with flat feet and ankle joints can indeed act as a good replacement for the original arced feet of Max can be found by comparing the performance measures for both designs. Table 5 gives the values of the performance measures for both designs. It shows that the new design has an efficiency that is similar to the previous arced feet design and only a slight reduction of robustness is present. The difference in the period of full foot contact underlines the fact that the new design creates the opportunity to obtain definite improvements as indicated in the introduction.

Table 5. Comparison of the new flat feet design to the arced feet design, based on the performance measures used throughout this paper

Design	H [-]	Allowed Variation $\phi_{h,ic}$ [rad]	Allowed Variation $\dot{\theta}_{ic}$ [rad/s]	Full Foot Contact Period [%]
Arced feet	7.38	0.21	0.28	0
Flat feet and ankle joints	7.14	0.20	0.21	68

5 Conclusion

This paper describes the introduction of flat feet and ankle joints in dynamic walking. It is expected that this flat feet design creates new possibilities for increasing the efficiency, robustness and versatility of dynamic walkers. Through simulation the influence of the main design parameters on the performance of a walker with flat feet was qualified. First three criteria were defined to measure the performance, considering especially efficiency and robustness. The variation of these measures due to change in the design parameters gives an insight into the influence of these design parameters. It turns out that the flat feet design mainly involves two very important and sensitive parameters. In general the efficiency and robustness of the walker can be increased by:

– Decreasing the horizontal distance from the ankle joint to the heel of the foot.
– Increasing the rotational stiffness in the ankle joint.

These two parameter changes both result in a shorter period of full foot-ground contact during walking. The parameter sensitivities that were determined in this study have resulted in a final prototype design which will soon be implemented on the dynamic walker Max at the Delft Biorobotics Laboratory.

References

1. S. H. Collins, M. Wisse, and A. Ruina. A 3-d passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, 20(7):607–615, 2001.
2. M. S. Garcia, A. Chatterjee, and A. Ruina. Efficiency, speed and scaling of two-dimensional passive-dynamic walking. *Dynamics and Stability of Systems*, 15(2):75–99, 2000.
3. M. S. Garcia, A. Chatterjee, A. Ruina, and M. J. Coleman. The simplest walking model: Stability, complexity, and scaling. *ASME Journal of Biomechanical Engineering*, 120(2):281–288, 1998.

4. K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. International Conference on Robotics and Automation 1998, pp. 1321–1326, May 1998.
5. A. D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *ASME Journal of Biomechanical Engineering*, 124(2):113–120, 2002.
6. Y. Kuroki, M. Fujita, T. Ishida, K. Nagasaka, and J. Yamaguchi. A small biped entertainment robot exploring attractive applications. International Conference on Robotics and Automation 2003, pp. 471–476, Sep 2003.
7. T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990.
8. F. Pfeiffer, K. Loffler, and M. Gienger. The concept of jogging johnnie. International Conference on Robotics and Automation 2002, pp. 3129–3135, May 2002.
9. R. Tedrake, T. W. Zhang, M. Fong, and H. S. Seung. Actuating a simple 3d passive dynamic walker. International Conference on Robotics and Automation 2004, 2004.
10. M. Wisse, D. G. E. Hobbelen, and A. L. Schwab. Adding the upper body to passive dynamic walking robots by means of a reciprocating hip mechanism. *IEEE Transactions on Robotics*, 2004, submitted.
11. M. Wisse and A.L. Schwab. First steps in passive dynamic walking. CLAWAR, 2004, submitted.
12. M. Wisse and J. v. Frankenhuyzen. Design and construction of mike; a 2d autonomous biped based on passive dynamic walking. 2nd International Symposium on Adaptive Motion of Animals and Machines, 2003.

Robotic Walking Aids for Disabled Persons

G.S. Virk[1]*, S.K. Bag[1], S.C. Gharooni[1], M.O. Tokhi[2], R.I. Tylor[3], S. Bradshaw[4], F. Jamil[5], I.D. Swain[6], P.H. Chapple[7], and R.A. Allen[7]

[1] University of Leeds, UK
[2] University of Sheffield, UK
[3] INSPIRE Foundation, UK
[4] Spinal Injuries Association, UK
[5] Pinderfields General Hospital, UK
[6] Salisbury District Hospital; UK
[7] University of Southampton, UK

Abstract. This paper presents the results of theoretical and experimental studies carried towards design and development of robotic walking aids for spinal cord injured (SCI) persons. The current orthoses technologies are reviewed and survey results from SCI people are used to identify the real requirements from users' viewpoint. These include providing the capability to perform sitting to standing and vice versa, walking and standing tasks. Powering and actuator selection, as well as control and user interface design issues are discussed in this work. Results from rig tests are presented as well as how actuator size, (and hence power consumption) can be reduced by adopting energy storage devices at the joint level. Results of a prototype robotized walking system are also presented.

1 Introduction

In medical history, the use of orthosis is not new but most of them have limited functionality and most are static, i.e. they do not use active power sources and have limited movement. The INSPIRE foundation and subsequently the University of Leeds conducted surveys among young SCI persons to determine the most important issues in orthoses and to identify how well existing aids addressed their needs, see Wright et al. [1]. The abilities to stand and have upright mobility have been identified as the most important functions, but respondents felt that current devices failed dismally to meet these needs and there was significant scope for improvement. It is felt that recent developments in humanoid robotic technologies can be utilized to help disabled persons. The aims of this research are to explore this potential of current robotic

*Corresponding author: Prof. G.S. Virk, School of Mechanical Eng, Univ of Leeds, LS2 9JT, UK; Tel: +44 113 343 2156; Fax: +44 113 343 2150; E-mail: g.s.virk@leeds.ac.uk

technologies to contributing to this area and to reducing the discrepancy between desired and available functions in powered orthoses.

The outline of this paper is as follows: Sect. 2 gives a brief review of current orthoses and robotic technologies, the orthoses requirements and possible solutions are described in Sect. 3. Prototype experimental results of research are presented in Sect. 4.

2 Brief Review of Orthosis and Robotic Technology

There are many prototype rehabilitation systems that have been reported recently. A brief review on both the current rehabilitation orthosis as well as humanoid/mobile robots is presented here, further details can be found in Wright et al. [2]. Walking orthoses can be classified into three categories: passive, powered and hybrid; in the powered devices we can include functional electrical stimulation (FES) or actuators supplied with external power. The latter falls under the category of mobile robotic technologies.

One of these is a spring brake orthosis that produces a natural-looking swing phase developed at the University of Sheffield, see Gharooni et al. [3]. Recently, a computer controlled break orthosis for FES aided gait was developed at Vanderbilt University, see Goldfrab et al. [4]. The drawback in these systems is that electrical power is required to operate the FES as well as compressed air to operate the brake. The Louisiana State University Reciprocating gait orthosis (LSU-RGO-II) has been used by many patients, see Solomonow et al. [5]. This allows only coupled flexion/extension of the hips and locks the knees. An active orthosis developed at Politecnico di Torino controls each knee with a pneumatic actuator via a chain and sprocket drive, see Belforte et al. [6]. It is believed that the power was supplied via a compressed air line which limits its usability. A device developed in Switzerland (driven gait orthosis (DGO) or Lokomat) enables paraplegics to walk on a treadmill, see Hocoma-Medical-Engineering [7] and Colombo et al. [8] but as the patient remains static, the power source is not carried. All of these devices have their own merits and demerits but none offer a viable solution that is needed by the majority of SCI persons for their every day life.

On the robotics side technology has also steadily developed over the years and some impressive systems have been demonstrated. Among these the biped developed by the Shadow Robot Company is a life size robot consisting of legs and torso, see Shadow Robot Company [9]. The joints are actuated by 28 air muscles, controlled by 12V input 56 solenoid valves and powered by an external compressed air. The RS-01 RoboDog is another impressive walking robot demonstrator produced by a UK company RoboScience, details are available on the website [10]. The joints are actuated by Epidex gear motors and RCPE power electronics. Honda has also developed several pioneering humanoid robots, see Honda [11]. The joints in the Honda humanoids are operated by DC servomotors with harmonic speed reduction. The aim of this

project is to assess these latest developments in robotics and identify if viable robotised orthoses can be realised.

3 Orthoses Requirements

The orthoses must fulfil the requirements of SCI persons. Based on the need this can be classified as functional and technical requirements. The functional requirements are many and depend on the class of injury but most of the persons need to perform sitting to standing and vice versa, walking and standing operations. The engineering specification comprises the technical requirements of an orthosis and its components necessary to achieve the desired functionality. It is primarily concerned with parameters related to the design and function of the orthosis (e.g. size, weight, power, degrees of freedom, etc), but requirements concerned with the user's interaction with the orthosis are also included as they are vital to producing practical, useable and commercially viable orthoses. The development of an orthosis focusing in these important features is discussed next.

3.1 Design Aspects of Orthosis

Freedom of movement: The human leg has six primary degrees of freedom: three at the hip, one at the knee and two at the ankle. Determination of the degrees of freedom necessary in a walking orthosis is dependant on the tasks required. To achieve a relatively normal gait pattern and allow stair climbing and rising from a chair, it is necessary to control at least hip and knee flexion. If forward propulsion is not to be provided by the upper body (e.g. with crutches) then planar flexion of the ankle is also required. It was assumed that the orthosis should manage balance in the sagittal plane (front to back) and forward propulsion. Balance in the coronal plane (side to side) is assumed to be provided by the user. Therefore, one degree of freedom (flexion/extension) needs to be provided at each of the three joints.

Motion of the joints: The range of motion was determined for each degree of freedom from data on walking, stair climbing and sitting, see Wright et al. [1]. The extension angle for the ankle, knee and hip joint in the sagittal plane should be designed to $-20°, 0°$ and $-11°$ and the flexion angle as $25°, 100°, 100°$ respectively.

Size of the orthosis: A commercial orthosis must cope with differences in the population size. Anthropometric data on stature variation in the Western population may be used to identify the size of relevant body segments, see Winter [12] and Wright et al. [1]. Table 1 lists values for three statures of Western populations. The smallest is the size which only 5% of females fall below and the largest is the size only 5% of men fall above, while the intermediate size is that of an average man. An orthosis able to accommodate this range of sizes would meet the needs of more than 90% of the population.

Table 1. Anthropometric data used to determine orthosis segment lengths

Stature	Mass (kg)	Length (mm)					
		Height	Thigh	Shank	Ft H	Ft L	Ft W
5% female	47.1	1495	366	368	58	227	82
50 % male	75.2	1774	435	436	69	270	98
95 % male	98.3	1878	460	462	73	285	103

It was decided to base the development of the prototype on an average male. From the data gathered, an average male was defined as having a body mass of 75 kg and a height of 1.77 m.

Safety, maintenance, sound and vibration: The orthosis must be designed with safety in mind and there are many aspects to be considered. For example, if the power source fails, the controller must have a backup power supply to enable a safe shut down of the system. A situation where the actuators are powered, but the controller is not, must not be allowed to occur. The power supply must be safe from potential hazards depend on the source, i.e. electric shocks, burns, chemical damage, fire, exhaust, etc. So, an appropriate safety level must be considered.

An aesthetic design and smooth operation is also important, hence, it should produce minimal noise and vibration and it should require minimal servicing or maintenance.

3.2 Powering and Actuation

Joint actuator parameters were derived from the anthropometric data mentioned above and kinetic and kinematic data, see Caballero et al. [13]. They are based on mean parameters for walking, stair-climbing and rising from a chair for an average male. While the time averaged parameters for walking are sufficient, the peak parameters may need to be higher to enable recovery from a trip. The maximum torques and power requirements for walking presented above are based on the assumption that energy can be dissipated by passive devices (e.g. brakes) with low power requirement. It follows that only power input has been included and only the peak torque during power input. Furthermore, the peak ankle torque and power prior to toe-off have not been included because it is planned to use an energy storage mechanism that will be charged by a small actuator, but will release the high short-duration torque required.

The power source must be lightweight, clean, quiet and require infrequent recharging. The aim is to provide a power source that will last for the day and can be recharged over night. Data from a healthy adult activity study was used to calculate the daily energy requirement for walking, see Hesselbach [14]. The peak power and torque required at the hip or knee during walking for

an average male is 56 W, 72 Nm respectively and maximum angular speed is 40.5°/sec, see Winter [12]. For example, if an average able bodied adult walks for 2.5 hours a day (see Goldsmith et al. [15]) and a 5 kg power source is used to run 4 actuators with an efficiency of 60% for one day, then it must have specific energy and power greater than 170 Wh/kg and 190 W/kg respectively, see Wright et al. [1].

The power consumption should be as low as possible to maximize the range and time of use of the orthosis. Hence, specific power and specific energy are important factors. With advancement in battery technologies many types of batteries currently available in the market satisfy these requirements. These include nickel zinc and nickel metal hydride batteries and Li-SO_2 and Li-$SOCl_2$ batteries, see Batteries Digest [16]. However alternative methods like energy storage technique are under developed to use low weight batteries. Other power sources that may be utilised include pneumatic, fuel cells and direct fuel combustion technologies. However, the choice is based on several factors, including the type of actuator used.

The size, weight and positioning of the actuators are of particular concern. They must not compromise good posture and load distribution when sitting down. Nor must they hinder fitting into seating and negotiating narrow openings. The use of transmissions allowing the actuator to be remote from the joint may help avoid some of these complications. Equally, the weight of the actuation and powering components should not be burden to the user. In addition, the size and weight of supporting hardware, e.g. actuator drivers, recharging/fuelling equipment should be minimized.

3.3 Control and User Interface

Robotic rehabilitation systems developed to date have significant differences from the walking orthosis envisaged by the current research. One of the main requirements is that the user should be free to supervise the orthosis, but should not need to drive it at the detailed low-task level.

Freedom of movement: It is imperative that the controller should not restrict the movement of the user. Therefore a fundamental requirement of the controller is that it should not be tethered to stationary objects. Thus, the controller must be embedded, i.e. capable of autonomous control without reference to external backup. This requirement will continue for a fully developed walking orthosis, but may not be necessary in an orthosis developed for therapeutic use in physiotherapy.

Size and weight: The size of the controller and associated components should be as small as possible to enable them to be incorporated in the orthosis structure without causing restriction of movement or unsightly bulk. Equally, the weight should not burden the user.

Power consumption: The power consumption of the controller should be as low as possible to maximise time between recharges and minimise battery weight. The powering systems must be designed with safety in mind. In

particular, it must be ensured that if the batteries ran out of power, the controller has a backup supply to enable a safe shut down to be carried out.

Modularity: In line with the philosophy of modularity in robot design conceived by CLAWAR (www.clawar.net), the controller should ideally be one that is readily available. This not only simplifies sourcing and replacement (if required), but also avoids development costs and ultimately should reduce the unit cost. Furthermore, the controller should be integrated into the system in such a way that replacement and potentially upgrading are straightforward. This is essentially a matter of opting for popular conventions such as the PC104 standard for architecture, hardware, and software and communication bus compatibility.

User interface: The user interface aspects of walking orthoses are extremely important to ensure effective control. Current interfaces are designed to operate electric wheelchairs, functional electrical stimulation (FES) orthoses, computers and manipulators and it ranges from manual to thought control so that the wide variation of disability across the targeted user groups can be covered. A successful orthosis must maximise the freedom of the user. This places significant demands on the control system and the interface between the user and orthosis. The following points are among the key requirements of the interface.

Minimise user effort: Training to operate, learning of commands, physical skill, number of limbs used to operate device, physical effort of operation, mental effort of operation, number of tasks performed by the user, and automation of low level tasks (e.g. balance, phasing of gait cycle).

Maximise the user's control of the orthosis: High level decision making, intervention at all control levels, alteration of walking parameters (e.g. speed, stride), adjustment of automated settings (e.g. control gains) and feedback. Some of these requirements are in conflict with others and the optimum solution must establish a balance between them.

Hand operated interfaces: Perhaps the most common interface in the area of assisted locomotion is that of the electric wheelchair for people with use of their arms. This typically comprises an on/off power switch, a joystick to steer, speed selector and perhaps a horn. In this common form the user has total control, but also undertakes all steering and managing tasks. However, as the control is very intuitive and requires little mental or physical effort, this interface suits many users. A number of developments have been applied to this pattern, mainly aimed at users with significant tremor or spasticity of the hands. These include replacing the joystick with a set of switches or modifying the joystick (to isometric i.e. force-controlled) and damping the input signal either physically or electrically, see Borgolte et al. [17]. Among other interfaces, the MANUS manipulator which is mounted on a wheelchair has successfully used a number of devices to control both the manipulator and the wheelchair, see Colombo et al., [8], a head controlled mouse, see Chen [18], voice control which uses a headset fitted with mercury switch tilt sensors also includes a microphone and a voice recognizer for voice, see Su and Chung [19]

and has also been applied to the control of an electric wheelchair, see Pires and Nunes [20] and neurological control is being studied, see Warwick [21].

From the investigations carried out currently developed rehabilitation devices do not have the required autonomy or functionalities, see Belforte et al. [6]; Colombo et al. [8]; Gharooni et al. [3] and hence much work still needs to be carried out.

4 Experimental Results

In order to test and assess feasible solutions different rig tests involving swing pendulum studies and able body trials have been carried out. The former is carried out to characterize the actuators needed and the latter is mainly to test the size of actuators needed to perform basic walking and sitting to standing operations. Some of the details are summarised below.

4.1 Swing Pendulum Experiment

Basic human walking and performing sitting to standing may be modelled as a swinging pendulum because it involves oscillation about a pivot point. Hence swing pendulum tests have been chosen to select the actuator requirements; a test rig was developed using a mass attached to a bar and candidate actuators used to drive it. During actuation the mass moves forward and the maximum displacement and time taken are measured so that required parameters (torque, power and speed) can be calculated to assess the potential of the actuator for adoption in future designs.

Three actuators namely air muscles, pneumatic cylinders working at 2 bar pressure and DC motors have been tested using this rig at various load. The pressure could be increase but it is kept at 2 bar for safety purposes. A comparison of the factors are presented in Table 2. The test results for these three actuators show that the maximum output specific power is 840 W/kg, 72 W/kg and 64 W/kg and specific torque 987 Nm/kg, 78 Nm/kg and 92 Nm/kg respectively. Figure 2 gives the details of power and torque developed for various loads.

Table 2. Comparison of the tested actuators

Actuators	Weight (kg)	Size (mm)	$P_{\max,sp}$ (W/kg)	$T_{\max,sp}$ (N/kg)	Power	Interface
Air muscle	0.08	L30	840	987	Comp air	Complex
Air cylinder	1.03	D50, L190	72	78	Comp air	Complex
DC motor	1.15	D60, L80	64	92	Battery	Simple

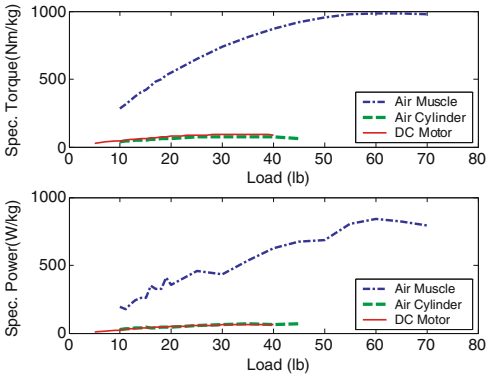


Fig. 1. Torque and power generated by actuators

Fig. 2. Prototype walking orthoses rig

An orthosis needs 56 W power and 72 Nm torque at the knee and hip joints, which are possible to achieve using all these actuators. The air muscle shows the highest specific power and torque as it is very light and it could be the first choice compared to the cylinder and motor; however we have not considered the compressed air powering system which may significantly increase the overall weight.

The air cylinder and DC motor are similar in their responses. Although these two actuators satisfy the requirements but the pneumatic supply aspects raise other concerns as stated above. Hence, DC motors are chosen as the best ones and used in the experimental system development. However, the use of air muscle actuators is worthwhile provided light weight pneumatic power supplies are made available (i.e. compressed air cylinders).

4.2 Prototype Walking Orthoses Rig

An experimental prototype orthosis developed constitutes a pair of motor driven mechanical system to demonstrate the power requirements in an orthotic system (see Fig. 2). The robotic walking orthoses rig has been designed and built to assess the potential of such a device for practical use by able bodied persons to perform sitting to standing operations as well as basic walking tests. In this rig, large 24 V/90 W rated DC motors (weighing 1 kg each) have been used; the motors are fitted with ball screw mechanisms for effective translation of motion of the knee. Tests have been performed with several people of different ages and weight in the range 65–80 kg. The requirements for power and torque to perform sitting to standing operations are verified with practical results. Basic walking is also possible with this rig via computer control of the joints. The rig does not as yet have facilities to control and step forward in an automatic way; hence an able bodied person needs to perform these tasks. The most general comment is that the orthoses is rather heavy due to the motors and frame (total weight is 14.5 kg), hence it needs to be reduced significantly.

5 Conclusions

Rehabilitation systems developed to date do not provide the functionalities required by the users even though the key technologies exist in the area of robotics. In particular, size, weight and power requirements which are important issues have been addressed by specific hardware. The main functional requirements for the design of powered walking orthoses have been discussed.

The most common and standard actuators available in the market have been tested. Air muscles have been found to be have the highest specific power and torque and seem to be the most suitable for this application. However a suitable powering system needs to be developed for day to day operation of the walking orthoses. DC motors are as the most appropriate to this stage. Their weight is high but they can be used in combination with other passive devices like energy storage to develop walking orthoses. The use of FES with active actuators may be another viable way for developing active orthoses so that the size and weight of the actuators and energy supplies that need to be carried around can be reduced.

A prototype test rig has been developed and tested to perform sitting to standing operations and basic walking. The results bring possible way to develop a robotic walking for disabled person. The rig does not yet consider the stability, so the able-bodied person needs to control it. However, a support is needed to be used for disabled persons use. The rig test results described in this paper show the process of selecting suitable actuators that can provide the power and torque needed to drive an orthoses.

The research clearly shows that a bio-robotic walking device can be developed to assist SCI injured persons. The work will be continued to develop real orthoses in future phases of the research.

Acknowledgments

The authors gratefully acknowledge funding received from the Engineering and Physical Sciences Research Council to conduct the research presented here under contract GR/R10981/02.

References

1. Wright P J, Virk G S, Gharooni S C, Tylor R I, Bradshaw S, Tokhi M O, Jamil F, Swain I D, Chappell P H, Allen R (2003) An Engineering Specification for bio-robotic walking orthosis, CLAWAR, Catania, Italy.
2. Wright P J, Virk G S, Gharooni S C, Smith S A, Tylor R I, Bradshaw S, Tokhi M O, Jamil F, Swain I D, Chappell P H, Allen R (2003) Powering and actuation technology for a bio-robotic walking orthosis. pp. 133-138, ISMCR.
3. Gharooni S C, Heller B, Tokhi M O (2001) A new hybrid spring brake orthosis for controlling hip and knee flexion in the swing phase IEEE Trans. Neural Syst. and Rehabil. Eng., 9(1): pp. 106–107.
4. Goldfarb M, Durfee W K (1996) Design of a controlled-brake orthosis for FES-aided gait, IEEE Trans. Rehabil. Eng., 4(1): pp. 13–24.
5. Solomonow M, Baratta R, D'Ambrosia R (2000) Standing and walking after spinal cord injury: Experience with the reciprocating gait orthosis powered by electrical muscle stimulation. Topics in Spinal Cord Injury Rehabilitation, 5(4): pp. 29–53.
6. Belforte G, Gastaldi L, Sorli M, Mazzeo M, Manuello Bertetto A (2000) Active orthosis – a rehabilitation device for disabled people, Proceedings of the 3rd International Conference on Climbing and Walking Robots: pp. 879–86, Professional Engineering Publishing, UK.
7. Hocoma-Medical-Engineering, (2001) Lokomat driven gait orthosis (DGO).
8. Colombo G, Wirz M Dietz V (2001) Driven gait orthosis for improvement of locomotor training in paraplegic patients. Spinal Cord, 39(5): pp. 252–255.
9. Shadow Robot Company, *www.shadow.org.uk/projects/biped.shtml.*
10. RoboScience, *www.roboscience.com/*
11. Honda, *http://world.honda.com/robot.*
12. Winter D A (1991) The Biomechanics and motor control of human gait: Normal, Elderly and Pathological, 2nd ed., Waterloo University Press.
13. Caballero R, Akinfiev T, Montes H, Armada M (2001) On the modelling of SMART non-linear actuator for walking robots, Proc Conf. CLAWAR'01.
14. Hesselbach J (1999) Shape memory actuators, in adaptronics and smart structures, H. Janocha, Editor, Springer – Verlag: Berlin, pp. 143–160.
15. Goldsmith A, Dowson D, Wroblewski B, Siney P, Fleming P, Lane J, Stone M, Walker R (2001) Comparative study of the activity of total hip arthroplasty patients and normal subjects. *Journal of Arthroplasty* Vol.16 No.5; pp. 613–618

16. Batteries Digest (2001) Teksym, Maple Plain, MN 55359; #11, ISSN 1086–9727.
17. Borgolte U, Hoyer H, Bühler C, Heck H, Hölper R (1998) Architectural concepts of a semi-autonomous wheelchair. *J Intelligent & Robotic Systems*, 22(3–4): pp. 233–253.
18. Chen Y L (2001) Application of tilt sensors in human-computer mouse interface for people with disabilities. *IEEE Trans. Neural Syst and Rehabil. Eng*, 9(3): pp. 289–294.
19. Su M C, Chung M T (2001) Voice-controlled human-computer interface for the disabled. *Computing & Control Eng. J.*, 12(5): pp. 225–230.
20. Pires G, Nunes U (2002) A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller. *J Intelligent & Robotic Syst.*, 34(3): pp. 301–314.
21. Warwick K (2002) *Implant update*. http://www.kevinwarwick.org.

The Tango of a Load Balancing Biped

Eric D. Vaughan, Ezequiel Di Paolo, and Inman R. Harvey

Centre for Computational Neuroscience and Robotics, University of Sussex, Brighton, BN1 9QH
{e.vaughan, ezequiel, inmanh}@sussex.ac.uk

Summary. One of the most popular approaches to developing bipedal walking machines has been to record the human gait and use it as a template for a walking algorithm. In this paper we demonstrate a different approach based on passive dynamics, neural networks, and genetic algorithms. A bipedal machine is evolved in simulation that when pushed walks either forward or backwards just enough to release the pressure placed on it. Just as a tango dancer uses a dance frame to control the movements of their follower, external forces are a subtle way to control the machines speed. When the machine is subjected to noise in its body's size, weight, or actuators as well as external forces it demonstrates the ability to dynamically adapt its gait through feedback loops between its actuators and sensors.

1 Introduction

Human walking is an elegant solution to bipedal locomotion. It is both resilient to disturbance and efficient. Recently the companies Honda, Sony, and Toyota have all developed their own androids that try to capture these qualities. To walk, these machines use an algorithm based on zero moment point (ZMP) [10] that computes their leg trajectories in order to keep the machines dynamically stable. The result are robots that are robust to disturbance but not particularly efficient. When humans walk they generally step forward on a straight leg and allow the opposing leg to swing past like an inverted pendulum. However these trajectory based machines step down on a bent leg to ensure a dynamically stable gait. The result is a walk that may be inefficient due to the extra power required to support their torso. To explore simpler more efficient models of walking, passive dynamic machines have been built that can walk like inverted pendulums un-powered down small inclines [6]. However, these machines' ability to resist disturbance and balance weight above their hips is limited due to their lack of an active control system. It would seem that humans somehow combine these two ideas to attain a gait that is both robust and efficient. Somehow their motor system keeps their gait stable while leveraging the passive dynamic properties of the body.

In this paper we design a bipedal robot that combines the concepts of dynamic stability and passive dynamics. While we are currently exploring a 3D machine with 12 degrees of freedom, to match the scope and size of this paper we have simplified it to 2 dimensions and 6 degrees of freedom. Our simplified machine was simulated in a physics simulator. An active control system based on neural networks was used to keep the machine stable. Using a genetic algorithm the bodies' and control systems' parameters were evolved to minimise energy consumption while exploiting the passive dynamic properties of the body. The result was a machine that walked both forward and backward. It was dynamically stable and resistant to external noise as well as discrepancies in its body's construction. It took advantage of passive dynamic properties by supporting its torso on a straight leg and using a passive knee joint.

2 Previous Work

In our previous work we evolved a three-dimensional bipedal robot in simulation that had 10 degrees of freedom but no torso above the hips. The parameters of the body and neural network were encoded in an artificial genome and evolved with a genetic algorithm. The machine started out as a passive dynamic walker on a slope but over many generations the slope was lowered to a flat surface. The machine demonstrated resistance to disturbance while retaining passive dynamic features such as a passive swing leg. [12].

At MIT the bipedal robot simulation M2 was created with 12 degrees of freedom [7]. It had passive leg swing and used actuators that mimicked tendons and muscles. Its control system was composed of a series of hand written dynamic control algorithms. A genetic algorithm was used to carefully tune the machines parameters.

3 Methods

The body of our simulated machine had eight-degrees of freedom: one at each hip, one at each knee, and one at each ankle (Fig. 1). While the machine was built in a 3D simulator only the x and y planes were explored. To make this possible the machine's legs were allowed to move freely though each other.

The physics of the body were simulated using the open dynamics engine (ODE) physics simulator [11]. Weights and measures were computed in meters and kilograms with gravity set to earth's constant of $9.81\,\mathrm{m/s^2}$. The body on average was one meter tall and had 12 parameters (Fig. 1): M_w is mass of waist, M_t is mass of thigh, M_s is mass of shank, M_f is mass of foot, L is length of a leg segment, Y_t is offset of thigh mass on the y-axis, Y_s is offset of shank mass on the y-axis. X_t is the offset of thigh mass on the x-axis, X_s is offset of shank mass on x-axis, L_f is length of foot, W is the radius of waist.

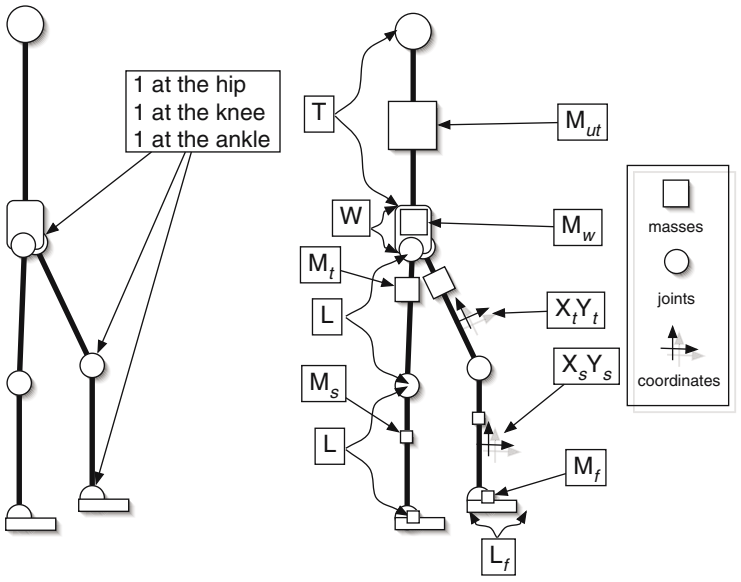


Fig. 1. *Left*: Degrees of freedom in the body. *Right*: body parameters evolved by the genetic algorithm

The weight of the torso M_t was 1.5 times the weight of M_w. W was 1/3 of L. T was $2 * L + \frac{W}{2}$. Parameter ranges were selected based on observations of the human body. The mass of the foot was restricted to be less than that of the shank, the mass of the shank was less than that of the thigh, etc. All parameters were encoded in the genome, for optimisation through a genetic algorithm. Feed-forward continuous time neural networks (CTNN) were used to add power to the machine. Unlike traditional neural networks, a CTNN uses time constants to allow neurons to activate in real time and out of phase with each other. For a detailed analysis of this kind of network refer to [1]. The state of a single neuron was computed by the following equation:

$$\tau_i \dot{y_i} = -y_i + \left[\sum_{j=1}^{N} w_{ji} \sigma(g_j(y_j)) \right] + I_i + \Omega \tag{1}$$

Where y is the state of each neuron, τ is a time constant, w is the weight of an incoming connection, σ is the sigmoid activation function tanh(), g is the gain, I is an external input, Ω is a small amount of noise in the range of $[-0.0001, 0.0001]$. The state of each neuron was integrated with a time step of 0.2 using the Euler method. In our model neurons were encoded in the genome with τ and g while the axons weights were encoded with real values in the range of $[-5, +5]$. Biases were omitted.

Two islands [13] of a geographically distributed genetic algorithm [4] were used each with a population of 50 individuals. The genotype of each individual

contained real valued gene. After each mating gene was mutated by adding a small random number in the range of $[-\frac{m}{2.0} + \frac{m}{2.0}]$. Intially the mutation rate m was set to 0.5 and then lowered slowly during evolution. Crossover was random. This kind of evolutionary algorithm was used as it has previously proved effective in this context but we do not discount other algorithms being equally effective.

4 Network Design

Each leg was given a copy of an identical neural network each of which had two different states either *stance* or *swing*. Upon creation they were connected to each other by sharing two neurons called centre of mass (COM) and *winner* (Fig. 2). To walk one network became *stance* keeping the leg straight and

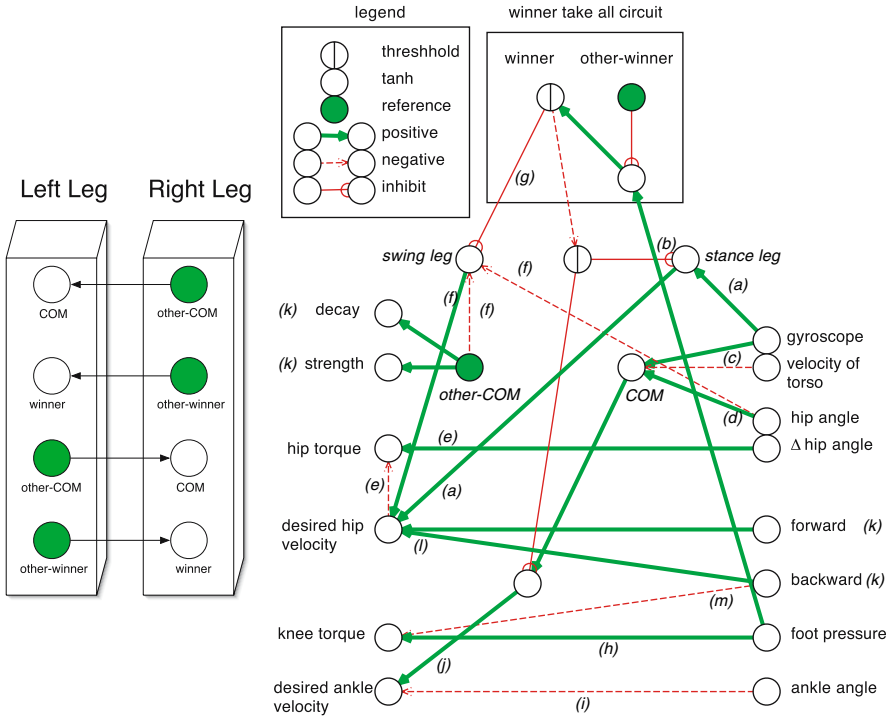


Fig. 2. *Left*: Each network is identical and communicates through shared neurons. Solid circles are actual neurons, transparent ones are references. *Right*: network structure. Inhibitory connections push the activation toward zero regardless of sign. Actuators had two inputs desired velocity and maximum torque available to achieve that velocity. If torque was not specified it became the magnitude of the desired velocity

supporting the torso while the other became *swing* and guided the leg either forward or backward. On each step the roles of the networks were swapped. Each network decided its state by using a winner take all circuit (Fig. 2). The leg with the most foot pressure became the stance leg and inhibited the other to become the swing leg. When the stance leg's foot later lost contact with the ground the other leg became the stance leg and inhibited the first. The *stance* state was implemented by making a positive connection between a gyroscope and the desired hip velocity (a) in (Fig. 2). If the torso fell forward the leg moved under it by powering the hip. An Inhibitory connection between the winner take all circuit inhibited this behaviour when the network was not in stance state (b).

When a network was in *swing* state it needed to move the leg forward just enough to catch the machines centre of mass (COM). This is a similar problem to that of a Segway human transporter [9]. A Segway has only two wheels but must support a person standing on top by computing how fast it must turn its wheels to drive under its COM. A simple algorithm to do this is to attach a gyroscope and wheel encoder to the machine and to set up a simple feedback loop to the wheel motors. If the sum of the gyroscope angle and its derivative added to the sum of the wheel angle and its derivative is connected to the wheel motor the machine will automatically balance itself. In our machine the legs take the place of the wheels so we compute our COM by summing our gyroscope and current hip angle with the velocity of the torso.This can be computed from a function of the derivative of the hip angle, the length of the leg, the derivative of the gyroscope and the length of the torso. However, for simplicity we take the velocity directly from the physical simulator (c). The hip torque was computed by taking the derivative of the hip angle and subtracting the desired hip velocity (e). To move the leg negative connections were made from the leg's hip angle sensor and the other leg's COM to the hip actuator's desired velocity (f). This allowed the leg to move to an equilibrium point just in time to catch its mass falling forward. Similar to the stance leg this behaviour was inhibited when not in *swing* state (g). To keep the knee straight when the foot was touching the ground a positive connection was made from the foot's pressure sensor to the knee velocity (h). When the foot was off the ground the knee was completely un-actuated and allowed to passively swing. Knee torque was set to the magnitude of its velocity. Ankles were implemented by placing a negative connection between their angle sensor and desired velocity. Ankle torque was automatically set to the magnitude of their desired velocity making the ankles act as a damped spring (i). To inject power into the stride when the COM was forward a positive connection was made from the COM to the ankle velocity (j).

4.1 Stepping Reflex

Two reflexes were used to automatically lift the swing leg forward or backward if the COM shifted past a threshold. Without these the machine could support

the torso and guide the swing leg but had no way of initially getting a foot off the ground. Each foot was given a recharge counter and four neurons were added to each network: *forward*, *backward*, *decay*, and *strength* (k). If the foot was fully charged and the COM shifted over a threshold, the forward or backward neurons were pulsed lifting the leg and reseting the recharge counter. The rate of decay and strength of the pulse was taken from the rate and strength neurons. Each of these neurons were connected to the COM allowing the machine to take larger steps depending on how far off balance it was. The greater the magnitude of the rate the slower the decay of the pulse over time. The greater the magnitude of strength the larger the magnitude of the initial pulse. To step forward or backward a positive connection was made between the *forward* and *backward* neurons and the desired hip velocity (l). In the forward case this allowed the hip to lift and the knee to swing passively forward. In the backward case an additional negative connection was made between the *backward* neuron and the knee torque(m). This momentarily compressed the knee enough to allow it to passively swing past the stance leg.

5 Experiments

5.1 Dynamic Gait Adjustment

In connected ballroom dances such as tango the lead communicates intended movements to the follower through a dance frame. When the lead moves forward they gently push into the follower causing them to step backward. The more the lead pushes the faster the follower moves. In our first experiment we use this idea to teach our machine to walk at different speeds both forward and backward. A population of machines were constructed and each one was evaluated for fitness in the following way:

1. Place the machine standing upright with legs together on a flat surface.
2. Constantly push it forward with just enough force to get it to reach a desired velocity chosen at random. Compute its fitness with the the following function:

$$fitness = t\left(\frac{1}{1+p}\right)\left(\frac{1}{1+z}\right)\left(\frac{1}{1+x}\right)\left(\frac{1}{1+f}\right) \tag{2}$$

   Where: t is time the machine walked before falling, p is the torque used, z is the acceleration of the hip along the z dimension, and x is the rotation of the hip around the x-axis. f is the force required to get the machine to the desired velocity. p, z, f, and x were averages taken over the entire evaluation time.
3. Put the machine back to the starting position and push it backward with just enough force to reach a desired velocity chosen at random. Compute its fitness again.

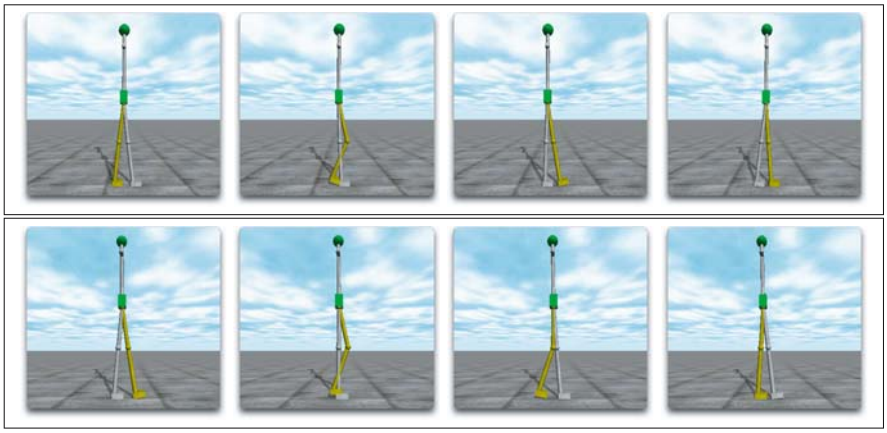


Fig. 3. *Top*: gait of machine walking forward. *Bottom*: gait of machine walking backward

4. Take the worse fitness of the two runs.
5. Repeat step (1–4) nine more times and take the average fitness. This type of evaluation ensures a machine walks equally well forward as backward with the minimum amount of force pushing on it. It also selects for machines that walk as long and straight as possible without explicitly specifying how they move their legs. This allows their leg trajectories to emerge from the dynamics of their bodies rather than from the observations of a human gait. The machine's gait after 800 generations is illustrated in Fig. 3.

To determine how closely our machine was able to match a desired velocity two graphs were made. One comparing the machine's desired velocity with its actual velocity and one showing its change in foot timing at several different velocities Fig. 4. The first graph shows a valley. As the machine is pushed

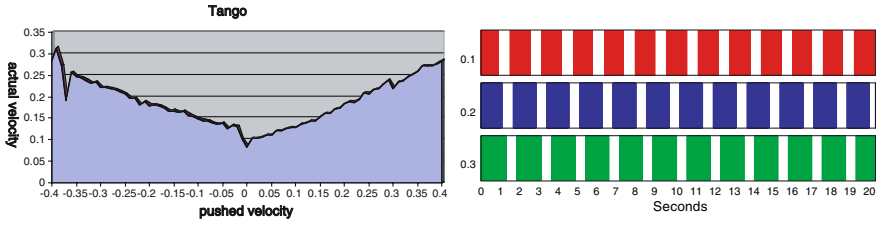


Fig. 4. *Left*: Comparison of the machine's desired velocity with its actual velocity. The x axis is the desired velocity forward (positive) or backward (negative) and the y axis is the absolute value of the machine's actual velocity. *Right*: The foot strike timing of the left leg at three different speeds 0.1, and 0.3 m/s. Solid bars indicate the foot is touching the ground while transparent indicate the foot has left the ground

either forward or backward its velocity increases or decreases respectively. If the machine is close to zero it attempts to stand still.

5.2 Robustness To Noise

One of the great difficulties with computer simulation is that it often fails to transfer to a physical robot due to unforeseen differences found in the real world. Therefore a controller that is adaptable to unforeseen changes may have a better chance of making the transfer. To determine how adaptable our machine was we subjected it to both internal and external noise. Internal noise was defined as errors in the body itself such as incorrect body massege, leg lengths, or noisy acutators. External errors were defiened as random external forces that attempt to push the machine off balance. Internal noise was introduced by adding an error to each body parameter and actuator upon construction (3).

$$p = p + p * (rand() - 0.5) * 2.0 * e \quad ae = (1 - rand()) * e \tag{3}$$

Where: p is the body parameter, $rand()$ is a function that returns a random number in the range of 0 and 1, e is the percentage of error. Errors were introduced to actuators by multiplying the desired velocity and torque from the network by ae and adding random noise in the range of $[-\frac{e}{10}, +\frac{e}{10}]$ on each time step. ae and p were computed just once on the construction of the machine. External noise was added by applying a random force along the x and z axis (4) each time step.

$$x = (rand() - 0.5) - 0.5) * 0.5 * e \quad z = (rand() - 0.5) * 0.5 * e \tag{4}$$

The machine was tested for the average number of forward steps it could take over 20 trials when pushed forward at 0.2 m/s for error rates between 0 and 50% (Fig. 5). The graph shows a graceful degradation in the number of steps taken as nois increases. With 10% noise the macine is able to take 93 steps on average and after 50% noise the machine can still take an average 10 steps.

5.3 Efficiency

In studies conducted in Kenya women there were observed to carry great weights on their heads while using very little energy [2]. This was attributed to the fact that they walk like inverted pendulums supporting the weight on a straight leg as they move forward. To study the efficiency of our system the population of machines was evolved for an additional 200 genrations and evaluated for their ability to walk forward while carrying varying amounts of weight. Initially the body weighed 42 kg, 32 kg of which were contributed by the torso. In the experiments 20 walks were conducted increasing the torso's

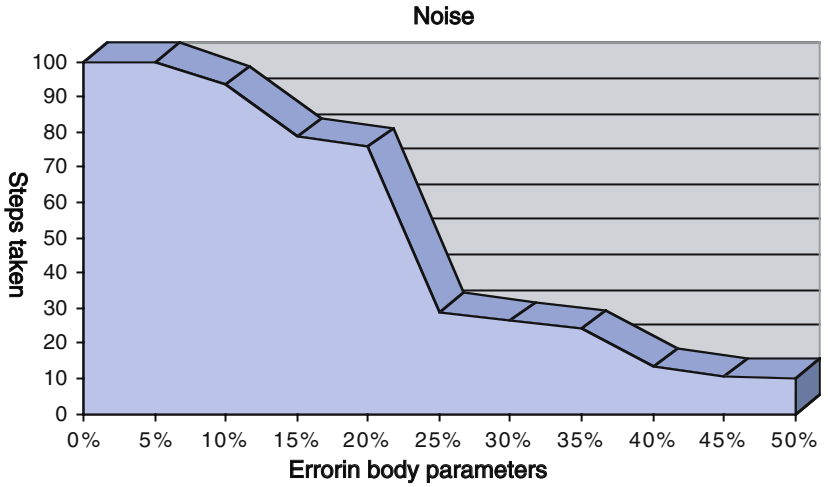


Fig. 5. Graph illustrating robustness to error. The y axis is the number of steps taken over 20 trials while walking forward at a velocity of 0.2 m/s. x is the percentage of error e. The number of steps taken was capped at 100

by a percentage of the total body mass. At 0% the torso was unchanged at 32 kg, at 200% (Fig. 6) the torso was $32 + 42 * 2 = 116$ kg. To start each walk the machine was pushed to a velocity of 0.25 m/s and its energy consumption was computed as it walked 60 meters. At 0% it used 193 kg-m of torque and at 200% it required 292 kg-m. This revealed that to carry twice its body weight (200%) this machine required only 51% more energy (Fig. 6).

5.4 Construction of a Physical Machine

To transfer our simulation to a physical machine we are currently developing special actuators based on series elastic actuators developed by MIT [8]. These devices combine a worm gear with a spring to make a device that has shock tolerance, high fidelity and low impedance. By setting up a negative feedback loop between spring deflection and a desired deflection they can be used to apply varying amounts of force. If the desired deflection is zero they can emulate passive components. When this is combined with a proportional-derivative (PD) controller whose input is a desired velocity and maximum torque, it can implement the type of actuators used in our simulation. Our primary concern is issues involving unforeseen differences between simulation and reality. While our experiments have demonstrated resistance to errors in the body and acutators we also have other techniques that can aid in the transfer. Jakobi successfully transfered a controller for a simulated khepera robot to a physical one by carefully adding noise during evolution [5]. Plastic weight updating rules have also been used to make the transfer to physical machines. In one experiment [3] evolved a controller that allowed a simulated

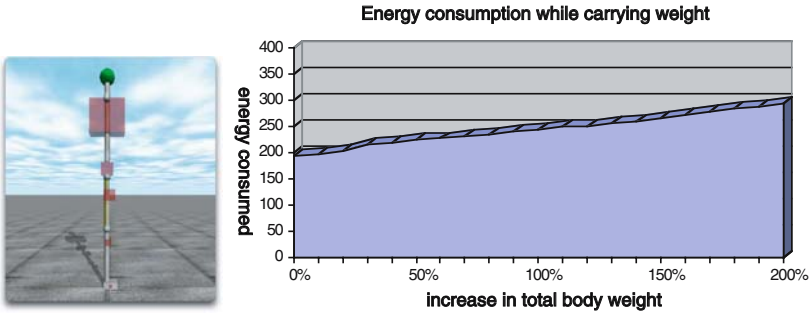


Fig. 6. *Left*: The machine carrying 200% of its original weight. *Right*: Graph illustrating ability to carry weight (Boxes indicate mass distribution). The y axis is the total torque required in kg-m and the x axis it the weight carried in kg

Khepera robot to navigate a maze and then transfered it to a physical one. To demonstrate how adaptive plasticity could be they then transferred the same controller to a different six wheeled Koala robot. In a second experiment they evolved a four legged walking robot in simulation and successfully transfered it to a physical machine. While our physical robot is only in its early stages we are taking appropriate measures to ensure our machine will transfer successfully.

6 Conclusion

We have demonstrated a machine that combined dynamic stability with passive dynamics. Its control system was composed of two identical neural networks that formed a dynamic system whose basin of attraction was walking. When pushed forward or backward it walked just enough to support its centre of gravity. By using passive knee swing and stepping down on a straight leg it demonstrated the ability to support large weights efficiently. The machine maintained stability even when subjected to noise such as external forces, body parameter errors, and actuator errors. This is an interesting result since the CTNNs of our model do not store information through weight changes, as many conventional artificial neural networks do. Instead it had to rely entirely on the feedback between its sensors and actuators. This adaptability may provide a mechanism for transferring simulated control systems to physical robots.

This technique is very powerful and we are currently using it to explore more complex 3 dimensional bipedal machines. Some of these simulated machines have demonstrated the ability to dynamically run. We are now beginning to build a physical android based on this model and hope to discover further insights into how to use these methods to develop

practical bipedal machines. Videos of our simulated machines can be found at (www.droidlogic.com).

References

1. R. D. Beer. Toward the evolution of dynamical neural networks for minimally cognitive behavior. In P. Maes, M. Mataric, J. Meyer, J. Pollack, and S. Wilson, editors, *From animals to animats 4: Proc. 4th International Conf. on Simulation of Adaptive Behavior*, pp. 421–429. MIT Press, 1996.
2. S. Eugenie. How to walk like a pendulum. *New Scientist*, 13, 2001.
3. D. Floreano and J. Urzelai. Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13:431–443, 2000.
4. P. Husbands. Distributed coevolutionary genetic algorithms for multi-criteria and multi-constraint optimisation. In T. Fogarty, editor, *Evolutionary Computing, AISB Workshop Selected Papers*, volume 865 (LNCS), pp. 150–165. Springer-Verlag, 1994.
5. N. Jakobi, P. Husbands, and I. Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. *ECAL*, pp. 704–720, 1995.
6. T. McGeer. Passive walking with knees. In *Proceedings of the IEEE Conference on Robotics and Automation*, volume 2, pp. 1640–1645, 1990.
7. J. Pratt and G. Pratt. Exploiting natural dynamics in the control of a 3d bipedal walking simulation. In *Proceedings of the International Conference on Climbing and Walking Robots (CLAWAR99), Portsmouth, UK*, 1999.
8. D. W. Robinson, J. E. Pratt, D. J. Paluska, and G. A. Pratt. Serieselastic actuator development for a biomimetic robot. *IEEE/ASME International Conference on Advance Intelligent Mechantronics.*, 1999.
9. Segway. Segway models. www.segway.com, 2004.
10. C. Shin. Analysis of the dynamics of a biped robot with seven degrees of freedom. In *Proc. 1996 IEEE International Conf. of Robotics and Automation*, pp. 3008–3013, 1996.
11. R. Smith. The open dynamics engine user guide. http://opende.sourceforge.net/, 2003.
12. E. Vaughan, E. Di Paolo, and I. Harvey. The evolution of control and adaptation in a 3d powered passive dynamic walker. In *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems, ALIFE'9 Boston*. MIT Press, September 12th-15th, 2004 (http://www.droidlogic.com/sussex/papers.html).
13. D. Whitley, S. Rana, and R. Heckendorn. The island model genetic algorithm: On reparability, population size and convergence. *Journal of Computing and Information Technology*, 7:33–47, 1999.

Locomotion Modes of an Hybrid Wheel-Legged Robot

G. Besseron, Ch. Grand, F. Ben Amar, F. Plumet, and Ph. Bidaud

Laboratoire de Robotique de Paris (LRP), CNRS FRE 2507 – Université Pierre et Marie Curie, Paris 6, 18 route du Panorama – BP61 – 92265 Fontenay-aux-Roses, France
{besseron,grand,amar,plumet,bidaud}@robot.jussieu.fr

Abstract. The main objective of this paper is to compare and evaluate the performance of several locomotion modes of an hybrid wheel-legged robot. Each studied locomotion mode is described, compared and evaluated according to the same criteria, which are the gradeability – i.e. the climbing ability – and the power consumption. A hierarchical scheme dedicated to the selection and the control of each locomotion mode is also presented.

Key words: gradeability, stability, controllability, power consumption, multi-modal locomotion, wheel-legged robot.

1 Introduction

Autonomous exploration missions require mobile robots that can carry out high performance locomotion tasks while insuring the system integrity. For applications such as planetary and volcanic exploration or various missions in hazardous areas or construction sites, the locomotion performance is of first importance.

Vehicle motion on uneven surfaces involves complex wheel-ground interactions that are related to the geometrical and physical soil properties. Therefore enhancing the locomotion performance in such environment requires the design and control of innovative locomotion systems.

Due to their ability to adapt their posture and to cross over high terrain discontinuities, legged systems have been considered for a long time (and are ever considered) as a possible way to increase the field of accessible terrains for autonomous vehicles.

More recently, wheeled systems with passive suspension systems have been introduced (see for example the Nomad [10], Shrimp [11], Nexus [15] robots or the Rocky rovers [13]) to enhance the terrain adaptability and allowing these vehicles to address more challenging terrain including ground discontinuities that are higher than the wheel radius.

Between these two classical categories of locomotion mechanisms, hybrid robots are articulated vehicle with active internal mobilities or hybrid wheel-legged vehicles like Azimut [9], GOFOR [12], SRR [8] and HyLoS [3] robots. With these redundantly actuated systems[1], the internal mobilities can be used to improve the stability and/or the wheel traction leading to a global rough-terrain mobility enhancement.

Hybrid wheel-legged robots like Workpartner [7] or HyLoS [3] can also combine different locomotion modes. We believe that one of the key factor of autonomous exploration mission's success is the ability of the system to automatically adapt its configuration and locomotion modes to the local difficulty of the traversed terrain.

In this paper, we describe the three main locomotion modes of the hybrid wheel-legged robot HyLoS (see Fig. 1). The locomotion performance in terms of gradeability and power consumption is compared for each mode.

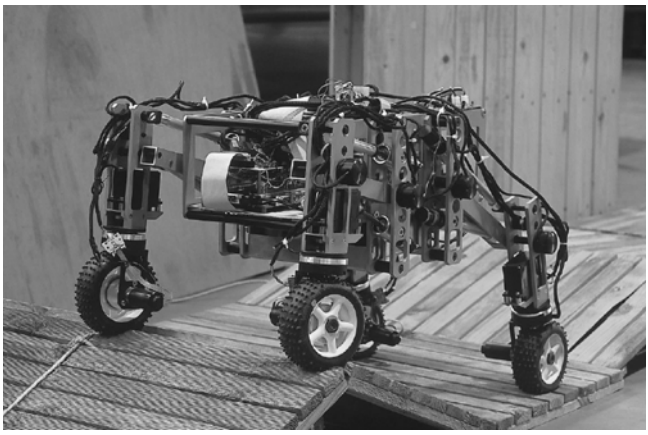


Fig. 1. HyLoS Robot

The knowledge of the locomotion performance of the system will then be used in an on-line hierarchical control scheme based on two main loops: an internal loop dedicated to the locomotion mode regulation and an external loop dedicated to the switching between different locomotion modes as a function of the current geometrical and physical soil properties.

[1]The number of actuated degrees of freedom is greater than the dimension of the system workspace

2 Locomotion Modes Description

2.1 Pure Rolling Mode (Mode 1)

This is a trivial mode where the internal mobilities of the system are not used. On ground without irregularities and discontinuities like road, it is the most efficient locomotion mode (in obvious condition that the leg transmission mechanisms are irreversible or passively blocked).

2.2 Rolling Mode with Reconfiguration (Mode 2)

In this case, the internal active mobilities are used to optimize the posture in order to enhance the locomotion performance. The used criteria are the tipover stability margin and the wheel-ground contact force balance.

A suboptimal posture of the robot that optimize the normal component of contact force is defined [6]. The normal forces balance is optimized by assuming the distribution of vertical component of contact forces. Because of the particular design of HyLoS this correspond to maintain the roll angle φ to zero, and to configure each leg in such way that projected distances between contact points and the platform center of gravity are equal. The other posture parameters that are the ground clearance z_g, the pitch angle ψ and the nominal wheelbase are specified by a high level controller with respect to the platform task (vision, manipulation). Here ψ is defined to make the longitudinal vehicle axis parallel to the slope.

This locomotion mode is adapted to irregular ground without discontinuities like sloping ground or rough terrain. A graphical representation of this posture for different slope configurations is given to Fig. 2.

$\theta_s = 0$ deg. $\theta_s = 45$ deg. $\theta_s = 90$ deg.

Fig. 2. Rolling mode with reconfiguration for different slope configurations

2.3 Peristaltic Mode (Mode 3)

This mode is similar to one used by worms. It consists in moving the system mass by using its inner mobilities (traction of wheels is used only to move the

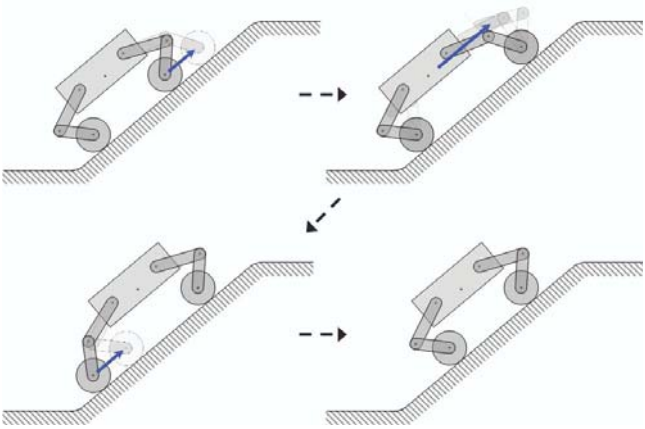


Fig. 3. Peristaltic mode

leg). It exists a lot of different cyclic gait motions to move the robot in this mode. In this study, we choose one cyclic gait in which each pair of wheels in the frontal plane moves only when the other one is firmly braced to the ground (see Fig. 3). This mode is well adapted for locomotion on non-cohesive soils [1].

3 Locomotion Modes Performance

In this section advantages and disadvantages of each locomotion mode are studied with respect to different performance criteria. Considered criteria are the gradeability (i.e. the maximum slope that a vehicle can climb without compromising the vehicle's stability or it's ability to move forward) and the power consumption.

A rolling resistance model, issue from Terramechanics equations generating wheel sinkage due to soil compaction, has been implemented to take this phenomena into account. This resistance force is expressed by Bekker [2] as:

$$R = b\left[(k_c/b + k_\phi)\frac{z^{n+1}}{n+1}\right] \tag{1}$$

where z is the wheel sinkage given by:

$$z = \left[\frac{3W}{b(3-n)(k_c/b + k_\phi)\sqrt{D}}\right]^{\left(\frac{2}{2n+1}\right)} \tag{2}$$

In theses equations, W is the load, b is the contact width, D is the wheel radius and k_c, k_ϕ, n are the Bekker's parameters of the ground pressure behavior. Traction force T, for a rigid wheel, is also given by Bekker:

$$T = (Ac + W \tan\phi)\left(1 - \frac{K}{sl}\left(1 - \mathrm{e}^{-\frac{sl}{K}}\right)\right) \tag{3}$$

where K is the shear deformation modulus, s the shear displacement due to wheel-slippage, c the soil cohesion, A the contact area, ϕ the friction angle, and l the length of contact area.

3.1 Evaluation of Gradeability

This criterion is based on the evaluation of the stability and the controllability limits for each locomotion modes according to the slope angle η. The stability limit is defined as the max angle η_s for which the contact is loss in at least one wheel ($F_n = 0$) and the controllability limit is the max angle η_c when one of the reaction force leaves the friction cone ($|F_t| = \mu F_n$). They are evaluated as function of the robot yaw angle θ from 0 to 360°. These two parameters (θ and η) define all the possible slope configuration (Fig. 4).

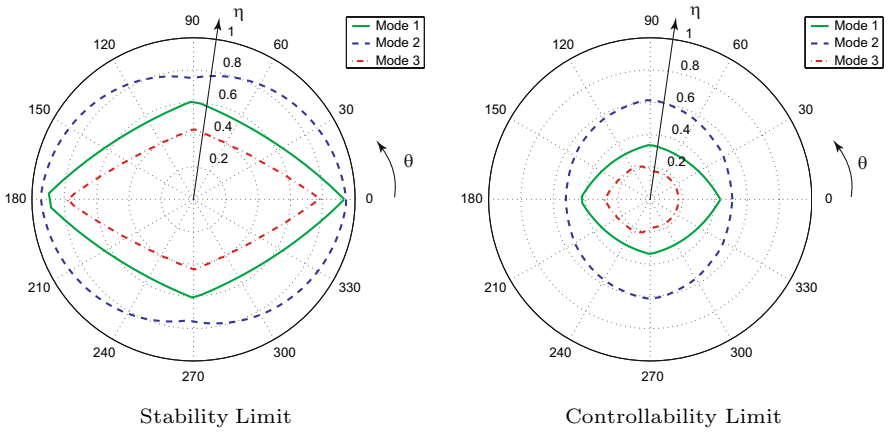


Fig. 4. Evaluation of Gradeability of different locomotion modes on hard soil

The results show an obvious superiority of mode 2 in terms of stability and controllability compared with other locomotion modes. Unlike locomotion mode 1 and 3, controllability limit of mode 2 increases when θ approaches to banked angles (90° and 270° values). Moreover its value is significantly higher for this mode than for the others (with factor three against mode 1 and six against mode 3). This is due to the reconfiguration capability which is more important in banked slope than frontal slope.

Concerning locomotion mode 3, the stability limit shows an interest mainly in the frontal direction ($\theta = 0°$). Besides, this is the standard configuration of the mode 3. However controllability limit for mode 3 is not significant because, as previously described at Sect. 2.3, the robot displacement is based on the motion of internal mobilities instead of the traction of wheels.

3.2 Evaluation of Energy Consumption

The second criterion consists in an evaluation for each locomotion mode of the energy consumption of the vehicle moving on a frontal slope. In order to compare average power or global energy, the calculus is done for constant distance and constant speed. For this evaluation we consider two terrains with different physical properties whom parameters are listed in Table 1. Then we compute the mean power consumption to travel a constant distance for different slope angles.

Table 1. Bekker's parameters for the two studied terrain (from Wong [14])

Type	n	K_c (kN/m^{n+1})	K_ϕ (kN/m^{n+2})	c (kPa)	K (cm)	ϕ (deg)
Hard soil	0.2	2.56	43.12	1.38	0.75	38
Soft soil	0.8	16.5	811	3.17	2.71	25.6

The results are showed in Fig. 5. Each plot represents the mean power consumption as function of slope angle. The advantage of peristaltic mode is clearly demonstrated for non-cohesive soils and/or terrain with high slope angle. Whereas mean power becomes rapidly infinity for pure rolling mode due to important slippage, peristaltic mode stay practically constant for any kind of ground. However on hard soil with low slope angle (<10°), pure rolling mode presents advantage to have a low energy consumption.

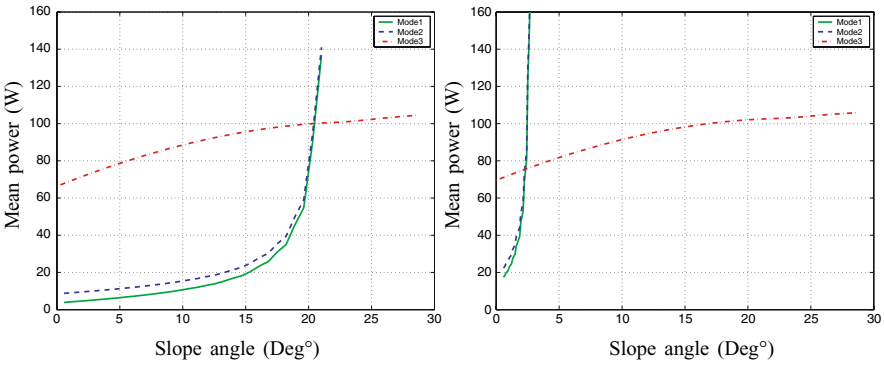


Fig. 5. Evaluation of Power Consumption on two kinds of ground

4 Control Scheme

The overall control scheme of HyLoS is hierarchically divided into two main loops. The external loop is based on a stereovision system that produces digital elevation map and texture information. Its aim is to identify the crossed terrain properties and select the most appropriated locomotion mode, and control the switch between each modes. The overall control scheme is depicted Fig. 6.

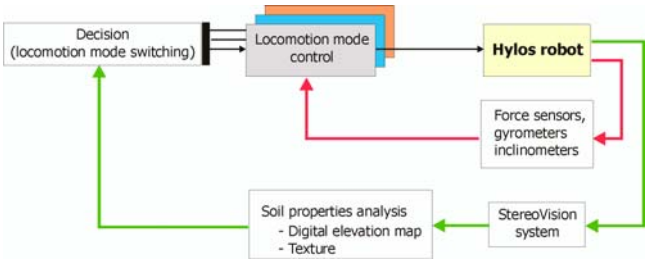


Fig. 6. Overall hierarchical control structure

The internal loop is dedicated to the control of the selected locomotion mode. The loop relative to locomotion mode 1 is based on traction control. For the rolling mode with reconfiguration, this loop consists of a velocity model-based posture control which is described in [4, 5] (see Fig. 7). The loop relative to the peristaltic mode is based on sequential motion generator.

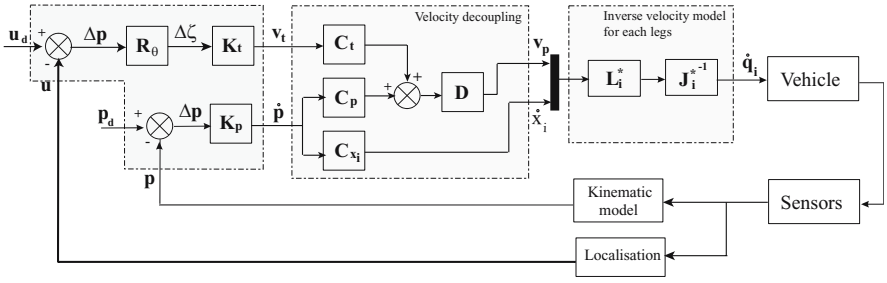


Fig. 7. Posture control scheme

5 Conclusion

The external control loop selects and switches between different locomotion modes. This selection is based on stereovision information and the measure of the locomotion performance.

The performance of each mode on different soils have been evaluated and compared with criteria that are gradeability and power consumption. In terms of energy consumption, peristaltic mode seems to be the most adapted ones on non-cohesive soils and/or terrain with high slope angle whereas both rolling modes are more suitable for hard soil with low slope angle. And rolling with reconfiguration mode is clearly superior in terms of gradeability.

However evaluation of locomotion modes performance is limited since it is done considering quasi-static notion. This will be improved in future work by performing dynamic simulation of the different locomotion modes.

References

1. G. Andrade, F. BenAmar, Ph Bidaud, and R. Chatila Modeling wheel-sand interaction for optimization of a rolling -peristaltic motion of a marsokhod robot. In *Int Conference on Intelligent Robots and System*, pp. 576–581, 1998.
2. M.G. Bekker. *Introduction to terrain-vehicle system.* The University of Michigan Press, 1969.
3. F. BenAmar, V. Baudanov, P. Bidaud, F. Plumet, and G. Andrade. Aigh mobility redundantly actuated mini-rover for self adaptation to terrain characteristics. in. *3rd Int.* Conference on Climbing and Walking Robots (*CLAWAR'00*), pp. 105–112, 2000.
4. C. Grand, F. BenAmar, F. Plumet, and P. Bidaud. Stability control of a wheel-legged mini-rover. in *5th Int. Conference on climbing on Walking Robots* (*CLAWAR'00*), pp. 323–331, 2002.
5. C. Grand, F. BenAmar, F. Plumet, and P. Bidaud. Decoupled control of posture and trajectory of the hybrid wheel-legged robot hylos. In *IEEE Int. Conference on Robotics and Automation,* pp. 5111–5116, 2004.
6. C. Grand, F. BenAmar, F. Plument, and P. Bidaud. Stability and traction optimisation of a recfigurable wheel-legged robot. In *to be publisged in the International Journal of Robotics Resarch,* Oct. 2004.
7. A. Halme, I. Leppänen, S. Salmi, and S, Ylónen. Hybrid locomotion of a wheel-legged machine. in *3rd Int. Conference on Climbing and Walking Robots (CLAWAR'00),* 2000
8. K. Iagnemma, A. Rzepniewski, S. Dubowsky, and P. Schenker. Control of robotic vehicles with actively articulated suspensions in rough terrain, *Autonomous Robots,* 14(1):5–16, 2003.
9. F. Michaud et al. Azimut: a leg-track-wheel robot. In *IEEE Int. Conference on Intelligent Robots and Systems,* pp. 2553–2558, 2003.
10. E. Rollins, J. Luntz, A. Foessel, B. Shamah, and W. Whittaker. Nomad: a demonstration of the transforming chassis. in *IEEE Int. Conference on Robotics and Automation,* pp. 611–617, 1998.
11. R. Siegwart, P. Lamon, T. Estier, M. Lauria, and R. Piguet. Innovatice design for wheeled locomotion in rough terrain. *Robotics and Autonomous System,* 40:151–162, 2002.
12. S.V. Sreenivasan and B.H. Wilcox. Stability and traction control of an actively actued micro-rover. *Journal of Robotics Systems,* 11(6):487–502, 1994.

13. R. Volpe. Rocky 7: A next generation mars rover prototype. *Journal of Advanced Robotics*, 11(4): 341–358, 1997.
14. J.Y. Wong. *Theory of ground vehicle.* Wiley-Interscience, 2001.
15. K. Yoshida and H. Hamano. Motion dynamics of a rover with slip-based traction model. In *IEEE Int Conference on Robotics and Automation,* pp. 3155–3160, 2002.

Stabilizing Dynamic Walking with Physical Tricks

Norbert M. Mayer[1], Amir A. Forough-Nassiraei[2], Ziton Hsu[3], Ferenc Farkas[3], and Thomas Christaller[4]

[1] Institut für Informatik, Universität Freiburg, Georges-Köhler-Allee, Geb. 52 D-79110 Freiburg,
[2] FAIS – Robotics Research Institute, 2-1 Hibikino, Wakamatsu-ku, Kitakyushu 808-0135, Japan
[3] Viewcam BT., Budapest, Hungary
[4] Fraunhofer Institut für Autonome Intelligente Systeme, Schloss Birlinghoven, D-53754 St. Augustin, Germany

Summary. We present here some concepts how a dynamic walker can be stabilized. One concept is to stabilize the stance by using a fast heavy rotor, a gyro. The dynamics of a symmetric, fast rotating gyro is different from from a non-rotating solid body, e.g. in the case of small disturbances it tends to keep the axes the same. We investigate in numerical simulations in which way these effects are useful for dynamic walking. In the discussion we outline further ideas how to use the gyro in actuated walkers. In addition, experiments with a real 2D walker are outlined. Here we used an alternative method in which the walker walks like "a man on crutches".

1 Introduction

The goal of passive dynamic walking is to exploit the natural dynamics of pendulum-like legs in order to achieve fast walking in bipedal robots. This approach is increasingly interesting as an alternative to the conventional methods like zero moment point control (ZMP) [1]. A large number of publications (i.e. [4]) have addressed properties of bipedal walking in machines and living beings. However, practical applications suffer from narrow stability intervals, complexities involving nonlinear control, and tight restrictions in achievable gait properties. A common idea is to overcome these problems by optimizing the anatomy and the control of the robots using evolutionary or other general optimization methods [2, 3]. Different from these approaches we focus on simple and basic engineering techniques and tricks in order to increase stability during walking.

We present here experiences on a real 2D walker and results from simulations with biped walkers with passive and – briefly – one with active joints. In the case of the 2D walker some modifications on the feet and the knees

have been done. In the case of the simulated biped walker we make use of the stabilizing effect of a fast rotating symmetric rotor or gyro. Stability with respect to perturbations orthogonal to the walking plane which is trivially present in 2D walkers, can be achieved to a similar extent when exploiting the dynamic effects of the gyro. We present the latter results in the second part of the paper.

2 The Real 2D Walker

A walker has been constructed according to the descriptions in [5]. However, since the walking properties depend on the exact weight distribution, a few modifications for this individual were necessary.

The aim was to improve the balance and the gait pattern in order to permit stable downhill walking. The balance can be maintained by adding weight either in front or in the back of the walker. The same is true for improving the gait, where, however, additional changes were necessary, namely

- the stiffening of the outer pair of knees and a modification of the positions of the feet. The walker tended to fall because the second pair of legs bend to early resembling a human using crutches
- shift position of the feed in the inner pair of legs in order to prevent the swinging leg from hitting the ground (cf. Fig. 1 for illustration)

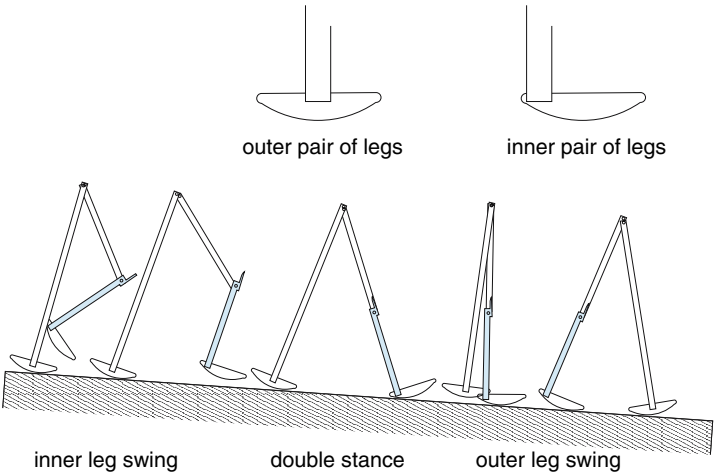


Fig. 1. Scheme of the changes on the feet of the 2D-walker with one knee. In order to prevent the walker stumbling over its stiffened leg the foot has been modified. *Left*: The attached point is shifted towards its middle. *Right*: Swinging is possible for both the kneed leg and the stiffened leg

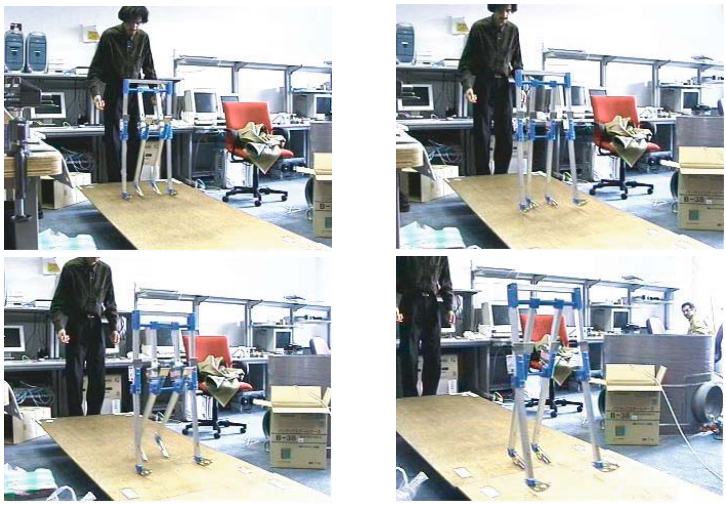


Fig. 2. 2-D Walker: Images are ordered from left to right and top to bottom, taken from test run of the walker

- an increase of the maximum stretch of the knee to more than 180 degrees. This enhances stability of the stretched knee and prevents the standing leg from bending.
- an increase of the weight of the upper leg. In this way the effective size of the upper leg was elongated, whereas the pendulum constant of the lower leg remained unchanged.

After the changes the walker showed a stable gait over the whole tested distance (about 15 steps) for a slope of around 3–5 degrees. The motion resembles now the motion of a person walking with crutches. The motion of our one-kneed walker appears to include two different step sizes which is due to the construction of the knee in the middle pair of legs and the stiffening of the outer legs.

At present the walker is brought by hand in the right set of initial conditions. With respect to theoretical considerations in [6] it would be interesting to find out the size of the basin of attraction. We see indications from our experiments that the stability of the walking is increased by stiffening one knee. The speed of the walker was around 0.5 m/s. The total weight of the walker is about 5 kg. Calculating the loss of potential energy per second one can calculate that the energy consumption is roughly 1 W. The walker seems to be able to walk by a slope from around 3 degrees to 8 degrees, however this depends of course strongly on the initial conditions, which at present are set by hand. The walker was presented at several exhibitions. Especially children could handle the starting conditions for the walker very well.

3 Stabilizing Dynamics with a Fast Rotating Gyro

The stabilizing effect of a gyro is well known. It has been applied to many technical application like toy motorcycles or satellites. If a force is applied from outside the gyro changes the direction in a different way than a solid body without rotation. The dynamics of the heavy gyro can be described by Euler-Lagrange equations.

They show that the axis of the rotation is stable if

- the axes of rotation coincides with on of the principal axes of the inertia tensor of the body and the component and
- the inertia tensor corresponding to this axes is either the biggest or the smallest component of the diagonal tensor [7].

In the following we present two variant walkers: First a passive dynamic walker with two degree of freedom in each leg. In this case the speed of the gyro is not controlled actively. And an active walker with two actuated degrees of freedom in each leg plus an active loop for balancing.

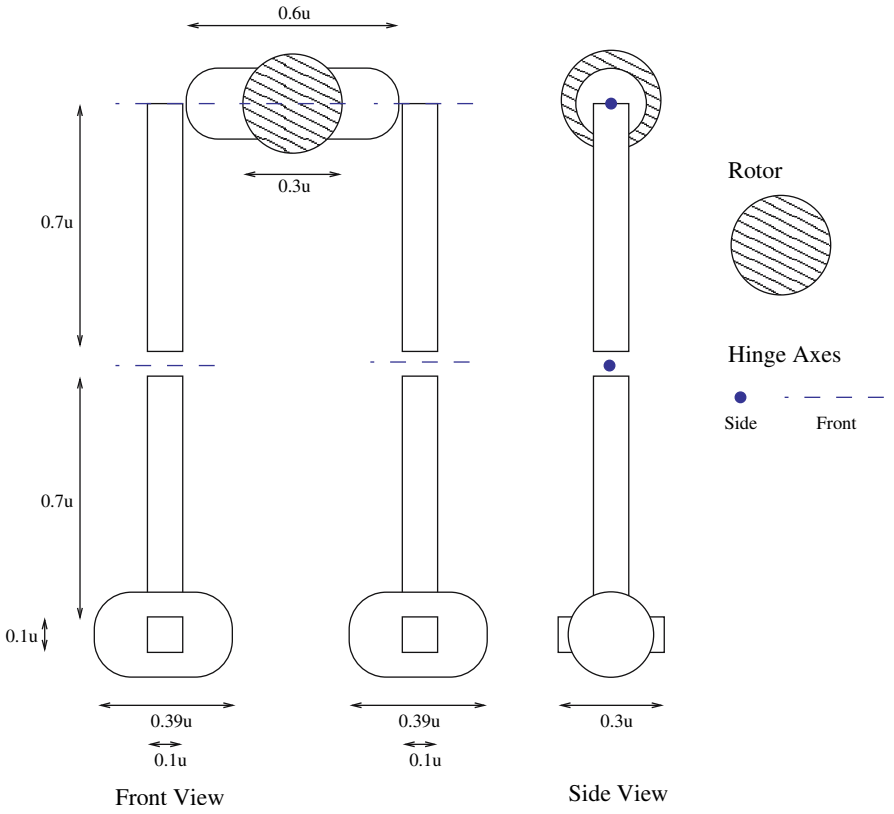


Fig. 3. Design of the walker in the simulation

In all our bipeds we used the gyro in that way that the axes of the gyro was parallel to the line from on hip joint to the other. The aim of the gyro is to improve the stability in the single-foot-stance phase. The gyro delays the falling aside. Thus the foot can be lifted for a longer time.

In addition, if the speed of the rotor can be controlled actively, the gyro can be used for the control of the pitch attitude. This can be done in a feedback control loop in connection with a sensor of the current pitch attitude. This was done in simulations with a robot with active joints.

As passive dynamic walker is simulated at a slope of 3 degree. The walking is stabilized by the gyro.

4 The Simulated Passive Dynamic Biped

For simulations we used the Open Dynamics Engine (ODE), which is an open source mechanics and dynamics simulator. The value of the gravitation was set to 9.81 units in both simulations. Thus, one time unit in the simulation can be interpreted as one second and one distance unit can be interpreted as one meter. the slope was set to 3.0 degree. The walker was not actuated except for a heavy gyro the rotated with a constant speed. The walker with a fast rotating gyro was tested against a walker with a non-rotating gyro. In the simulations the walker with the rotating gyro could walk some more steps then the walker with the non-rotating gyro. The walker was designed simple (please cf. 3). The masses of the parts were the following: Upper leg 0.07 kg, lower leg 0.012 kg, middle part (hip) 0.011 kg, rotor (heavy gyro) 0.565 kg. The speed of the gyro was 175 rad/s. The joints of the knees stop at an angle of 0.23 rad, which allows for a stable stance during walking. Except for the knee stop the joints are hinge joints without friction. The starting conditions are optimized manually.

For screen-shots of the simulated passive dynamic biped please cf. Figure 4. The walker is able to walk up to five steps in the simulation in the case of the rotating gyro. In the case of the non-rotating gyro the walker was not able to walk more than one to two steps.

5 Conclusion and Discussion

The work that was presented here consisted of two parts: The practical part was the construction of a real 2D Walker. The simulation work was about the idea to stabilize the walking by using a symmetric rotor or gyro.

- The real 2D walker was intended to give the authors a feeling for dynamic walking machines. The experiences could be used in the subsequent simulations. We consider the 2D walker as an intermediate step for understanding the walking process and to cope with the more difficult

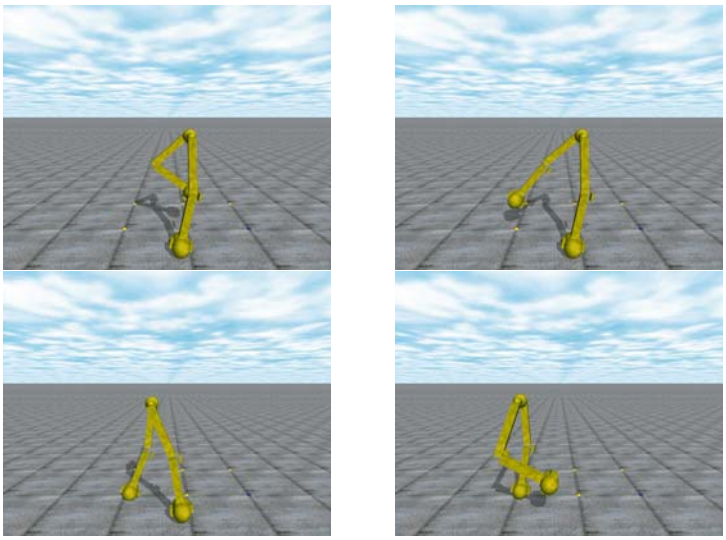


Fig. 4. Simulated biped

biped walker. This is true for the general 2D walker as it true for our modifications. Thus, our 2D walker is one additional alternative in the increasingly diverse zoo of this type of walkers.

- The usage of stabilizing gyros in dynamic walking is a new idea. The gyro stabilizes the direction of the axes of the hip, at least it changes the dynamics of the walker. In the work presented here the gyro is kept to constant speed. Balancing is still the key problem in biped walking. The gyro could be used to produce easier balancing conditions in the beginning of an unsupervised learning process. Step by step the speed of the gyro can be reduced in later stages of the learning process. While being far from any biological counterpart at present, the walker may be further improved by incorporating more realistic features such as moving arms or hips, and it may be tested in other walking regimes such as jogging or jumping.
- In addition the rotor can be used to control the pitch attitude, by increasing and decreasing the speed of the rotor (cf. Fig. 5). In this way moment can be added to the robot if needed in order to stabilize it. This can be used in the case of the dynamic walker as well as in the case of other legged robots. We have applied this in simulations to walkers with active legs (cf. Fig. 6.) These simulated biped robots are able to stand up and to jump. However, new problems arise: The time constant for adapting the speed of the rotor should not interfere with the movement of the walking. Further, very strong motors are needed to produce the torque and the speed of the rotor at the same time.
- The rotor with controlled speed can be seen as a new type of actuator. This actuator can be used to change the attitude of a robot and for dynamic

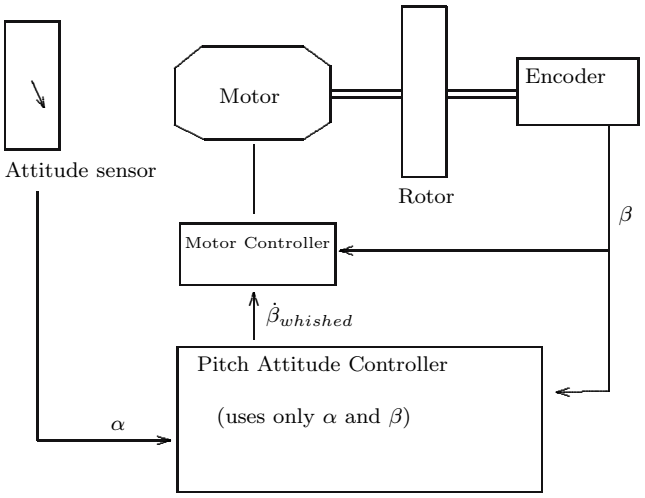


Fig. 5. Feet-back loop for pitch balancing: The controller uses only information about the attitude of the robot and the encoder values from the rotor as input

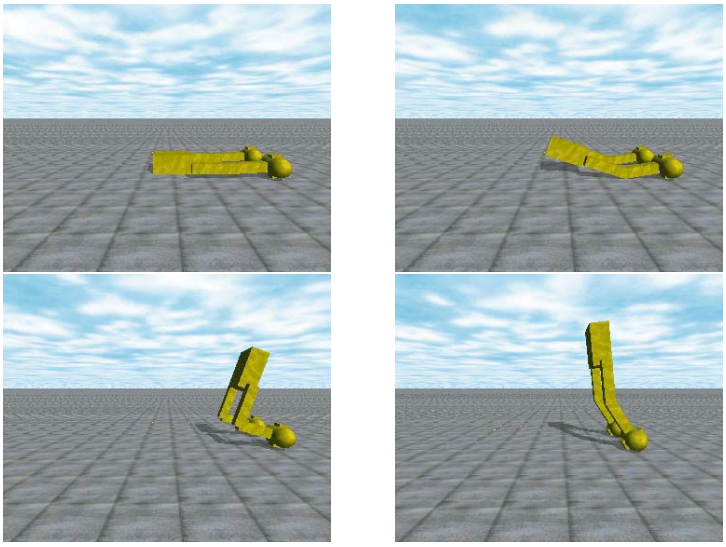


Fig. 6. Simulated actuated biped: The gyro is inside the box; controls the balance during getting up

balancing in a simple way. It seems possible use brakes in order to change the attitude of the robot rapidly. In addition the dynamics or the rotor allow for surprising movements that arise from basic physical principles. These ideas resulted in a project funded by NEDO for an artistic biped robot that is going to use this type of actuator.

Acknowledgments

We thank M. Wisse, J. M. Herrmann and M. Browne for helpful hints and dicussions.

References

1. T. Sugihara, H. Inoue, Real time Humanoid Motion Generation through ZMP Manipulation Based on Pendulum Control, IEEE Intl. Conference on Robotics and Automation, (ICRA 2002) pp. 1404–1409
2. J. Hass, J.-M. Herrmann, T. Geisel, Evolutionary design of an adaptive dynamic walker, submitted for this session at CLAWAR
3. S. Wischmann F. Pasemann, From passive to active dynamic 3D bipedal walking – an evolutionary approach-, submitted for this session at CLAWAR
4. T. McGeer. Passive Dynamic Walking. International Journal of Robotics Research 9:2 (1993) 62–82.
5. M. Garcia (1999) PhD Thesis. Cornell University.
6. A.L. Schwab, M. Wisse, Basin of Attraction of the simplest walking model, Proceedings of the DETC 2001, VIB-21363.
7. H. Goldstein, CP. Pole, JL. Safko, Classical Mechanics, 3rd Ed., Prentice Hall, 2002

Stability of a Spherical Pendulum Walker

J. Seipel

Department of Mechanical and Aerospace Engineering,
Princeton University, Princeton, NJ 08544, U.S.A.

Abstract. In this paper an inverted spherical pendulum walking model is analyzed to determine the stability of its three-dimensional walking gaits. Generally, "sprawled" side-to-side walking is stable, whereas "upright" walking is unstable. This model yields analytic results potentially useful for scientific or engineering studies.

1 Introduction

Are walking motions naturally stable? This question is compelling, often thought about, and yet still not fully understood. Recent work by Kuo et al. [1, 2] suggests that walking is inherently unstable. However, an anthropomorphic, passive mechanism has walked stably down a slope [3], and others, though not as anthropomorphic, have gone before [4, 5]. Even a simple walking toy can mystify us with its abilities.

When stability (or instability) is present in a walking experiment, the cause is often not clear. Furthermore, three-dimensional numerical studies have not answered or kept up with observed phenomena. For example, the walking mechanism of Collins, Wisse, and Ruina was designed more by "tinkering" with mechanical intuition than by analysis [3].

This paper is concerned with the stability analysis of three-dimensional walking gaits. It is hoped that the inverted spherical pendulum model captures the essential translational dynamics of walking, and thus, the results presented here might extrapolate to a more realistic rigid body problem with rotations; the subject of future work.

2 The Model and Stride Map

The inverted spherical pendulum model for walking consists of a point-mass atop a massless *stance leg* (Fig. 1), which contacts the ground at its foot

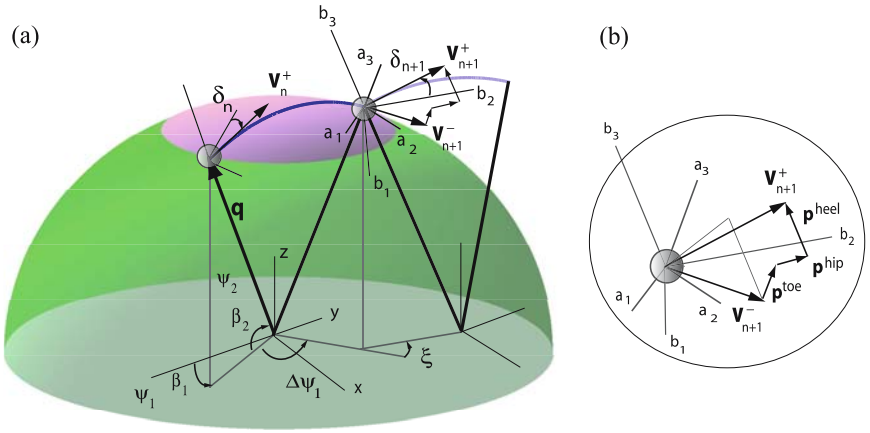


Fig. 1. (**a**) The inverted spherical pendulum walker, with coordinate systems. (**b**) A close-up view of the mapping from pre to post-impact velocity: $\mathbf{v}^-$ to $\mathbf{v}^+$

where it can pivot freely. The opposing *swing leg* is assumed to swing forward without colliding with the ground and to be held in position at angles β_1 and β_2 prior to impact, which occurs when the height of the mass falls to $\sin\beta_2$. In other words, the period of the swing leg is assumed to be less than the period of stance, and impact occurs in a predictable manner.

Walking gaits are composed of stance phases, during which the mass follows the motion of a spherical pendulum, and impact events, when support is transfered from one leg to the other (from the stance leg to the swing leg, upon which their labels switch). We consider a stride to be one stance phase followed by an impact event, after which another stride may ensue. Or, in the language of our mathematical model, the stride map P governing the evolution of the system's state from stride-to-stride is composed of stance and impact mappings:

$$P = P_{impact} \circ P_{stance} : [\delta]_n \rightarrow [\delta]_{n+1} \ , \tag{1}$$

where P_{stance} and P_{impact} are given below in (6) and (10) respectively. We need only track the angle δ (see Fig. 1) from stride-to-stride since the position at the beginning and end of stance is known from the leg geometry, and the velocity is perpendicular to the stance leg and has fixed magnitude at beginning and end. A complete walking gait, and thus a full stride map, would comprise two (left and right) strides, but we need only consider a single stride map P when investigating the stability of left-right symmetric gaits.

The Stance Map: The Lagrangian for the stance dynamics is, in nondimensional form, $L = \frac{1}{2}\dot{\psi}_2^2 + \frac{1}{2}\dot{\psi}_1^2 \cos^2\psi_2 - \tilde{g}\sin\psi_2$, which yields the following equations of motion:

$$p_{\psi_1} = \dot{\psi}_1 \cos^2\psi_2 = constant \ , \tag{2}$$

$$\ddot{\psi}_2 = -\dot{\psi}_1^2 \cos\psi_2 \sin\psi_2 - \tilde{g}\cos\psi_2 \ , \tag{3}$$

where $\tilde{g} = \frac{gL}{V^2} = \frac{E_{gravity}}{E_{kinetic}}$ is the nondimensional gravitational parameter (an inverse *Froude number*), and g, L, and V are the constants of the original system: acceleration due to gravity, leg length, and velocity magnitude at beginning of stance, respectively.

Using conservation of energy and angular momentum p_{ψ_1} during the stance phase, we can integrate (3) and then (2) to obtain the quadrature $\Delta\psi_1(\delta; \beta_2, \tilde{g})$, which gives us the location of the leg at the end of stance. In practice, $\Delta\psi_1$ can be found analytically or by numerical integration.

Now, since the height is equivalent at the beginning and end of stance and the total energy and angular momentum p_{ψ_1} are conserved, it follows that the angle δ, measured in the current leg frame, is the same at the both beginning and end of the stance phase. Thus the velocity vector in the (a_1, a_2, a_3) coordinate frame, that attached to the leg at the end of stance (see Fig. 1), is $\mathbf{v}^- = [-\sin\delta, \cos\delta, 0]$. However, the velocity vector is not so easily described in the frame of the *next* stance leg (b_1, b_2, b_3). For this we apply rotation matrices dependent on the angles β_2 and ξ:

$$\begin{aligned} v_{b_1}^- &= -\cos\xi\sin\delta + \sin\xi\cos\alpha_2\cos\delta\,, \\ v_{b_2}^- &= \cos\alpha_2\sin\xi\sin\delta + \cos^2\alpha_2\cos\xi\cos\delta - \sin^2\alpha_2\cos\delta\,, \\ v_{b_3}^- &= -\sin\alpha_2\sin\xi\sin\delta - \sin\alpha_2\cos\xi\cos\alpha_2\cos\delta - \sin\alpha_2\cos\alpha_2\cos\delta\,, \end{aligned} \tag{4}$$

where $\alpha_2 := \pi/2 - \beta_2$ is a parameter, and ξ is a function of δ:

$$\xi(\delta) := \pi - 2\beta_1 - \Delta\psi_1(\delta)\,. \tag{5}$$

The stance mapping, from δ in the initial leg frame to ξ and $\mathbf{v}_b^-$ in the new leg frame, is then:

$$P_{stance} : [\delta]_n \to [\xi, \mathbf{v}_b^-]_n\,. \tag{6}$$

The Impact Map: During impact, two impulses are imparted on the mass. A toe-off impulse $\mathbf{p}^{toe}$ along the axis of the pre-impact leg, and a heel-strike impulse $\mathbf{p}^{heel}$ along the next stance, or post-impact, leg. These impulses are analogous to those used in [6], only generalized to three-dimensions. The impact process is instantaneous compared with the stance dynamics, and we enforce conservation of momentum through impact:

$$\mathbf{v}^+ = \mathbf{v}^- + \mathbf{p}^{heel} + \mathbf{p}^{toe}\,, \tag{7}$$

where the impulse vectors in the (b_1, b_2, b_3)-frame are $\mathbf{p}^{heel} = p^{heel}[0, 0, 1]$, and $\mathbf{p}^{toe} = p^{toe}\left[\sin\alpha_2\sin\xi,\ (1+\tan\alpha_2)\cos^2\alpha_2\cos\xi,\ (1-\tan\alpha_2)\cos\alpha_2\right]$.

Support is successfully transfered to the post-impact leg when the post-impact velocity is perpendicular to it and has a magnitude of one: $v_{b_3}^+ = 0$, $|\mathbf{v}^+| = 1$, which, according to the right-hand-side of (7), is solved for the magnitudes of the toe-off and heel-strike impulses: $p^{toe} = \frac{-b+\sqrt{b^2-4ac}}{2a}$, and $p^{heel} = -v_{b_3}^- - p^{toe}$, where: $a = \frac{1}{2}\sin^2\alpha_2\sin^2\xi + \frac{1}{2}(1+\tan\alpha_2)^2\cos^4\alpha_2\cos^2\xi$,

$b = v_{b_1}^- \sin\alpha_2 \sin\xi + v_{b_2}^- (1+\tan\alpha_2)\cos^2\alpha_2 \cos\xi$, and $c = \frac{1}{2}(v_{b_1}^-)^2 + \frac{1}{2}(v_{b_2}^-)^2 - \frac{1}{2}$. Apart from $v_{b_3}^+$, which is necessarily zero, this yields:

$$\begin{aligned} v_{b_1}^+ &= v_{b_1}^- + p^{toe}(\xi, \mathbf{v}_b^-)\sin\alpha_2 \sin\xi \,, \\ v_{b_2}^+ &= v_{b_2}^- + p^{toe}(\xi, \mathbf{v}_b^-)\cos^2\alpha_2 \cos\xi + p^{toe}(\xi, \mathbf{v}_b^-)\cos\alpha_2 \sin\alpha_2 \cos\xi \,. \end{aligned} \quad (8)$$

The angle δ in the new stance frame is then found, by observation, to be

$$[\delta]_{n+1} = \tan^{-1}\left(-v_{b_1}^+ / v_{b_2}^+\right) \,. \quad (9)$$

Equation (9) along with (8), define the impact map

$$P_{impact} : [\xi, \mathbf{v}_b^-]_n \to [\delta]_{n+1} \,. \quad (10)$$

The Approximate Stride Map: To approximate the stride map, we assume that for ξ near zero, the map behaves as when $\xi = 0$, with impulses only along the b_3 and b_2 axes, and so (8) reduces to

$$v_{b_1}^+ = v_{b_1}^- \,, \quad v_{b_2}^+ = \sqrt{1 - (v_{b_1}^-)^2} \,. \quad (11)$$

Equation (11) along with (9) determine the approximate impact map: $P_{im,a} = \delta_{n+1} = \tan^{-1}(-v_{b_1}^-(1-(v_{b_1}^-)^2)^{-1/2})$. Substituting in $v_{b_1}^-$ from (4), the approximate stride map follows directly:

$$P_{approx} = \delta_{n+1} = \tan^{-1}\left(\frac{\cos\xi \sin\delta_n - \sin\xi \cos\alpha_2 \cos\delta_n}{\sqrt{1-(\cos\xi \sin\delta_n - \sin\xi \cos\alpha_2 \cos\delta_n)^2}}\right) \,, \quad (12)$$

where ξ, given in (5), is a function of δ_n . If $\xi = 0$, then $\delta_{n+1} = \delta_n$, and we have a fixed point of the approximate stride map. Evaluating the Jacobian of P_{approx} at symmetric fixed points ($\xi = 0$) yields:

$$DP_{approx} = 1 + \cos\alpha_2 \frac{\partial \Delta\psi_1}{\partial\delta} \,. \quad (13)$$

The Exact Stride Map: For the exact stride map, we find

$$P = \delta_{n+1} = \tan^{-1}\left(\frac{-v_{b_1}^- - p^{toe}\sin\alpha_2 \sin\xi}{v_{b_2}^- + p^{toe}\cos^2\alpha_2 \cos\xi + p^{toe}\cos\alpha_2 \sin\alpha_2 \cos\xi}\right) \,, \quad (14)$$

where $v_{b_1}^-$, $v_{b_2}^-$, ξ, and p^{toe} are functions of δ_n. Again, $\xi = 0$ or $\Delta\psi_1 = \pi - 2\beta_1$ is a sufficient condition for fixed point solutions. Evaluating the Jacobian at general fixed points does not lead to a reduced expression. However, evaluating at the special case of upright walking solutions ($\xi = 0$, $\delta = 0$), we find

$$DP = 1 + h(\alpha_2)\frac{\partial \Delta\psi_1}{\partial\delta} \,, \quad (15)$$

where $h(\alpha_2) \approx \cos\alpha_2$ for small α_2, in agreement with (13).

Results

By observation, if $\xi = 0$, then $\delta_{n+1} = \delta_n$ for both the approximate and exact stride maps: (12) and (14) respectively. Therefore, $\xi = 0$ is a sufficient condition for fixed points, and from (5) can be rewritten as

$$\Delta\psi_1(\delta; \beta_2, \tilde{g}) = \pi - 2\beta_1 \ , \tag{16}$$

in terms of the parameter β_1 and quadrature $\Delta\psi_1$, which is itself a function of the state δ and parameters β_2 and $\tilde{g}$. Thus (16) depends on three parameters, $(\beta_2, \beta_1, \tilde{g})$, making it difficult to visualize solutions. However, noting that small changes to β_2 does not alter the qualitative features of the fixed point sets, β_2 can be held fixed (where $\beta_2 = 65\pi/180$ for this study). The quadrature $\Delta\psi_1$ is then plotted over the $(\delta; \tilde{g})$ space resulting in a three-dimensional surface. The quadrature surface implicitly defines a unique two-parameter family of fixed points $(\delta; \tilde{g}, \beta_1)$ satisfying (16), with β_2 held constant.

Figure 2 shows the quadrature and corresponding eigenvalue surfaces for both approximate and exact stride maps. Generally, the exact eigenvalue is slightly more unstable than the approximate eigenvalue, but they are otherwise in good agreement. If we follow along the curve where $\delta = 0$ (the case of upright walking) and thus $\Delta\psi_1 = \pi$ for all $(\beta_2, \tilde{g})$, we find instability for both the approximate and exact case. However, sprawled side-to-side gaits are stable, when δ is sufficiently large.

Conclusion

We have derived an explicit stride map (a Poincaré return map) for a three-dimensional, point-mass walker and demonstrated that left-right symmetric

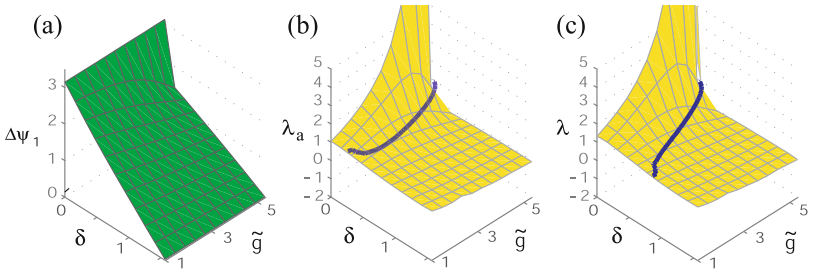


Fig. 2. (**a**) The quadrature $\Delta\psi_1$ over a range of the parameter $\tilde{g}$ and input δ, (**b**) the eigenvalue for the approximate map P_a, and (**c**) the eigenvalue for the exact map P. Here $\beta_2 = 65\pi/180 = constant$ is used. Note that $\lambda_a = -DP_{approx}$ and $\lambda = -DP$, for ease of display in (b) and (c) respectively. The thick solid curve is a level set at $\lambda = 1$, above which gaits are unstable. If we follow along the curve where $\delta = 0$ (the case of upright walking) and thus $\Delta\psi_1 = \pi$ for all $(\beta_2, \tilde{g})$, we find instability. However, sprawled side-to-side gaits are stable, when δ is sufficiently large

gaits are both stable and unstable depending on parameter choice (i.e., type of gait). The map and its linearization are appealingly simple, especially for the approximate case. In general, sprawled side-to-side gaits are stable and upright gaits are unstable.

Acknowledgements

The author was supported by an NSF fellowship. Discussions with Philip Holmes and Darryl Thelen were helpful in the preparation of this paper.

References

1. A.D. Kuo. Stabilization of lateral motion in passive dynamic walking. *Int. J. Rob. Res.*, 18 (9):917–930, 1999.
2. C.E. Bauby and A.D. Kuo. Active control of lateral balance in human walking. *J. Biomech.*, 33(2000):1433–1440, 2000.
3. S.H. Collins, M. Wisse, and A. Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *Int. J. Rob. Res.*, 20 (7):607–615, 2001.
4. M.J. Coleman and A. Ruina. An uncontolled walking toy that cannot stand still. *Phys. Rev. Lett.*, 80:3658–3661, 1998.
5. M.J. Coleman, M. Garcia, K. Mombaur, and A. Ruina. Prediction of stable walking for a toy that cannot stand. *Phys. Rev. E.*, 64:022901–1–3, 2001.
6. A.D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *Journal of Biomechanical Engineering*, 124:113–120, 2002.

A CLAWAR That Benefits From Abstracted Cockroach Locomotion Principles

T.E. Wei[1], R.D. Quinn[1], and R.E. Ritzmann[2]

[1] Mechanical Engineering and
[2] Biology, Case Western Reserve University, Cleveland, Ohio, USA

Abstract. MechaRoach II is a hexapod robot under development that will walk on horizontal surfaces, climb on inclined or vertical mesh surfaces, and test strategies for transitioning between the two. The locomotion principles that allow cockroaches to make these transitions have been studied and mechanisms using abstractions of those principles have been developed for the robot. These principles include leg and foot morphology, gait adaptation, and body flexion. MechaRoach II will have a single drive motor, a motor for steering, and a motor to actuate a body flexion joint. The single drive motor will power all six legs, and each leg will use 4-bar mechanisms to recreate cockroach-like foot trajectories. Cockroaches have been shown to flex their bodies downward between the first and second thoracic segments or rear their bodies upward using their middle legs during transitions between climbing on vertical surfaces and walking on horizontal ones. Similarly, MechaRoach II's body joint will be used to rear the front of the robot upward or downward during transitioning. The robot will normally walk in a tripod gait with contralateral legs 180 degrees out of phase, but will use passive torsionally compliant devices to bring contralateral legs into phase for climbing.

1 Introduction

Robots with the ability to climb inclined or vertical surfaces have numerous applications. Legged robots have already been developed to climb on the underside of bridges [1] and crawl on the insides of pipes [7] for inspection purposes. Other legged robots have been developed for jobs such as weld inspection in nuclear plants [18]. Robots are the ideal choice for many of these applications because the working environments can be either poor or hazardous for humans.

In the process of these tasks, a climbing robot may have to negotiate a transition between surfaces that are at different inclinations, such as between a horizontal surface and a vertical one. Successful negotiation of that transition will require more than just the ability to adhere to the surfaces. Legged robots that can walk on horizontal surfaces, climb vertical surfaces, and make the

transition between the two include Robug IIs [13], Robug III [12], Ninja-1 [9], and Ninja-II [8]. These robots take advantage of a number of design features, such as legs that allow for larger ranges of motion than a normal walking robot would have.

Cockroaches have been shown to employ many different locomotion principles when they switch between walking on a horizontal surface and climbing on a vertical or inclined one, such as features of leg morphology, gait adaptation, and body flexion. It also includes foot morphology such as pads on their tarsi that allow them to grip a variety of surfaces [11] and an actuated claw. Mechanical devices that employ these principles can be used to allow robots to successfully negotiate transitions between surfaces of different inclinations.

Exactly reproducing the behavior of a biological system may be overly complex and may necessitate the development of new technologies. Abstracting the useful biological principles using simpler mechanical devices can more quickly result in mission-capable robots [2]. Other robots that use abstracted biological locomotion principles include MechaRoach I [4], the Whegs robots [2], PROLERO [14] and RHex [16]. The design of MechaRoach II benefits from biological principles that cockroaches use to walk and climb.

The chassis of MechaRoach II has been fabricated and is shown in Fig. 1. It is 70 cm long, 40 cm wide, 10 cm tall and it weighs 1.5 kg. This was a convenient size to manufacture, although the concept can be scaled up or down in size. The frame is made of a network of thin aluminum supports, which allows it to be relatively lightweight.

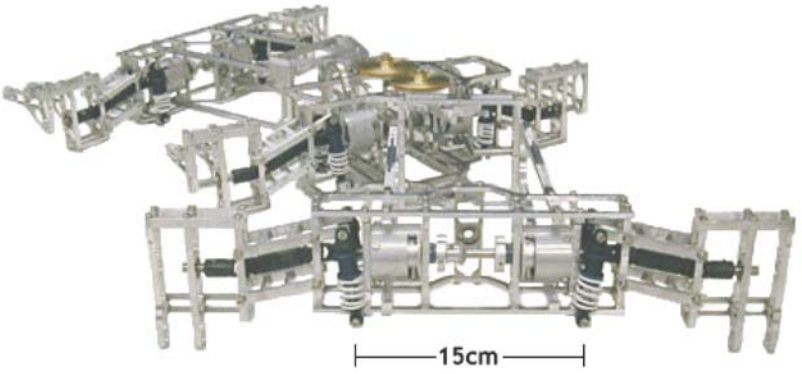


Fig. 1. The chassis of MechaRoach II. It is 70 cm long, 40 cm wide, 10 cm tall and weighs 1.5 kg

MechaRoach II will have three motors: One for propulsion, one for steering and one for vertical body flexion. To turn, the robot has body joints that permit the front and rear segments of the robot to pivot left and right independently of the middle segment, as shown in Fig. 1. This motion is

independent of the vertical body flexion, which permits the front half of the robot to rotate up and down 45 degrees relative to the middle and rear segments. This large range of motion will ease the vehicle's transition to and from horizontal and vertical surfaces.

2 Leg Design

Figure 2 shows a series of photographs of the front of a cockroach as it moves its front leg during normal walking. It normally raises its front legs high so that it can climb over small barriers without changing its gait. The last illustration in Fig. 2 is a composite with the five leg positions overlaid. Although this movement would not have reached the top of the 11 mm block shown here, it would have been sufficient for a 5.5 mm barrier [17].

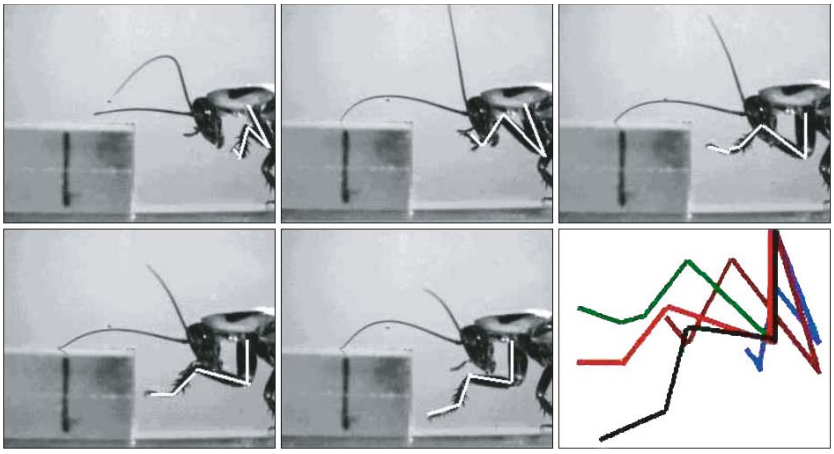


Fig. 2. Normal cockroach front leg trajectory. A cockroach normally raises its front legs high so that it can reach the top of small barriers without changing its gait. The foot trajectory shown above would have been sufficient to reach the top of a 5.5 mm barrier, but not the 11 mm barrier shown

Previous robots, such as Boadicea [3], have used cockroach footpaths before. MechaRoach I imitated cockroach foot trajectories, and performance evaluations showed that cockroach-like mobility resulted. MechaRoach I used 4-bar mechanisms in order to recreate those foot trajectories, and could climb over obstacles up to 70% of its own height.

MechaRoach II will also use 4-bar mechanisms to create cockroach-like leg trajectories. 4-bar mechanisms can be driven with a continuously rotating shaft input, while allowing different legs to be specialized for different tasks. The front legs have been designed to raise high during the swing phase so that

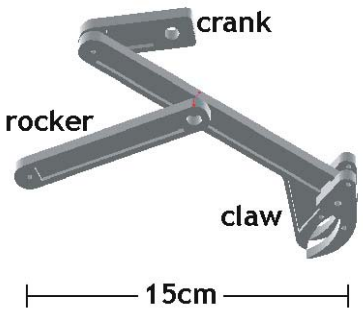


Fig. 3. A CAD drawing of the front leg of MechaRoach II. The front leg is a crank and rocker 4-bar mechanism, with the crank being driven by a continuously rotating driveshaft, and the rocker attached to the chassis through a pivot point

obstacles can be climbed without a change in foot trajectory. The rear legs have been designed to generate propulsive forces similar to those measured in cockroach locomotion as described by Full [6].

Figure 3 is a CAD drawing of MechaRoach II's front leg. The leg is a 4-bar crank and rocker mechanism that will be attached to the chassis of the robot at two points – with the crank attached to the driveshaft at one point, and the rocker attached though a pivot at the other point.

When the driveshaft applies torque to the input link of the leg, the foot will move in a trajectory based on the observed behavior in cockroaches. The motion will allow the robot to walk, while also allowing it to climb over obstacles.

Cockroach legs have inherent passive compliance and damping. As an illustration of this, a large impulsive force that is too rapid to be counteracted by the nervous system can be absorbed by the passive compliance of the legs [10]. The legs stabilize the cockroach and vibrations of the body are absorbed by their damping properties. Furthermore, these passive properties can reduce the forces transmitted from the ground to the body.

MechaRoach II has been constructed with a suspension system inboard of its legs. Shock absorbers with adjustable bump and rebound damping were placed in series between the frame and the legs as shown in Fig. 4. The suspension will be tuned to provide the desirable damping properties. One advantage of this configuration is that the rate of compliance is independent of leg orientation.

3 End Effection

Cockroach tarsi have a number of features that allow them to grip a large variety of surfaces. Among those features are an actuated claw and spines (see Fig. 5) [5].

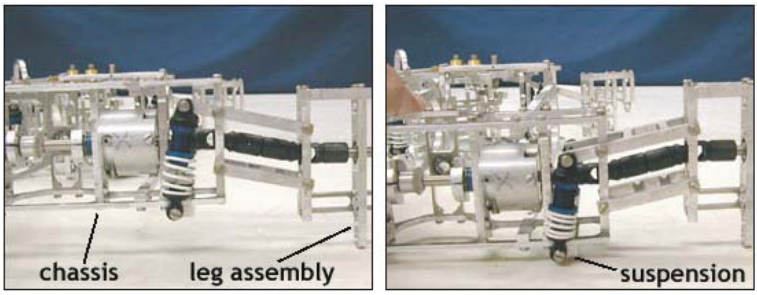


Fig. 4. MechaRoach II's passive leg compliance and damping. On the *right*, the suspension is fully extended. On the *left*, it is fully compressed. The compliance and damping will stabilize the body and reduce forces transmitted to it from the ground

Fig. 5. The claw and spines of a cockroach tarsus (image courtesy of Sasha Zill)

One of the advantages to using 4-bar mechanisms as legs, when compared to continuously rotating devices such as wheels, is the ability to more easily mount other mechanisms to them to aid in gripping surfaces. Like cockroach tarsi, MechaRoach II will have claws at the end of each of its legs (see Fig. 3), which will enable the robot to grasp and climb meshed or coarse substrates. The robot claw will be spring-loaded so that it is normally closed, and a tendon will open the claw at the appropriate moment in its trajectory.

4 Gait Adaptation

Cockroaches typically walk in an alternating tripod gait in which the front and rear legs on one side support the body at the same time as the middle leg on the other side [19]. This tripod of support alternates with the tripod formed by the other three legs.

Cockroaches do not always maintain a tripod gait. When they climb larger obstacles they alter their gait such that their climbing ability and stability are improved [17]. For example, as illustrated in Fig. 6, the front, middle and rear

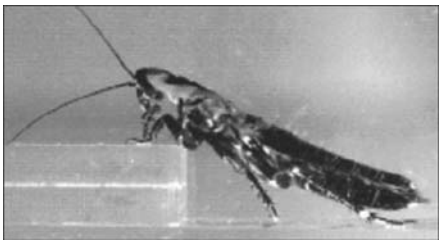


Fig. 6. Cockroach leg pairs move in phase when climbing larger obstacles

leg pairs of the cockroach are each in-phase and all supporting the body during a climb over an 11 mm obstacle.

This gait change can be created in the robot through use of passive torsionally compliant devices [2]. MechaRoach II will normally walk in a tripod gait, in which contralateral legs are 180 degrees out of phase. However, when the loading of one leg increases significantly, as would happen when the robot is climbing, the torsionally compliant devices will allow the contralateral leg to come into phase. After this occurs, both legs can work together to share the effort needed to lift the robot up to or over an object.

5 Body Flexion

As the cockroach climbs over various barriers its posture changes to maintain appropriate leg angles. When making the transition between a vertical and horizontal surface, a cockroach can flex its body downward at the joint between the first and second thoracic segments (Fig. 7a). This keeps the legs close to the top surface of the block and prevents unstable actions such as high centering. If flexion of that joint is prevented, the animal high centers and the legs are greatly extended (Fig. 7b).

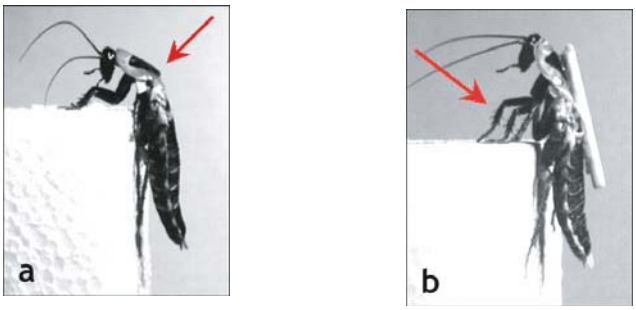


Fig. 7. (**a**) Body flexion at the top of a block. (**b**) Body flexion is prevented by a splint, and the cockroach high centers and has to extend its legs much further. Adapted from [15]

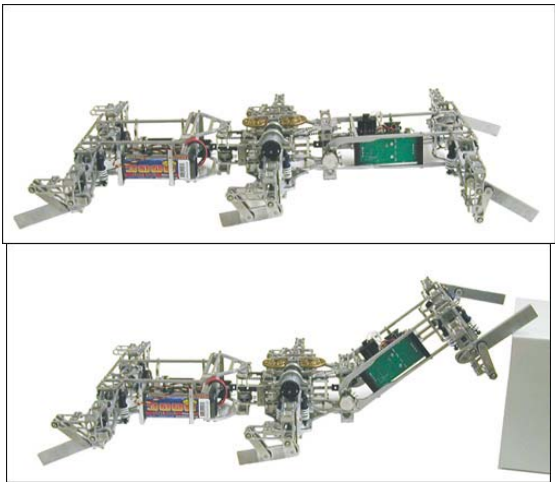


Fig. 8. MechaRoach II's actuated body flexion joint when unflexed (*left*), and activated to allow the robot to reach the top of an obstacle (*right*). The legs shown are models and are not indicative of actual leg geometry

Body flexion joints have been implemented in robots before, and have been shown to extend a robot's climbing ability [2]. A body flexion joint will be used in MechaRoach II in a similar way to rear the front of the robot downward during transitions.

When faced with a barrier higher than a cockroach's front legs can normally reach, it rears its entire body upward by actions of its front and middle legs [17]. The rearing action extends the reach of the front legs, enabling the cockroach to climb over larger obstacles.

MechaRoach II's middle legs will not be able to rear the body of the robot up in a similar way. However, the body flexion joint can be used to rear the front of the robot upward in order to extend the reach of its front legs (Fig. 8).

6 Conclusions

MechaRoach II is under development and will be used to test strategies for a robot transitioning between surfaces of different inclinations, such as between horizontal and vertical surfaces. Locomotion principles, which have been abstracted from the study of cockroaches, will be implemented to aid in climbing and transitioning. The abstracted locomotion principles include leg and foot morphology, gait adaptation, and body flexion. 4-bar mechanisms will create foot trajectories similar to a cockroach's, and allow for the specialization of legs for particular tasks.

Acknowledgements

This work was supported by United States Air Force contract F08630-01-C-0023 and the National Science Foundation through the NSF/IGERT Neuromechanics grant DGE 9972747.

References

1. Abderrahim M, Balaguer C, Gimenez A, Pastor JM, Padron VM (1999) ROMA: A Climbing Robot for Inspection Operations. Proceedings of the IEEE International Conference on Robotics and Automation, Volume 3, 10–15 May 1999
2. Allen TJ, Quinn RD, Bachmann RJ, Ritzmann RE (2003) Abstracted Biological Principles Applied with Reduced Actuation Improve Mobility of Legged Vehicles. IEEE International Conference On Intelligent Robots and Systems, Las Vegas, Nevada
3. Binnard M (1995) Design of a Small Pneumatic Walking Robot. MIT. M.S. thesis
4. Boggess MJ, Schroer RT, Quinn RD, Ritzmann RE (2004) Mechanized Cockroach Footpaths Enable Cockroach-like Mobility. Proceedings of IEEE International Conference on Robotics and Automation, New Orleans, LA, 26 April-1 May, 2004
5. Frazier SF, Larsen GS, Neff D, Quimby L, Carney M, DiCaprio RS, Zill SN (1999) Elasticity and movements of the cockroach tarsus in walking. Journal of Comparative Physiology, 185:157–172
6. Full RJ, Blickhan R, Ting LH (1991) Leg Design in Hexapedal Runners. Journal of Experimental Biology, 158: 369–390
7. Galvez JA, Conzalez de Santos P, Pfeiffer F (2001) Intrinsic Tactile Sensing for the Optimization of Force Distribution in a Pipe Crawling Robot. IEEE/ASME Transactions on Mechatronics, Vol. 6, No. 1
8. Hirose S, Kawabe K (1998) Ceiling Walk of Quadruped Wall Climbing Robot. Proceedings of the First International Conference on Walking and Climbing Robots, Brussels
9. Hirose S, Nagakubo A, Toyama R (1991) Machine That Can Walk and Climb on Floors, Walls and Ceilings. Proceedings of the IEEE International Conference on Robots in Advanced Robotics, 19–22 June 1991
10. Jindrich DL, Full RJ (2002). Dynamic Stabilization of Rapid Hexapedal Locomotion. Journal of Experimental Biology, Vol 205, 2803–2823
11. Jiao Y, Gorb S, Scherge M (2000) Adhesion Measured on the Attachment Pads of *Tettigonia viridissima* (orthoptera, insecta). Journal of Experimental Biology, Vol 203: 1887–1895
12. Luk BL, Collie AA, Piefort V, Virk GS (1996) Robug III: A Tele-operated Climbing and Walking Robot. UKACC International Conference on Control, Volume 1, Number 427
13. Luk BL, Cooke DS, Collie AA, Hewer ND, Chen S (2001) Intelligent Legged Climbing Service Robot for Remote Inspection And Maintenance in Hazardous Environments. Proceedings of the IEEE Conference on Mechatrinics and Machine Vision in Practice, Hong Kong

14. Martin-Alvarez A, de Peuter W, Hillebrand J, Putz P, Mattheyssen A, de Weerd JF (1996) Walking Robots for Planetary Exploration Missions. Second World Automation Congress (WAC '96). May 27–30, 1996. Montpellier, France
15. Rizmann RE, Quinn RD, Fischer MS (2004) Convergent Evolution and Locomotion Through Complex Terrain by Insects. To appear in Arthropod Structure and Development, Special Issue on Arthropod Locomotion and Biorobotics
16. Saranli U, Buehler M, Koditschek D (2001) Rhex: A Simple and Highly Mobile Hexapod Robot. International Journal of Robotics Research, 20(7): 616–631
17. Watson JT, Ritzmann RE, Zill SN, Pollack AJ (2002) Control of Obstacle Climbing in the Cockroach, *Blaberus discoidalis*: I. Kinematics. Journal of Comparative Physiology, Vol. 188: 39–53
18. White TS, Hewer N, Luk BL, Hazel J (1998) The Design and Operational Performance of a Climbing Robot Used for Weld Inspection in Hazardous Environments. Proceedings of the IEEE International Conference on Control Applications, Trieste, Italy
19. Wilson DM (1966) Insect walking. *A. Rev. Entom.* 11: 103–123

iSprawl: Autonomy, and the Effects of Power Transmission

Sangbae Kim, Jonathan E. Clark, and Mark R. Cutkosky

Department of Mechanical Engineering, Stanford University, Stanford, CA 94305
cutkosky@stanford.edu

Abstract. We describe the design features that underlie the operation of *iSprawl*, a small (0.3 Kg) autonomous, bio-inspired hexapod that runs at 15 body-lengths/second. These features include a light and flexible power transmission system that permits high speed rotary power to be converted to periodic thrusting and distributed to the tips of the legs, and a tuned set of leg compliances for efficient running. Examination of the trajectory of the center of mass and the ground reaction forces for *iSprawl* show that it achieves the same stable, bouncing locomotion seen in insects and in previous (slower) bio-inspired robots.

1 Introduction

In recent years a number of fast, legged robots have been developed that draw their inspiration from running arthropods including *Sprawlita* [5], *Scorpion* [8], *Whegs* [13], and *RHex* [14]. When insects are moving rapidly they typically employ an alternating tripod gait and they rely heavily on passive mechanical properties to achieve dynamic stability. The sprawled posture with large forces in the horizontal plane, and the compliance and damping in the limbs and joints, serve as "preflexes" [9, 11] that promote stable running and rapid recovery from perturbations.

In the case of the *Sprawl* family of robots, the main principles adapted from insects, and the cockroach in particular, are:

- a sprawled posture
- a bouncing, alternating tripod gait based on an open-loop motor pattern
- specialization in which the rear legs primarily accelerate the robot while the front legs decelerate it at the end of each step
- actuators that provide propulsion by thrusting along the axes of the legs
- passive "hip" joints that swing the legs forward between steps
- compliance and damping that absorb perturbations.

The *Sprawl* robots are fabricated using a multi-material rapid prototyping process, Shape Deposition Manufacturing [1, 12], that makes it possible to

achieve local variations in structural compliance and damping and to embed components such as sensors and actuators for increased ruggedness. Like their exemplars, the *Sprawl* robots are capable of fast locomotion over belly-height obstacles. Speeds of 7 bodylengths/s (1.0 m/s) have been achieved as well as rapid turns (4.0 rad/s) while running at full speed. The robots can run without any proprioceptive or exteroceptive feedback; however, the addition of ground contact sensors allows the stride period to adapt automatically to changes in terrain or slope [3]. A closer look at the dynamics of the running robots reveals motions and ground reaction forces similar to those found in insects and other small animals. The locomotion pattern has been termed SLIP (spring loaded inverted pendulum) in the literature and is seen in many jogging animals [7].

A limiting factor in the design of the *Sprawl* robots has been their use of pneumatic pistons for propulsion. Although electric motors are ubiquitous in small robots, pistons were chosen for the *Sprawl* robots as powerful, compact linear actuators. The main disadvantage to pneumatic pistons is of course that they virtually preclude autonomous operation. The volume of compressed gas needed for 10 minutes of operation is such that a gas storage tank would be too heavy to carry on board.

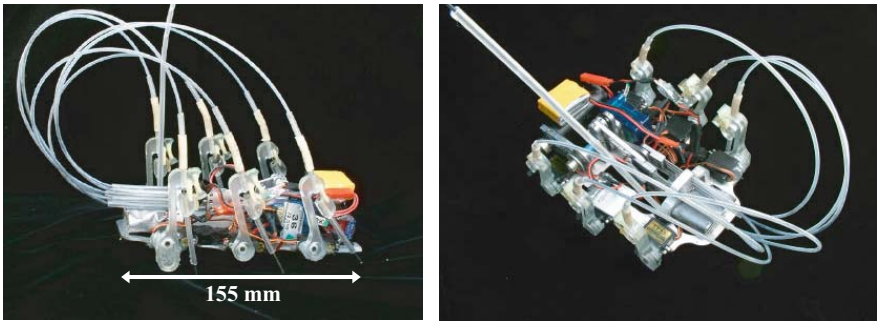


Fig. 1. *iSprawl*: a fully autonomous hexapedal robot driven by an electric motor and flexible push-pull cables

In this paper, we present an independent version of the *Sprawl* robots utilizing electric propulsion. The incorporation of a new power transmission system, lithium polymer batteries, and a redesigned set of complaint legs have enabled *iSprawl* to run autonomously at speeds of over 15 bodylengths/second (2.3 m/s, or a Froude number of 3.5. Despite significant changes in the actuation and force generation mechanism, we show that by appropriately tuning the passive compliance in the legs the fast, self-stabilizing behavior of the robot is preserved. This invariance to actuation scheme underscores the generality of the locomotion principles encapsulated in the *Sprawl* family of robots.

2 Mechanical Design of *iSprawl*

The most challenging aspects of utilizing electrical actuation for the *Sprawl* robots are converting rotary to linear motion and incorporating sufficient flexibility into the power train to accommodate the repositioning of the compliant legs. Several schemes were investigated before settling on the system presented in this paper. One possibility is to have a motor put energy into an elastic storage device that is released with each step.

A second possibility is to store kinetic energy, in a flywheel or other rotating mass, which can be tapped at various points during the stride period. This is the approach, as shown in Fig. 2a, that was ultimately adopted for *iSprawl*.

A particular challenge of the *iSprawl* design is that power must be conveyed to the distal ends of the limbs, which are flexing back and forth with each stride. By utilizing central power source and a lightweight and flexible transmission system, the rotational inertia of the limbs can be minimized, which in turn permits a faster stride frequency. Several transmission mechanisms were explored; ultimately, best results were obtained with the cable system shown in Fig. 2b. By adding rigid elements to both ends of the shaft and tube, the cables are able to thrust as well as pull.

As in previous *Sprawl* robots, the motions of the legs back and forth with each step are achieved passively by operating the robot as a resonant system. RC servos are mounted at the hips of the middle legs to change the equilibrium leg angles to effect turns, as motivated by the results of [10]. The physical specifications for *iSprawl* are given in Table 1.

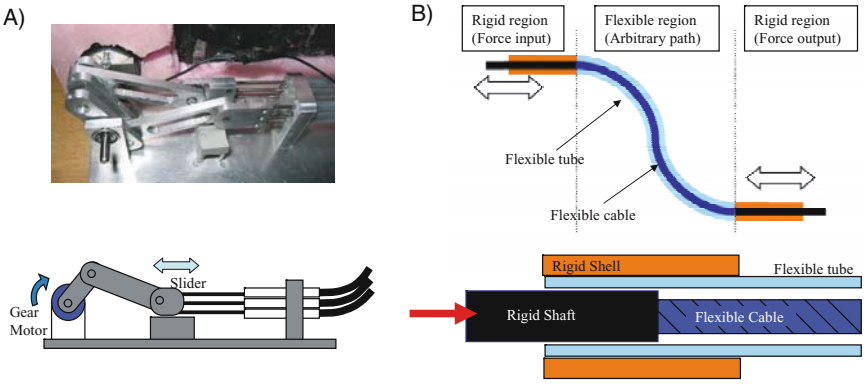


Fig. 2. Power transmission system for *iSprawl*. Figure (**a**) shows the crank-slider used to store and convert the rotational energy from the motor to linear oscillations. Figure (**b**) schematically shows the flexible and rigid sections ocF the push-pull cables

Table 1. Physical Parameters for *iSprawl*

Body size	155 × 116 × 70 mm (excluding cables)
Body mass	315 g (including batteries and RC circuit)
Maximum speed	2.3 m/s (15 bodylengths/s)
Stride frequency	17 Hz
Power consumption	12 W (0.5 A, 24 V)
Motor	Maxon 2.7 W 2023909; size: 20 × 17.5 × 8 mm
Gear ratio	20:1
Legs	Polyurethane 72DC and 90A from Innovative Polymers
Servo motor	Cirrus cs-5.4g
Typical leg motion	25 mm stroke, 25° swing
Battery	6 pack of lithium polymer (3.7 V, 250 mAh)

3 Transmission Effects on Locomotion

Early experiments with cable-driven *iSprawl* revealed that vertical foot forces increased too rapidly after inital contact. This caused abrupt changes to the momentum of the robot, increasing wear and reducing efficiency. This is not surprising given that we have replaced a compliant force actuator (pneumatics) with a displacement actuation from the slider-crank mechanism. To achieve a smoother, more SLIP-like motion, it was necessary to add some tuned axial compliance to the push/pull cables, as shown in Fig. 3.

3.1 Leg Extension Profile and SLIP-Like Motion

We desire SLIP-like running for the robot, in which the center of mass moves along an approximately sinusoidal trajectory, as shown in Fig. 4. Using

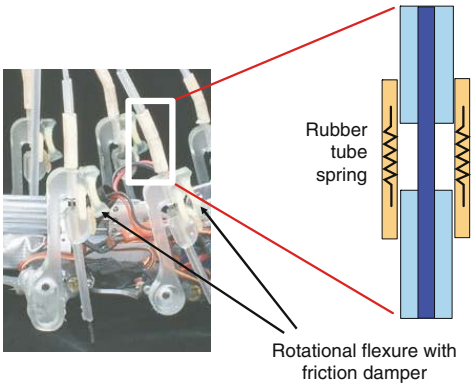


Fig. 3. Schematic of the leg compression spring design utilizing a tension spring on the flexible tubing around the push-pull cable. Also shown are the frictional dampers on the front and middle legs

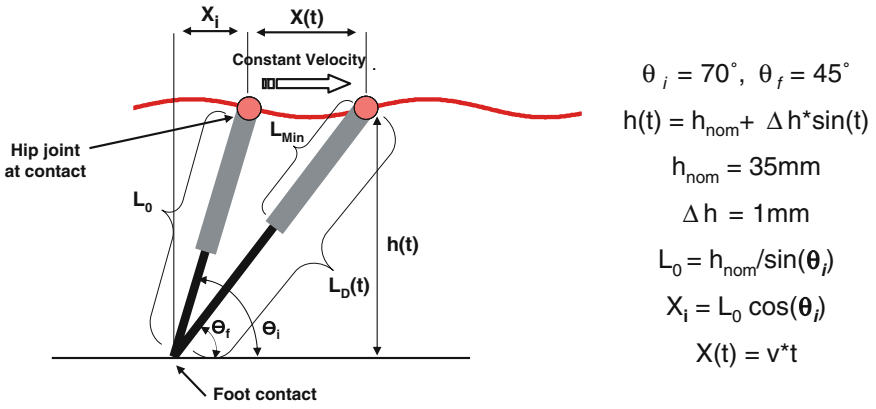


Fig. 4. Schematic of the desired leg extension profile needed to produce a sinusoidal trajectory of the center of mass during stance

the geometric relationships depicted in Fig. 4, we can calculate the desired axial stiffness of the legs to minimize body accelerations at contact while maintaining a desired level of thrust. The values of the constants in Fig. 4 were measured experimentally. If the body is assumed to move along a sinusoidal path during contact, the desired leg length $L_D(t)$ is given by:

$$L_D(t) = \sqrt{h(t)^2 + (x_i + x(t))^2} \tag{1}$$

For *iSprawl* the nominal leg extension profile $L_{nom}(t)$ is fixed as:

$$L_{nom}(t) = A_0 \; sin(\omega t) \quad \text{where } A_0 = 12.5\,\text{mm} \tag{2}$$

The leg compression L_s is given by the difference between these and is:

$$L_s(t) = L_{nom}(t) - (L_D(t) - L_0) \tag{3}$$

We can then tune the stiffness of the spring such that the vertical energy at impact equals the potential energy stored in the spring at maximum compression. Due to the geometry and pitching dynamics of *iSprawl* the gravitational potential energy and rotational kinetic energy are non-negligible, and the total vertical energy at contact (E_{impact}) is 0.026 Nm. If we approximate the desired maximum compression of the spring to equal the maximum of $L_s(t)$, with the potential energy stored in the spring being:

$$PE_{spring} = \frac{1}{2} k \, (\Delta L)^2 \tag{4}$$

where $\Delta L = max(L_s(t)) = 4\,\text{mm}$, then the desired stiffness for a tripod is:

$$k = \frac{E_{impact}}{\frac{1}{2}(\Delta L)^2} = 3.3\ \text{N/mm} \tag{5}$$

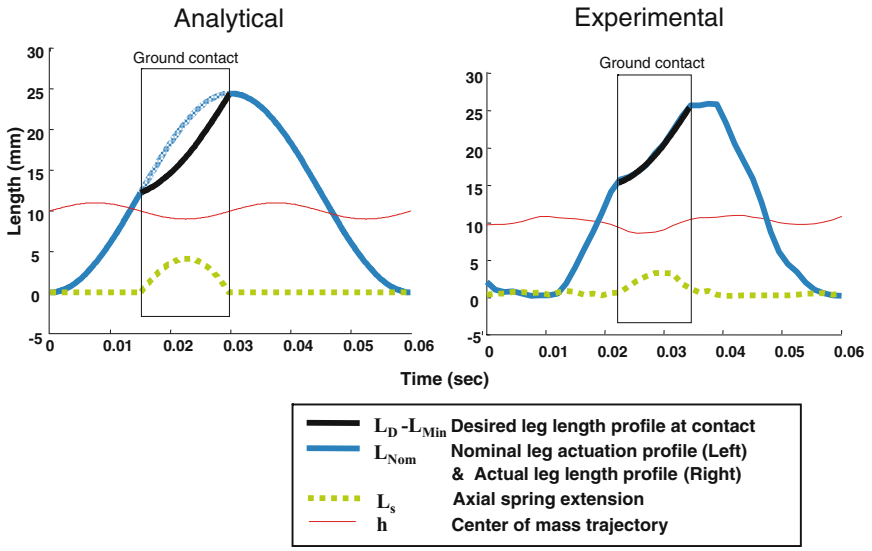


Fig. 5. The theoretical and experimental leg extension profiles for *iSprawl*. Also shown are the path of the COM and the extension of the axial spring for each case

The front leg has the largest contribution (about 50%) to the vertical stiffness of the tripod. Accordingly, springs with a stiffness of 1.7 N/mm were inserted into each leg.

Figure 5 shows the theoretical and the measured leg trajectories. The trajectories for the measured case were obtained by filming *iSprawl* at 500 frames/second as it ran on a treadmill. The dark lines represent the desired leg extension profile during contact, and the thin lines represent the trajectory of the center of mass. The dotted segment in the analytical plot indicates the center of mass trajectory that would occur without the leg spring, whose compression is indicated by the dashed line at the bottom of the plot. The experimental data show that both the leg extension and center of mass trajectories match the model predictions closely. The experimentally measured axial spring compression is slightly smaller than the predicted value. This is compensated for by the inherent elasticity of the push-pull cable system.

After adding axial compliance to the legs the robot ran 50% faster than before. It also had a considerably smoother period-1 gait and a reduction in mechanical failures.

In addition to tuning the axial compliance of the leg extension system, it is necessary to adjust to rotational compliance and damping of the passive hips. As with the earlier *iSprawl* robots, the legs are multi-material structures of hard and soft urethane. If the urethane flexures are too stiff, the legs do not flex enough and the stride length is reduced; if they are too soft the robot stumbles and loses open-loop stability [4]. Empirically, rotational stiffnesses of approximately 72 Nmm for the front legs, 54 Nmm for the middle, and

36 Nmm for the rear legs was found to give best results. In earlier *Sprawl* robots, the inherent visco-elasticity of the soft urethane provided adequate damping; for *iSprawl* it was necessary to add small friction dampers to the front and middle legs, as seen in Fig. 3.

3.2 Ground Reaction Forces

A final subject of comparison among *iSprawl*, the earlier *Sprawl* robots, and insects is the pattern of ground reaction forces (GRF). The pattern seen in insects is that the rear legs provide most of the forward propulsion at the start of each step while the front legs provide a braking force at the end of each step. The middle legs provide a mixture of propulsion and braking [6]. In addition, the front legs, being most nearly upright, have the largest vertical and smallest horizontal forces. In the upper half of Fig. 6 we see the averaged GRFs for *Sprawlita*, one of the first *Sprawl* robots with pneumatic pistons (from [2]). The pattern is similar to that seen in insects except that the rear legs drag somewhat (with a negative horizontal force) at the end of each stride. The pattern for *iSprawl* is again similar, with a couple of noticeable differences: the front legs provide less braking force and the rear legs have less drag. The reduction in parastic foot drag is partly responsible for the greater speed of *iSprawl*.

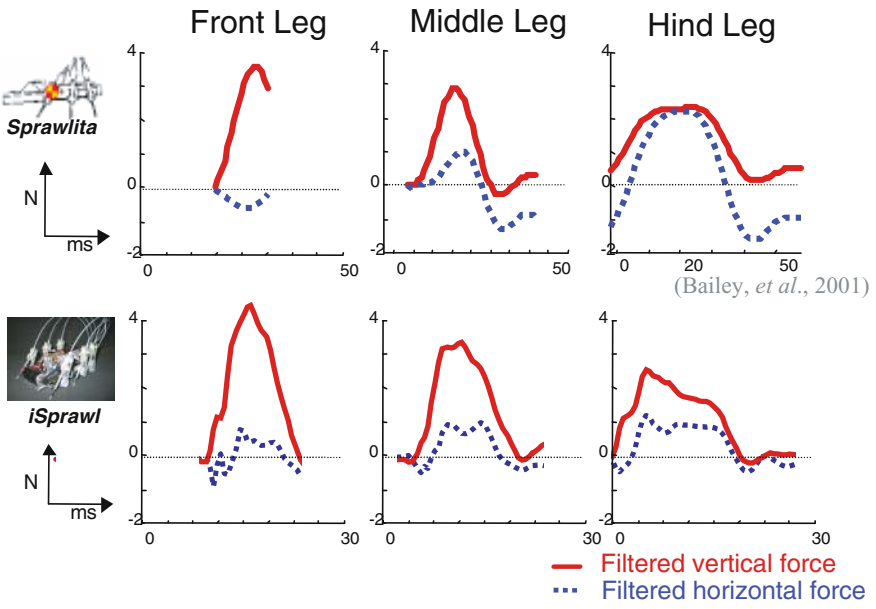


Fig. 6. The vertical and horizontal individual leg ground reaction forces for *Sprawlita* (from [2]) and for *iSprawl*

4 Discussion of Results and Conclusions

The development of a light and flexible power distribution system has allowed the creation of an autonomous, biomimetic sprawled hexapod. A comparison of the locomotion dynamics of the electrically powered *iSprawl* and the pneumatically driven *Sprawl* robots shows that despite the difference in actuation schemes, both robots demonstrate comparably fast and stable running with an open-loop actuation pattern. This suggests that the key design principles embodied in the *Sprawl* robots (namely: sprawled posture, thrusting legs, and passive hip joints with rotational compliance and damping) may have practical utility beyond this family of robots. A comparison of the leg extension profiles and ground reaction forces between the electric and pneumatic variants of the *Sprawls* shows that despite small differences, the essential motions and forces for fast and stable locomotion have been preserved. We have also found that by adjusting the passive dynamics of the robot to better match the theoretical predictions of the SLIP model, the robot is able to run more than twice as fast as its tethered cousins. A more detailed tuning of the leg impedance may yet result in faster and more stable running.

Acknowledgements

We would like to thank Trey McClung and Emily Ma for their help in conducting the experiments described in this paper. The development of the *Sprawl* robots was supported by the Office of Naval Research under grant N00014-98-1-066.

References

1. S. A. Bailey, J. G. Cham, M. R. Cutkosky, and R. Full. Biomimetic mechanisms via shape deposition manufacturing. In J. Hollerbach and D. Kodistchek, editors, *International Symposium on Robotic Research*. Springer-Verlag, London, 1999.
2. S. A. Bailey, J. G. Cham, M. R. Cutkosky, and R. J. Full. Comparing the locomotion dynamics of the cockroach and a shape deposition manufactured biomimetic hexapod. In *Experimental Robotics Vii*, volume 271, pp. 239–248, 2001.
3. J. G. Cham, J. Karpick, J. E. Clark, and M. R. Cutkosky. Stride period adaptation for a biomimetic running hexapod. In *International Symposium of Robotics Research*, Lorne Victoria, Australia, 2001.
4. J. E. Clark. *Design, Simulation, and Stability of a Hexapedal Running Robot*. PhD thesis, Stanford University, 2004.
5. J. E. Clark, J. G. Cham, S. A. Bailey, E. M. Froehlich, P. K. Nahata, R. J. Full, and M. R. Cutkosky. Biomimetic design and fabrication of a hexapedal running robot. In *Proceedings – IEEE International Conference on Robotics and Automation*, volume 4, pp. 3643–3649, 2001.

6. R. J. Full, R. Blickhan, and L. H. Ting. Leg design in hexapedal runners. *Journal of Experimental Biology*, 158(UL):369–390, 1991.
7. R. J. Full and D. E. Koditschek. Templates and anchors: Neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202(23):3325–3332, 1999.
8. Bernhard Klaassen, Ralf Linnemann, Dirk Spenneberg, and Frank Kirchner. Biomimetic walking robot scorpion: Control and modeling. In *Proceedings of the ASME Design Engineering Technical Conference*, volume 5, pp. 1105–1112, 2002.
9. T. M. Kubow and R. J. Full. The role of the mechanical system in control: a hypothesis of self-stabilization in hexapedal runners. *Philosophical Transactions of the Royal Society of London Series B-Biological Sciences*, 354(1385):849–861, 1999.
10. A. J. McClung, J. G. Cham, and M. R. Cutkosky. Dynamic maneuvering of a biologically inspired hexapedal robot. In *ASME IMECE Proceedings*, 2004.
11. K. Meijer and R. J. Full, Stabilizing properties of invertebrate skeletal muscle. *American Zoologist*, 39, 1999.
12. R. Merz, F. B. Prinz, K. Ramaswami, M. Terk, and L. E. Weiss. Shape deposition manufacturing. In *Proceedings of the Solid Freeform Fabrication Symposium*, University of Texas at Austin, 1994.
13. R. D. Quinn, G. M. Nelson, R. J. Bachmann, D. A. Kingsley, J. T. Offi, T. J. Allen, and R. E. Ritzmann. Parallel complementary strategies for implementing biological principles into mobile robots. *International Journal of Robotics Research*, 22(3/4):169–86, 2003.
14. Uluc Saranli, Martin Buehler, and Daniel E. Koditschek. Rhex: A simple and highly mobile hexapod robot. *International Journal of Robotics Research*, 20(7):616–631, 2001.

Locomotion of a Modular Worm-like Robot Using a FPGA-based Embedded MicroBlaze Soft-processor

J. Gonzalez-Gomez, E. Aguayo, and E. Boemo

Escuela Politecnica Superior, Universidad Autonoma de Madrid, Spain
{Juan.Gonzalez, Estanislao.Aguayo, Eduardo.Boemo}@uam.es

Summary. Modular reconfigurable robots offer the promise of more versatility, robustness, and low cost. They are composed of simple and small modules, capable of attach and detach one to each other. In this paper, a modular worm-like robot composed of a chain of 8 similar modules is presented. A travelling wave, that moves from the tail to the head, propels the robot forward. The positions of the articulations are calculated using the following parameters: waveform, amplitude, and wavelength. Instead of a conventional architecture, a FPGA-based soft-processor core is utilized. It includes a set of custom peripheral cores, written in VHDL. FPGAs make modular robots more versatile, adding some new featureas to the design of robots like reconfigurable control, hardware reuse, lower cost, fault-recovering, and software/hardware co-design.

1 Introduction

Modular self-reconfigurable robots offer the promise of more versatility, robustness and low cost [1]. They are composed of modules, capable of attach and detach one to each other, changing the shape of the robot. This scheme allows them to perform unusual actions like to traverse through any kind of terrain as well as climbing over obstacles or crawling inside tubes. Utilities outside the research world has not been seen yet, but they are planned to be used in space applications [3] and urban search and rescue [2].

A modular robot with N different types of modules is called N-modular. Heterogeneity is tend to be reduced, decreasing the ratio between N and the total number of modules. In the last years, the number of robot following this approach has growth substantially [5–8].

One of the most advanced systems is Polybot [1, 4], a 2-modular reconfigurable robot. Different reconfigurations and gaits has been probed; for example, from a loop, that uses a rolling gait, to a snake, with a sinusoidal gait, and finally to an spider. Currently, the third generation of modules (G3) is being

developed [9]. Each module has its own embebed PowerPC 555 processor with a traditional processor architecture.

An addtional step on moduratity is the use of FPGA technology instead of a conventional microprocessor chip. It gives the designer the possibility of implementing new architectures, faster control algorithms, or dinamically modify the hardware to adapt it to a new situation. In summary, Modular Reconfigurable Robot controlled by a FPGA not only are able to change their shapes, but also their hardware, so that, complete versatility can be achieved.

In this paper, a modular worm-like robot (Fig. 1), named Cube is presented. This is the simplest kind of modular robot, composed of 8 equal linked modules (1-modular robot). The locomotion is achieved by the propagation of waves that travel through the robot, from tail to the head. The entirely locomotion controller is embedded into an FPGA.

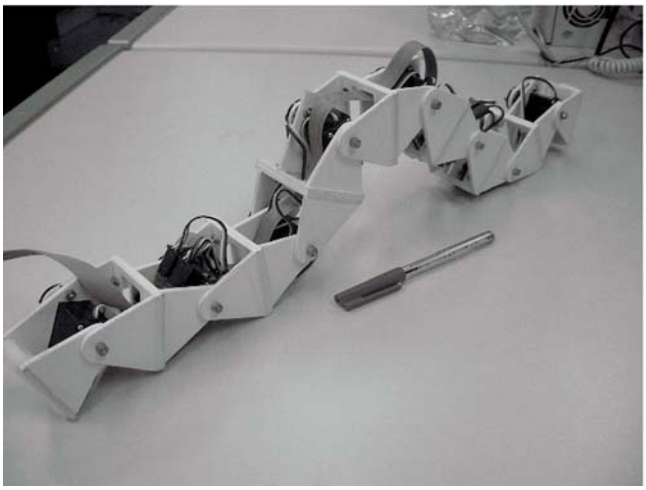


Fig. 1. The worm-like modular robot Cube. It is composed of 8 similar linked modules, connected in phase

In this first prototype, the problem of worms locomotion and FPGA-based control has been solved. The control system is centralized; an unique custom FPGA processor can control the 8 modules.

The MicroBlaze soft-processor [10] has been selected as core processor. Additional hardware units has been implemented as VHDL modules. MicroBlaze is a powerful 32-bit processor, that offers new capabilities not available on conventional processors, like the addition of custom peripheral, duplicated modules to increase reliability, or the use of dynamic reconfiguration to adapt the control to a new enviroment. This soft-processor can also run operating systems like uC/OS-II [11], a real time OS, or uCLinux [12].

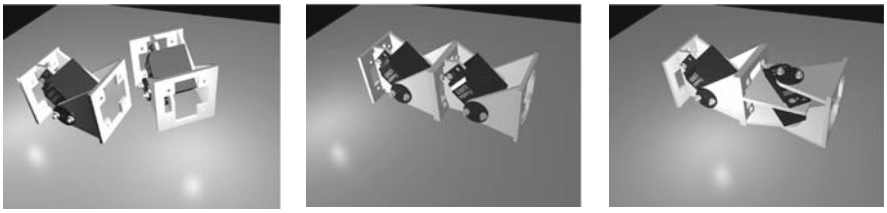


Fig. 2. CAD rendering of two Y1 modules. On the *left*, Two isolate modules in the middle, connection in phase. On the *right*, they are connected out of phase

The organization of the paper is as follows. Firstly, the mechanics and the modules is presented. Secondly, the robot locomotion, algorithms, and the locomotion controller is addressed. Finally, the implementation on FPGA is explained and the results are presented.

2 Mechanical Description

The current version of the prototype is a chain of 8 similar linked modules called Y1. In Fig. 2, a CAD rendering is showed. They have just one degree of freedom, actuated by a Futaba 3003 RC servo. The design is based on generation G1 Polybot modules [4]. In this first version, no sensors are included. Our main interest was focused on the study of locomotion, and its implementation on FPGA.

Y1 modules are simple and cheap: it is very easy to build prototypes of worm-like robots with them. These modules can be connected in two different ways, as shown in Fig. 2. One way is the connection in phase, in which two adjacent modules have the same orientation. Robots constructed using this link have all the articulations in the same plane, perpendicular to the ground (Fig. 3). Cube comprises 8 Y1 modules, connected in phase, so that it can only move along a line, forward or backwards.

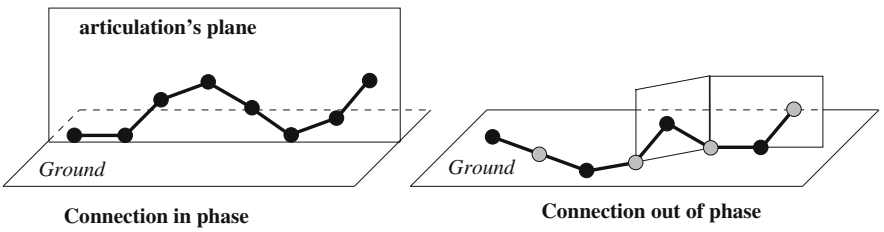


Fig. 3. Two schemes of worm-like robots. On the *left*, all the articulations are connected in phase so that they all are on the same plane. On the *right*, the connection is out of phase. With this configuration, the modules can be on different planes

The other way of connecting the modules is out of phase. Two adjacent modules are rotated 90 degrees one to each other, obtaining two degrees of freedom. One articulation moves on the ground plane (yaw) and the other does perpendicularly (pitch). The right image of Fig. 3 shows a worm with this kind of links. Black circles represent articulations that moves on the ground plane and grey circles represents articulations that moves perpendicular. This kind of robot can turn and move on different directions, not just in straight line.

The dimensions of each module, in its initial position (0 degrees angle), are $52 \times 52 \times 72$ mm, and the weight is 50 gr. They are made out of PVC. The rotations range is between -90 and 90 degrees. The robot is 576 mm in length and 400 gr in weight. The electronic and power supply are located off-board.

The consumption depends on the gaits, but typically it is 200 mA per servo, giving a total of 1.6 A. All the locomotion experiment at this first stage are realized using an off-board power supply.

3 Locomotion

Locomotion is achieved by the propagation of waves that traverse the worm, from the tail to the head. For programming simplicity, gait control tables are used [1], described in more detail in Sect. 3.1. The locomotion controller (Sect. 3.3) generates these tables automatically. The position controller reads them, producing the PWM signals to actuate the servos, and thus propelling the robot.

3.1 Gait Control Tables

Each articulation is characterized by the angle between the two segments it links. The shape of the worm, at a given instant t, is determined by the angular position vector $\overrightarrow{\varphi(t)} = (\varphi_1, \varphi_2, \ldots, \varphi_n)$. Figure 4 shows a six articulations worm-like robot and the angular position vector at a given time.

For every instant, an angular position vector there exist, determining the shape of the worm: $\overrightarrow{\varphi(t_0)}, \overrightarrow{\varphi(t_1)}, \ldots, \overrightarrow{\varphi(t_m)}$. The control table is a matrix,

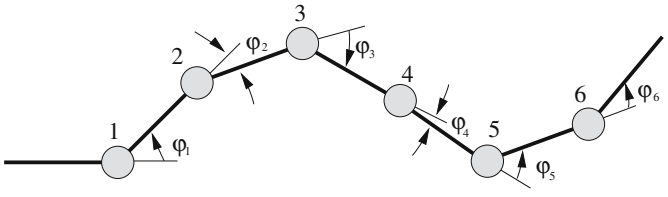


Fig. 4. Angular position vector for a worm-like robot composed of 6 articulations: $\overrightarrow{v} = (\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6)$

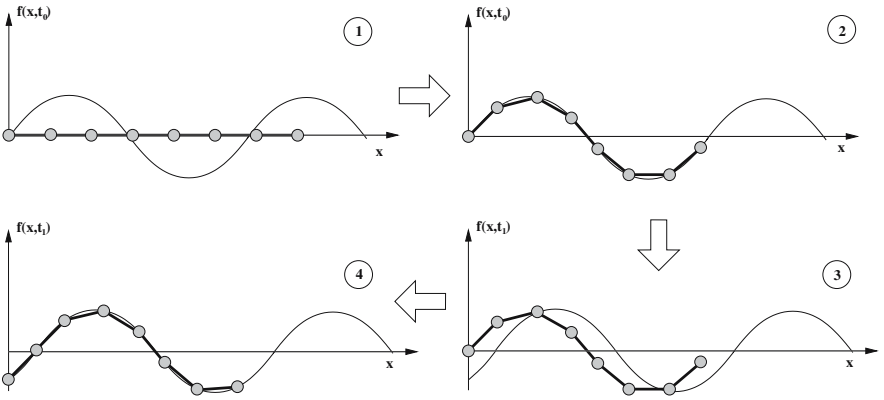


Fig. 5. The algorithm used to generate the control tables

which rows contains the angular position vectors for every instant. In order to generate the movement, the controller has to read the table, row by row, positioning the servos.

In robots like Polybot, this tables are pre-calculated and downloaded into the modules. Each table represents one gait. It is not possible to calculate or store all the possible tables for all the different gaits. Those tables are generated automatically in *Cube*.

3.2 Automatic Generation of Gait Control Tables

Control tables are generated using a wave propagation model. The algorithm is as follows (Fig. 5). Having a waveform in its initial state, $f(x, t_0)$ (in the figure, sinusoidal waves are drawn, but other waveforms could be used) and a worm with all its articulations over the x axis (Fig. 1). Let (x_i, y_i) be the coordinates of the articulation i, at some instant t. The angular position vector for the initial time, $\overrightarrow{\varphi(t_o)}$, is calculated fitting the articulations to the wave, so that $y_i = f(x_i, t_0)$ for all i. The distance L between articulations is maintained. It could be said that "the worm fits the wave" (Fig. 2). Next, the wave is shifted (instant t_1. Figure 3) and the worm fits the wave again, obtaining $\overrightarrow{\varphi(t_1)}$ (Fig. 4). Points 3 and 4 are repeated until the wave reach its initial phase. After m instant of time, all the vector that comprises the table are generated.

By means of this algorithm, control tables are obtained, regardless of the waveform used, $f(x, t)$. In the locomotion test, sinusoidal and semi-sinusoidal waves (just the positive part of the sinusoidal wave) have been used.

3.3 Locomotion Controller

The locomotion controller generates the PWM signals for positioning the servos from the wave parameters: waveform, amplitude, and wavelength.

Higher level systems could move the robot just specifying this parameters. Furthermore, at this stage, the movement of the robot is independent of the number of articulations. The planificator algorithm will determine the best wave and its parameters based on the terrain characteristics. For example, if the robot had to pass through a tube, an amplitude smaller than the section of the tube will be needed. If the obstacle is an step, a bigger amplitude will be used.

The architecture is shown in Fig. 6. The controller is composed of three subsystems. *Control table* is the central part, where the angular position vectors are stored. The contents of this table determines the movement (Sect. 3.1). The *position controller* generates the PWM signals that are applied to the servos to set their angular position.

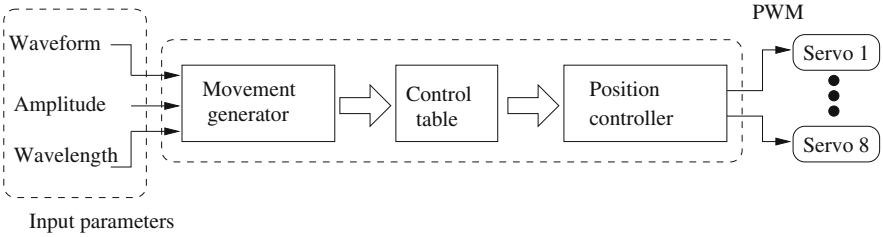


Fig. 6. Architecture of the locomotion controller

Finally, the *movement generator* obtain the gait control table from the parameter of the wave (waveform, amplitude and wavelength). It is implemented by software, using the algorithm described in Sect. 3.2.

4 Implementation on FPGA

Mainly two different approaches can be used for the implementation of the locomotion controller:

(i) Using a conventional microprocessor system, either centralized (a CPU that controls all the modules) or distributed (every module has its own embedded CPU, connected by a network). All the functionality is implemented in software. In order to add a hardware controller, a new printed circuit board design would be needed.
(ii) Using an FPGA system. Different hardware/software architectures can be designed and tested. Some subsystems could be implemented by hardware, while others by software.

We have focused on the second approach: a centralized FPGA systems. All the locomotion controller is embebed on the FPGA. The movement generator, as

well as the control tables, are implemented by software. We have used the soft-processor Microblaze. Algorithms are coded in C language, first tested on a Linux PC and then ported to Microblaze, using the GCC Cross compiler [13], supplied by the FPGA manufacturer.

The position controller is a hardware unit, written in VHDL, that acts as a peripheral for the MicroBlaze. Software can access to this unit through ports, mapped on the main memory. The positions for the 8 servos are stored in the corresponding ports, where the position controller read them and generates the PWM signal. Then main advantage of this hardware devices is its scalability. In order to control more servos, new controllers can be mapped, without physical redesign of the board, always limited to the resources available on the FPGA: the area and pins available.

4.1 The Microblaze Soft-Processor

The MicroBlaze is soft-core 32 bit Harvard-style processor described in HDL (Hardware Description Language). It was released by Xilinx recently [10]. Figure 7 shows the design loaded in the FPGA. The buses of the processor follow the Core Connect standard from IBM [14]. Also, a debug module has been included in order to be able to perform an intrusive debug of the processor using the GNU tool gdb [13].

C language can be used to design both the controller and the position computing algorithms. As the whole system (memory, buses, peripherals and processor) is being described in HDL, the hardware architecture is much simpler than traditional processor board architectures. A modification of the

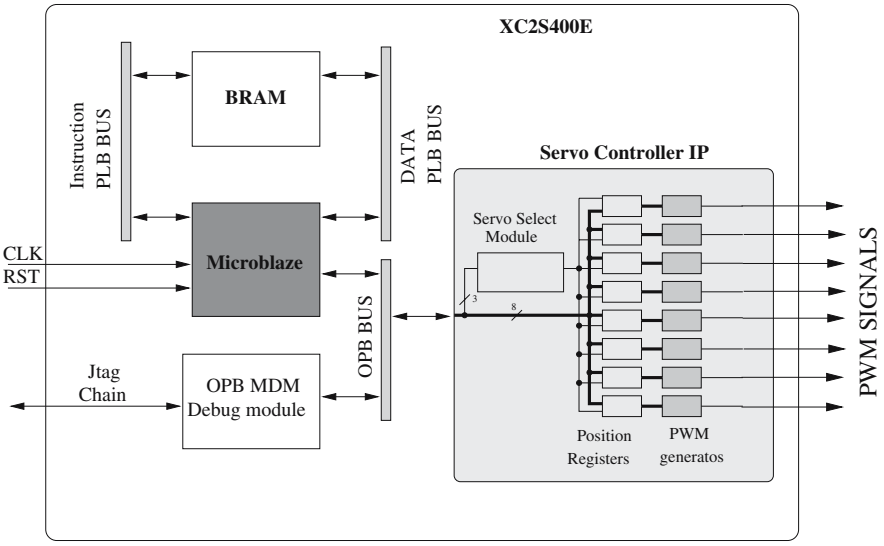


Fig. 7. Locomotion controller scheme loaded in the FPGA

controller system, only needs the loading a new design into the FPGA. No PCB modification is necesary. Thus, testing and debugging stages of the design are a much simpler task. This is a fundamental feature, since robotics is a field where testing is not only a simulating task, when a modification to the hardware system is made. The Cube prototype can loads the new system in microseconds.

Traditional robotic systems have separated hardware and software design stages, the hardware system is constructed once and then the software is loaded as many times as needed to make it work. The use of an FPGA in Cube gives this robot the facility to have many hardware and software design stages so achieving desired results. It adds more flexibility in the design stages.

Finally, as the Microblaze is being designed to use very little space in the FPGA (near a 10% of a SpartanIIE400 chip is used for the Cube controlling system), all the space left can be used to implement extra hardware.

4.2 Implementation Results

The implementation of the controlling system has been developed using the latest released Xilinx software for HDL synthesis, mapping and implementation, ISE 6.1. And the processor system developer tool, also from Xilinx EDK 6.1. The FPGA used in Cube is a SpartanIIE 400, a low cost FPGA that maintains the objective of a low cost robot. The obtained results for the final place and route of the hardware system are shown in Table 1.

Table 1. Implementations results using an SpartanIIE 400 FPGA

	Total	Used	Available
BRAMs	14	8	6 (43%)
Slices (Area)	2352	1312	1040 (44%)
I/O pins	146	10	136 (93%)
System Clock frequency (MHZ)	–	50	–

The 8 BRAM are configured to build a 32 bit words memory, having each BRAM a 4K×4 bit capacity sharing the address bus. The results obtained for the controller leave a 44% of space and 93% of the pins free in the FPGA. So that, the system still has a remarkable amount of resources available for future improvements. The board uses a 50 MHz clock generator, even considering that no optimization of the design has been carried out.

The average robot power comsuption depends on the movement performed and will be analized in detail in future work. A Typical value is 8W (1.6A, 5v).

5 Conclusions and Future Work

A modular worm-like robot has been constructed, capable of moving in a straight line, using a wave propagation gait. Locomotion controller is based on control tables, automatically generated from the parameters of the waves applied: waveform, amplitude and wavelength. Locomotion is achieved by means of the propagation of these waves along the worm, from the tail to the head. Higher level software just need to specify this parameters to locomote the robot.

The controller has been implemented on a low cost FPGA using custom cores, described in VHDL, together with the MicroBlaze soft-processor, where the algorithms are executed. FPGAs increases the robot versatility so that the designer can select the architecture that better fix the requirements. Main limitations of this approach are the memory and FPGA resource availability. The main advandages are: possibility of implementing new architectures, faster control algorithms, dinamyc hardware modification, hardware/software codesign, and remote hardware reconfiguration.

A working platform has been developed. Current research is focused on worm locomotion, studying its characteristics as a function of the wave parameters, getting insights of its relation with velocity, stability, and consumption. One approach will be the use of genetic algorithms to find the optimal parameters of the wave, given an stability, velocity, and power consumption restriction. We also are planning to study locomotion in a plane, not restricted only to straight lines. Finally, a new generation of modules, with embeded FPGAs are being constructed.

Acknowledgements

This research is supported by Project Number 07T/0052/2003-3 of the *Consejeria de Educacion de la Comunidad Autonoma de Madrid*, Spain.

References

1. Mark Yim, Ying Zhang & David Duff, Xerox Palo Alto Research Center (PARC), "Modular Robots". IEEE Spectrum Magazine. Febrero 2002.
2. M. Yim, D. Duff, K. Roufas, "Modular Reconfigurable Robots, An Aproach to Urban Search and Rescue," Proc. of 1st Intl. Workshop on Human-friendly welfare Robotic Systems (HWRS2000) Taejon, Korea, pp. 69-76, Jan. 2000.
3. M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, "Modular Reconfigurable Robots in Space Applications", Autonomous Robot Journal, special issue for Robots in Space, Springer Verlag, 2003.
4. D. Duff, M. Yim, K. Roufas,"Evolution of PolyBot: A Modular Reconfigurable Robot", Proc. of the Harmonic Drive Intl. Symposium, Nagano, Japan, Nov. 2001, and Proc. of COE/Super-Mechano-Systems Workshop, Tokyo, Japan, Nov. 2001.

5. Mark Yim, David G. Duff, Kimon D. Roufas, "Polybot: a Modular Reconfigurable Robot", IEEE Intl. Conf. on Robotics and Automation (ICRA), San Francisco, CA, April 2000.
6. P. Will, A. Castano, W.-M. Shen, "Robot modularity for self-reconfiguration," SPIE Intl. Symposium on Intelligent Sys. and Advanced Manufacturing, Proceeding Vol. 3839, pp. 236–245, Sept. 1999.
7. K. Kotay, D. Rus, M. Vona, C. McGray, "The Self-reconfiguring Robotic-Molecule," Proc. of the IEEE International Conf. on Robotics and Automation, pp. 424–431, May 1998.
8. S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, S. Kokaji, "A 3D self-Reconfigurable Structure," Proc. of the IEEE International Conf. on Robotics and Automation, pp. 432–439, May 1998.
9. M. Yim, Y. Zhang, K. Roufas, D. Duff, C. Eldershaw, "Connecting and disconnecting for chain self-reconfiguration with PolyBot", IEEE/ASME Transactions on mechatronics, special issue on Information Technology in Mechatronics, 2003.
10. Xilinx inc, "Microblaze processor Reference Guide". San Jose, California. Julio 2003.
11. Jean J. Labrosse, "Use an RTOS on your Next MicroBlaze-Based Product". Xcell journal. Issue 48. Spring 2004.
12. Microblaze uClinux Project Home Page. [on-line] http://www.itee.uq.edu.au/∼jwilliams/mblaze-uclinux/.
13. GNU project. [on-line] http://www.gnu.org.
14. IBM inc, "On-Chip Peripheral Bus, architecture specifications". Research Triangle Park, North Carolina. April 2001.

Evolutionary Synthesis of Structure and Control for Locomotion Systems

O. Chocron[1], N. Brener[2], Ph. Bidaud[2], and F. B. Amar[2]

[1] Laboratoire d'Ingénierie Informatique, ENIB/CERV, Brest
[2] Laboratoire de Robotique de Paris, Université Paris 6/CNRS 18

Abstract. This paper presents a global design approach for self-reconfigurable locomotion systems based on evaluation by dynamic simulation and optimization by artificial evolution. The main objective of this approach is to obtain fully integrated robotic solutions in term of morphology (topology and kinematics) and control (architecture and command). To achieve this goal, a modular robotic system is designed, including modules design and topology representation. Both topology and control are to be co-evolved through an evolutionary algorithm that accounts for the technological constraints (with precise module design) and multi-objective robotic missions (with detailed realistic simulation). Several design examples of modular systems are studied (rover and snake) and the autonomous reconfiguration ability is shown through simulation from rover to snake. The snake is simulated using the ODE tool showing how the robot actually walks (or crawls). The robot is feedback controlled in velocity to perform a snake-like crawling. Results of behavior and desired and simulated velocities validate the design method for locomotion systems.

1 Introduction

Many efforts have been made since a decade to obtain self-reconfigurable robotic systems (SRS) that are able to adapt their morphology to the task being undertaken. This need in adaptation has been empathized particularly in locomotion systems because many obstacles and uncertainties may be encountered while walking on rough terrains [1]. The reconfiguration possibilities can obviously be highly improved with modular robotic systems (MRS) that can change their topology by simply connect or disconnect a large number of modules that are identical one to another. The challenges of such systems are fundamental as well as technological [2]. The advantages of using MRS in association with the modular approach to achieve adaptation by reconfiguration are many:

- Each module being previously designed, the properties of the system are easily obtained from extrapolation of the modules ones.

- A deficient module can be straightforwardly replaced by another available module. This brings fault tolerant capacities for out of reach environments.
- The number of different assemblies (i.e. topologies) is exponential to the modules number, enabling a very basic functional design of modules.
- Distribution of critical resources (like power, actuation, sensors and CPU) through many modules brings robustness to the overall robot in operation.
- Mass production of a few different types of modules allows for a low cost design and production of a great number of modular robotic systems.

This promising approach has motivated researchers to develop modular SRS for critical robotic applications like space exploration [3], locomotion [4] and micro-robotics [5]. Some concepts have been thoroughly explored and led to interesting high potential systems [6]. The way and means the modular system has to adopt a new configuration according to the task considered is yet a whole new field in robotics that we could call: Modular Adaptation in Robotic Systems.

The adaptation is modular, not only because it applies to MRS, but also because of the task modularity itself. In locomotion task, we know that several objectives or constraints must be optimized such as speed, energy consumption, stability, safety and reliability [7]. Each objective can be seen as a part of the behavioral solution that could be solved separately. Meanwhile, the interdependency of the objectives (through their involved solutions) makes the design process a truly global optimization problem.

The use of highly mobile robots has been proposed for locomotion over natural terrains in planetary exploration or in military applications. Many research works have focused on behavior adaptation using advanced control methods [8, 9] or artificial evolution [10], but very few of them have concerned the evolution of the mechanical structure and its behavior simultaneously. What we have learned from the different design experiences of such complex systems is that a co-evolution of the structure and its behavior, demands a realistic simulation to be relevant [11, 12].

To answer these three issues (optimal module combination, adaptive behavior and relevant evaluation) we proposed an evolutionary design process in an earlier work [13]. A more advanced design of modules and a more powerful simulation tool has been proposed in a later work [14]. In this paper, we extend our previous works on evolutionary optimization for modular mobile robots to the design of integrated and controllable modular locomotion systems and present first results.

2 Global Evolutionary Design Process for Adaptation

Evolutionary (or artificial evolution) algorithms are optimization methods inspired by natural evolution principles (see Fig. 1). The candidate solutions of a given optimization problem are considered as individuals in a population

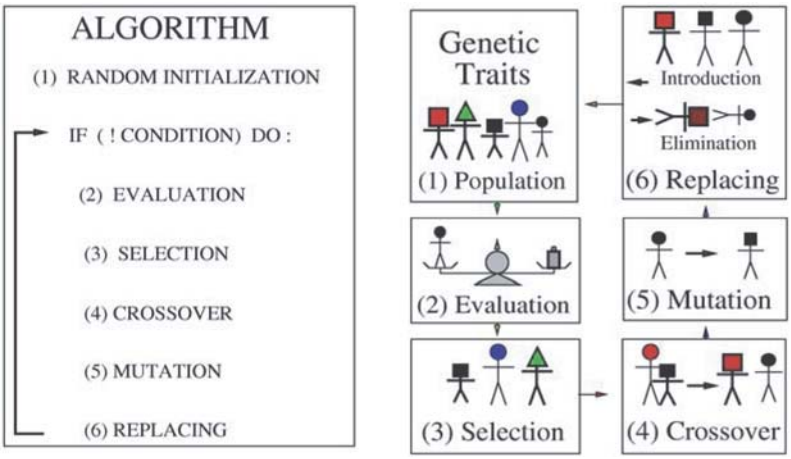


Fig. 1. Evolutionary Process over a symbolized population

that undergoes genetic operators. In that way, a population is bred under the ***survival of the fittest*** Darwinian law and will evolve toward fitter and fitter individuals.

2.1 Genotype Encoding

The use of a modular approach in robot design requires defining mechatronic modules as well as their assembly modes and their control system. Several kinds of modules and connecting ways may be considered. Actuators and sensors can also be modules since they can be combined together (motor with its gear or a photodiode with an optical lens). Some basic modules for locomotion have been proposed: **base** (payload), **segments, wheels** and **joints** including actuators (Fig. 2).

All basic modules of the same kind are strictly identical. Revolute joints are used to link two bodies (base, segments or wheels). The modular system architecture results from connections between the modules. These connections are encoded using an incidence matrix M_{ij} in which solid bodies $\boldsymbol{S}_i$ are placed

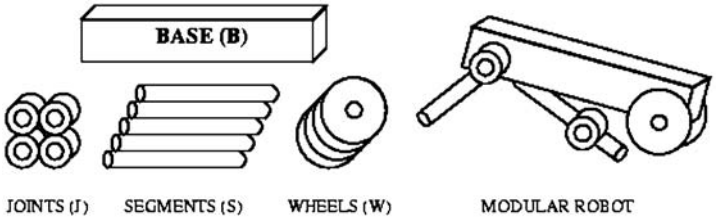


Fig. 2. Examples of basic modules for locomotion systems

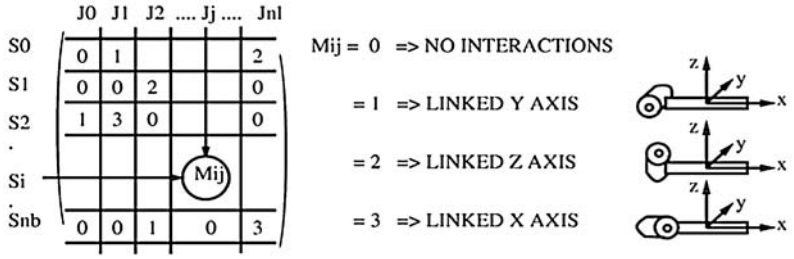


Fig. 3. Incidence matrix for mechanical system encoding

in rows and joints J_j in columns. The integer value M_{ij} at the matrix nodes determines the nature of the connection (i.e. the linkage) between a body and a joint (see Fig. 3).

The platform (base) is represented by the first row and the position of attachment points for joints are regularly and symmetrically fixed on the platform sides. The segments are attached by both extremities and the wheels at their center. Since joints and segments are bivalent, only two slots at most in the associated column or row can be filled (number different from zero), only one for the wheels and as many as the number of joints for the base.

A construction algorithm as been designed to interpret the incidence matrix. In such a way, we can describe any topological configuration for a given set of modules. The advantage of such a representation is its compactness and its generality (every modular robot can be encoded in such a way). The control laws applied on the joints have to be task-based and dependent of the robot state with regard to the task and the environment. We propose to include the control system into the genotype and to let the genetic process search for an adapted command in parallel with the topology. The control law is defined as the input voltage of the actuators (modeled as DC-motors with gears) associated to the joint (see [13] for more details).

$$U_j = U \max_j^* \cos(\omega_j^* t + \phi_j)$$

- Umax : Max. voltage on joint j
- ω_j: Pulse of signal on joint j
- ϕ_j: Phase of signal on joint j
- t: Simulated time

This control law allows only to generate periodic functions and to command the system in an open loop way without any feedback. The goal is to test the adaptation power of artificial evolution for ill conditions. Every control law parameters ($\mathrm{Umax}_j, \omega_j$ and ϕ_j) are normalized (from 0 to 1) and placed in three float vectors which constitute the behavior genotype. The resulting genotype is constituted by the integer incidence matrix (encoding the structure) and three float vectors (encoding the behavior) as shown in figure

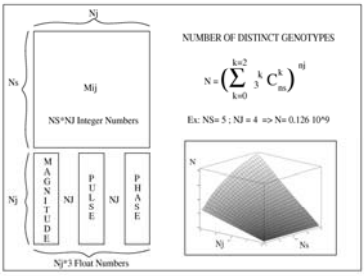


Fig. 4. Topology/Behavior genotype

(Fig. 4). The number N of possible distinct genotypes for the structure grows exponentially with the number of bodies.

2.2 Phenotype Evaluation

We have to deal with two distinct genetic entities; **Topology** and **Behavior**. Both are evolved simultaneously in the same global evolutionary algorithm (See Fig. 5). The evaluation process is done by an approximated dynamic simulation of the robot in interaction with its evolution environment (with choc and friction).

The evolutionary algorithm calls upon the simulation each time it needs to evaluate a robot for completing the specified task. The simulation being the most time consuming stage, it has to be used sparingly. We use a hierarchical evaluation (mathematical evaluation, short and full simulation) to assess the robot fitness.

2.3 Simulation Results

The population has been evolved to perform four tasks on flat terrain: maximizing speed, reaching a goal (for a fixed time), getting altitude and

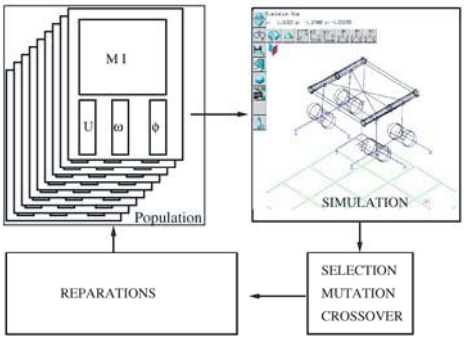


Fig. 5. Global Evolutionary Design Algorithm

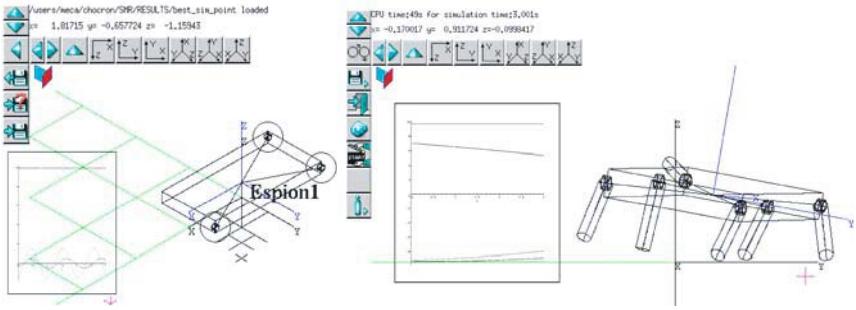


Fig. 6. Results for the reaching goal (*left*) and the getting altitude tasks (*right*)

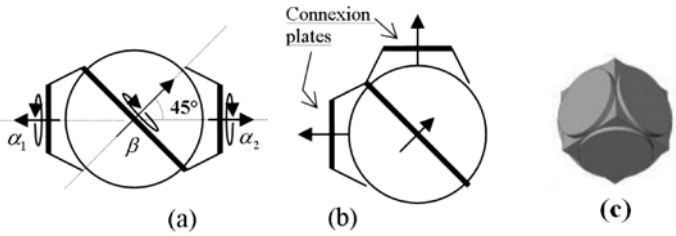


Fig. 7. Ball module configuration straight (**a**) and full (**b**) and cube module (**c**)

fast spinning. The results turned out original and worthy in spite of the open loop control architecture. Complete solutions (topology and behavior) for the tasks of reaching a goal on the ground and getting as higher as possible are presented (Fig. 6). These results prove that the evolutionary and simulation approach proposed is efficient since different types of robots (wheeled and legged) as arisen from a strictly task-based criterion.

3 Design of a Self Reconfigurable Robot

In [14] we proposed an original design of a Self Reconfigurable System, which consists of several mechatronic modules with connecting plates. By manipulating itself the system can dynamically change its topology depending on the situation. For a review on already existing or prototyped systems see [1–5, 15, 16].

This design provides wheeled locomotion when two balls are connected on opposite faces of a cube. Figure 8 shows two rover topologies. By coordinating its three rotations the ball can provide in-plan bending or can orientate the bending plane. Let α_1 and α_2 be the rotation angles of the two plates and β be the middle axis angle, with zero values in configuration (a) of Fig. 7. Let θ and ϕ be respectively the azimuth and the pitch angles of the ball, set to zero in configuration (a). The relations between inner angles (α_1, α_2 and β)

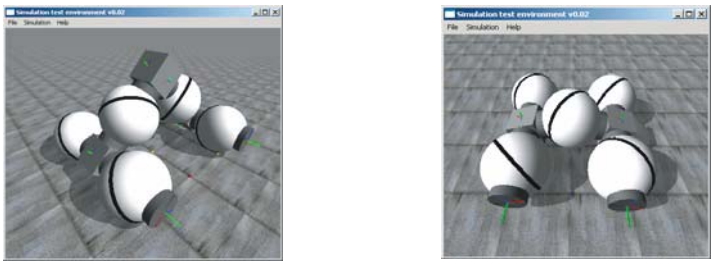


Fig. 8. Two rover topologies: independent steering (**a**) and linked steering (**b**)

and the outer angles (θ and ϕ) are given in (1). Pitch and azimuth provide easier control of module attitude since they are closer to the natural 3D space representation of motions.

$$\begin{cases} \alpha_1 = \arctan \frac{\cos(2\phi)-1}{\sqrt{2}\sin(2\phi)} + \theta + \lambda\pi \\ \beta = 2\phi \\ \alpha_2 = \arctan \frac{\cos(2\phi)-1}{\sqrt{2}\sin(2\phi)} - \theta + \lambda\pi \end{cases} \text{with} \begin{cases} \lambda = 1 & \text{if } \phi < -\pi/2 \\ \lambda = -1 & \text{if } \phi > \pi/2 \\ \lambda = 0 & \text{else} \end{cases} \quad (1)$$

3.1 Topologies and Reconfiguration

The Open Dynamics Engine library for C (ODE) has been used to study and simulate some robotic locomotion systems (see Fig. 9).

The robot can change its topology by moving, connecting and disconnecting modules. The reconfiguration stages from a rover to a snake are shown in Fig. 10.

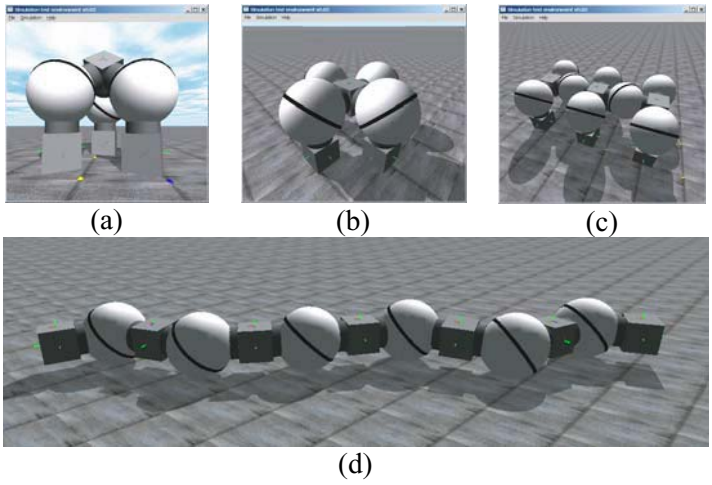


Fig. 9. Some topologies: (**a**) a tripod, (**b**) a quadruped, (**c**) a hexapod, (**d**) a snake

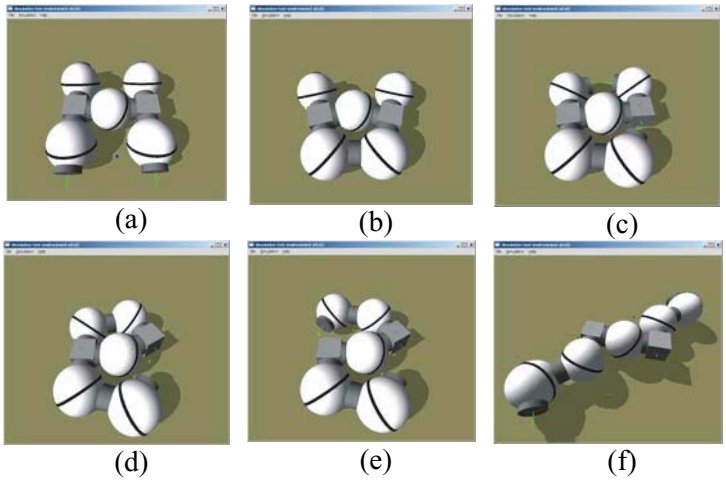


Fig. 10. Reconfiguration from rover to snake. (**a**) rover configuration, (**b**) connecting both south wheels, (**c**) disconnecting the southeast wheel from the east cube, (**d**) connecting both north wheels, (**e**) disconnecting the northwest wheel from the west cube, (**f**) expanding in snake configuration

To verify that ODE is suitable for the evaluation of performing robots, a locomotion simulation was run for the snake topology (Fig. 9d). We plotted the desired motor angular speed (empirically designed to obtain a crawling) versus the observed speed (Fig. 11). ODE computes the real speed according to the maximum actuator torque that is user defined. Here, we limited the actuation torque to 15 Nm.

- Cubes modules : mass = 1 kg, edge = 0.1 m
- Ball modules: mass = 4.5 kg radius = 0.1 m

At each time step the pitch of all 6 ball modules is computed as below. A phase lag is set between each ball module. This phase lag has been introduced so that a motion wave propagates through the snake body. The azimuth angle is set to zero.

```
   A: = 1.7; step = 0.04; phase_lag: = -3.9;
repeat
    for each i in [1..6] do
        ball(i).set_pitch(A* cos (inc + i * phase_lag)
  endFor
inc := inc+step
endRepeat
```

The inner angles are then computed using (1). The motor velocity command is computed by numerical derivation of the desired angle positions. The results are then passed to the module actuators. The ODE simulator takes

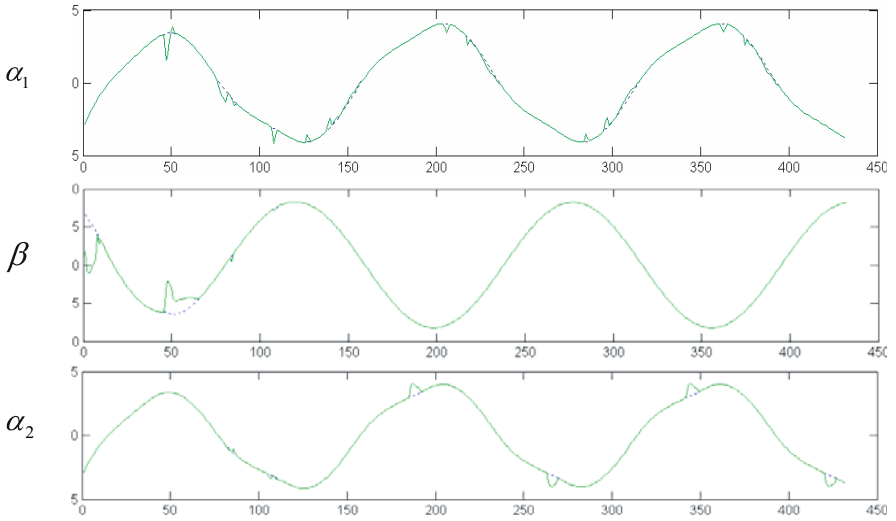


Fig. 11. Third ball module angular speeds versus time step: *Dotted line* shows the command speed and the *solid line* shows the simulated speed

into account the maximum defined torque for each joint and applies only acceptable torques. Hence, there are some discrepancies between the desired and observed joint angular speed caused by interactions of the robot modules with the ground (Fig. 11).

4 Conclusion

We have implemented an evolutionary algorithm in order to evolve modular robotic systems for locomotion. The representation with incidence matrices is generic enough to obtain any type of mobile robot (wheeled, legged, crawling or hybrid). The association of evolutionary computations and dynamic simulation has yielded some consistent results proving the reliability of the method. The new design of modules (ball joints and cube modules) integrates structure and kinematics. The combination of these modules allows for any kind of poly-articulated mechanisms (as serial or parallel ones), actuation (redundant or not) and thus, many modes of locomotion that could be exploited by the search power of artificial evolution. For future works, we consider using a graph representation for connection performances and to integrate the control architecture and the topology. We could also exploit the *step fast* mode of ODE that approximates the solving of dynamic equations with a computation complexity proportional to the module number. This can be applied to obtain a hierarchical evaluation through progressive simulation.

References

1. Ueyama T, Fukuda T, Arai F (1992) Structure Configuration using Genetic Algorithm for Cellular Robotic Systems. In: Proc. IEEE/RSJ ICIRS
2. Farritor S, Dubowsky S, Rutman N, Cole J (1996) A System-level Modular Design Approach to Field Robotics. In: Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA'96)
3. Yim M, Roufas K, Duff D, Zhang Y, Homans S (1999) Modular reconfigurable robots in space applications. Autonomous Robot Journal, robots in space, Springer Verlag, 2003. II
4. Kotay K, Rus D (1999) Locomotion versatility through self-reconfiguration. Robotics and Autonomous Systems", 26 (1999), 217–232
5. Yoshida E, Murata S, Kokaji S, Tomita K, Kurokawa H (2001) Micro selfreconfigurable modular robot using shape memory alloy. Journal of Robotics and Mechatronics, Vol.13, No.2 212
6. Murata S, et al. (2002) M-TRAN: self-reconfigurable modular robotic system. IEEE/ASME Trans. Mech. Vol. 7, No. 4, pp. 431–441
7. Celaya E, Albarral JL (2003) Implementation of a hierarchical walk controller for the LAURON III hexapod robot. CLAWAR 2003, Int. Conference on Climbing and Walking robots
8. Caballero R, Akinfiev T, Armada M (2002) Robust cascade controller for robicam biped robot: preliminary experiments. CLAWAR 2002, Madrid
9. Adouane L, Le Fort-Piat N (2004) Hybrid Behavioral Control Architecture for the Cooperation of Minimalist Mobile Robots. In: Proc. of the IEEE Int. Conf. on Robotics and Automation
10. Ziegler J, Barnholt J, Busch J, Banzhaf W (2002) Automatic evolution of control programs for a small humanoid walking robot", CLAWAR 2002
11. Sims K (1994) Evolving 3D Morphology and Behavior by Competition. Artificial Life IV, Proc. ed by R. Brooks and P. Maes, MIT Press, pp. 28–39
12. Jakobi N (1998) Running Across the Reality Gap: Octopod Locomotion Evolved in a Minimal Simulation. Lecture Notes in Computer Science 1468, Eds P. Husbands and J-A. Meyer
13. Chocron O, Bidaud P (1999) Evolutionary Algorithm for Global Design of Locomotion Systems. In: Proc. IEEE of IROS'99, Kyongju (S. Korea)
14. Brener N, Ben Amar F, Bidaud P (2004) Analysis of Self-Reconfigurable Modular Systems, A Design Proposal for Multi-Modes Locomotion. IEEE Int. Conf. on Robotics and Automation
15. Ünsal C, Khosla PK (2000) Solutions for 3-D selfreconfiguration in a modular robotic system: implementation and motion planning. Sensor Fusion and Decentralized Control in Robotic Systems III
16. Dubois M, Le Guyadec Y, Duhaut D (2003) Control of interconnected homogeneous atoms: language and simulator. CLAWAR 03, Int. Conf. on Climbing and Walking Robots

Kinematic Model and Absolute Gait Simulation of a Six-Legged Walking Robot

G. Figliolini and V. Ripa

LARM: Laboratory of Robotics and Mechatronics, DiMSAT, University of Cassino, Italy

Abstract. This paper deals with the kinematic model and absolute gait simulation of a six-legged walking robot that mimics the locomotion of the stick insect. In particular, a three-revolute (3R) kinematic chain has been chosen for each leg mechanism, as composed by the coxa, femur and tibia links.

Thus, the direct and inverse kinematic analysis are formulated for each leg mechanism in order to develop the overall kinematic model of a six-legged walking robot and thus to perform the absolute gait with respect to the ground in different operating conditions.

A significant numerical simulation of the absolute gait is shown.

1 Introduction

The coordination of six-legged walking robot is usually carried out by using a suitable stability margin between the ground projection of the center of gravity of the robot and the polygon between the supporting feet during the walking motion. Thus, a static walking can be performed.

In particular, Waldron and his co-workers at Ohio University, designed, built and tested the ASV (Adaptive-Suspension-Vehicle), as widely described in [1], while the dynamic analysis of a six-legged vehicle was proposed in [2].

A different approach was proposed and applied in the development of TUM Hexapod that mimics the locomotion system of the *Carausius morosus*, also known as *stick insect*, as reported in [3–6]. In fact, a biological design for actuators, leg mechanism, coordination and control, is much more efficient than technical solutions.

The present paper is based on this approach and deals with the kinematic model and absolute gait simulation of a six-legged walking robot that mimics the locomotion of the stick insect. The proposed algorithm has been tested through significant numerical simulations.

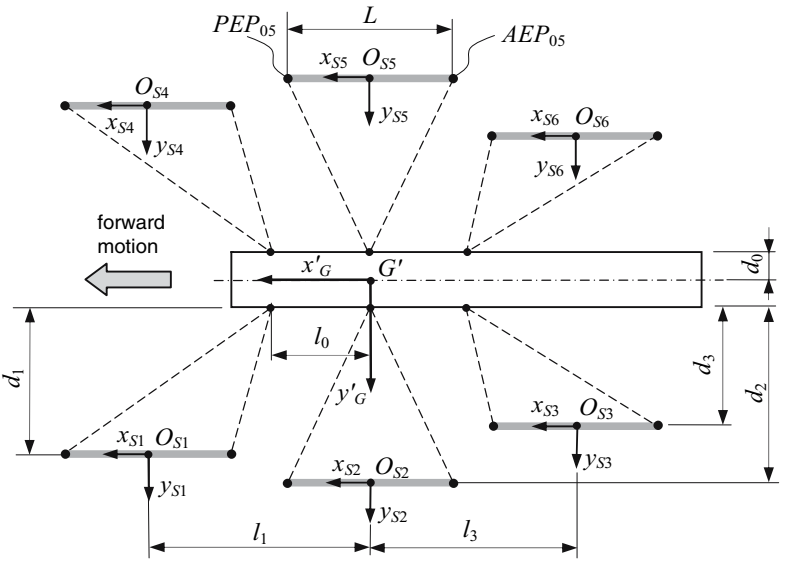


Fig. 1. Sketch and sizes of the insect: $d_1 = 18\,\text{mm}$, $d_2 = 20\,\text{mm}$, $d_3 = 15\,\text{mm}$ $l_1 = l_3 = 24\,\text{mm}$ $L = 20\,\text{mm}$, $d_0 = 5\,\text{mm}$, $l_0 = 20\,\text{mm}$

2 Leg Coordination

The gait analysis and optimization has been obtained by analyzing and implementing the algorithm proposed in [7], which was formulated by observing in depth the walking of the stick insect and it was found that the leg coordination for a six-legged walking robot can be considered as independent by the leg mechanism.

Referring to Fig. 1, a reference frame $G'(x'_G\ y'_G\ z'_G)$ having the origin G' coinciding with the projection on the ground of the mass center G of the body of the stick insect and six reference frames $O_{Si}(x_{Si}\ y_{Si}\ z_{Si})$ for $i = 1, \ldots, 6$, have been chosen in order to analyze and optimize the motion of each tip leg with the aim to ensure a suitable static stability during the walking.

Thus, in brief, the motion of each tip leg can be expressed as function of the parameters ${}^{Si}p_{ix}$ and s_i, where ${}^{Si}p_{ix}$ gives the position of the tip leg in $O_{Si}(x_{Si}\ y_{Si}\ z_{Si})$ along the x-axis for the stance phase and $s_i \in \{0; 1\}$ indicates the state of each tip leg, i.e one has: $s_i = 0$ for the swing phase and $s_i = 1$ for the stance phase, which are both performed within the range $[PEP_i, AEP_i]$, where PEP_i is the *Posterior-Extreme-Position* and AEP_i is the *Anterior-Extreme-Position* of each tip leg. In particular, L is the nominal distance between PEP_0 and AEP_0.

The trajectory of each tip leg during the swing phase is assigned by taking into account the starting and ending times of the stance phase, which are given by the algorithm proposed in [7].

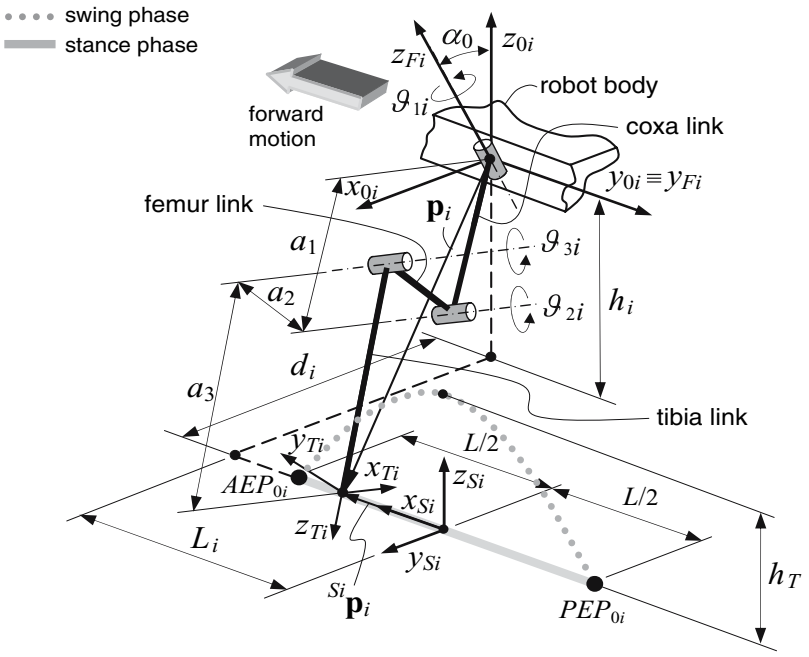


Fig. 2. A 3R leg mechanism of the six-legged walking robot

3 Leg Mechanism

A three-revolute (3R) kinematic chain has been chosen for each leg mechanism in order to mimic the leg structure of the stick insect through the coxa, femur and tibia links, as shown in Fig. 2.

A direct kinematic analysis of each leg mechanism is formulated between the moving frame O_{Ti} $(x_{Ti}\ y_{Ti}\ z_{Ti})$ of the tibia link and the frame $O_{0i}(x_{0i}\ y_{0i}\ z_{0i})$, which is considered as fix frame before to be connected to the robot body, in order to formulate the overall kinematic model of the six-legged walking robot, as sketched in Fig. 3.

In particular, the overall transformation matrix $\mathbf{M}_{Ti}^{0i}$ between the moving frame $O_{Ti}(x_{Ti}\ y_{Ti}\ z_{Ti})$ and the fix frame $O_{0i}(x_{0i}\ y_{0i}\ z_{0i})$ is given by

$$\mathbf{M}_{Ti}^{0i}(\vartheta_{1i},\ \vartheta_{2i},\ \vartheta_{3i}) = \begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^{0i}p_{ix} \\ r_{21} & r_{22} & r_{23} & {}^{0i}p_{iy} \\ r_{31} & r_{32} & r_{33} & {}^{0i}p_{iz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

This matrix is obtained as product between four transformation matrices, which relate the moving frame of the tibia link with the three typical reference frames on the revolute joints of the leg mechanism.

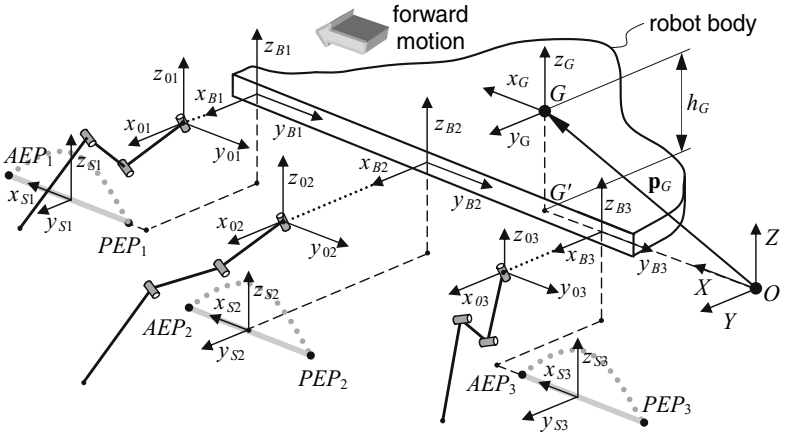


Fig. 3. Kinematic scheme of the six-legs walking robot

Thus, each entry r_{jk} of $\mathbf{M}_{Ti}^{0i}$ for $j,k=1,2,3$ and the Cartesian components of the position vector $\mathbf{p}_i$ in frame $O_{0i}(x_{0i}\, y_{0i}\, z_{0i})$ are given by

$$
\begin{aligned}
r_{11} &= \mathrm{c}\alpha_0\, \mathrm{s}\vartheta_{1i}; \quad r_{21} = -\mathrm{c}\vartheta_{1i}; \quad r_{31} = -\mathrm{s}\vartheta_{1i}\, \mathrm{s}\alpha_0 \\
r_{12} &= \mathrm{s}\vartheta_{3i}\,(\mathrm{s}\alpha_0\, \mathrm{s}\vartheta_{2i} - \mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i}) - \mathrm{c}\vartheta_{3i}(\mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{s}\vartheta_{2i} + \mathrm{s}\alpha_0\, \mathrm{c}\vartheta_{2i}) \\
r_{22} &= -\mathrm{s}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i}\, \mathrm{s}\vartheta_{3i} - \mathrm{s}\vartheta_{1i}\, \mathrm{s}\vartheta_{2i}\, \mathrm{c}\vartheta_{3i} \\
r_{32} &= \mathrm{s}\vartheta_{3i}(\mathrm{c}\alpha_0\, \mathrm{s}\vartheta_{2i} + \mathrm{s}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i}) + \mathrm{c}\vartheta_{3i}(\mathrm{s}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{s}\vartheta_{2i} + \mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{2i}) \\
r_{13} &= \mathrm{c}\vartheta_{3i}(\mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i} - \mathrm{s}\alpha_0\, \mathrm{s}\vartheta_{2i}) - \mathrm{s}\vartheta_{3i}(\mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{s}\vartheta_{2i} + \mathrm{s}\alpha_0\, \mathrm{c}\vartheta_{2i}) \\
r_{23} &= \mathrm{s}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i}\, \mathrm{c}\vartheta_{3i} - \mathrm{s}\vartheta_{1i}\, \mathrm{s}\vartheta_{2i}\, \mathrm{s}\vartheta_{3i} \\
r_{33} &= -\mathrm{c}\vartheta_{3i}(\mathrm{s}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i} + \mathrm{c}\alpha_0\, \mathrm{s}\vartheta_{2i}) + \mathrm{s}\vartheta_{3i}(\mathrm{s}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{s}\vartheta_{2i} - \mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{2i}) \\
{}^{0i}p_{ix} &= [\mathrm{c}\vartheta_{3i}(\mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i} - \mathrm{s}\alpha_0\, \mathrm{s}\vartheta_{2i}) - \mathrm{s}\vartheta_{3i}(\mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{s}\vartheta_{2i} + \mathrm{s}\alpha_0\, \mathrm{c}\vartheta_{2i})\,]\, a_3 \\
&\quad + (\mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i} - \mathrm{s}\alpha_0\, \mathrm{s}\vartheta_{2i})\, a_2 + \mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{1i}\, a_1 \\
{}^{0i}p_{iy} &= a_3\,(\mathrm{s}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i}\, \mathrm{c}\vartheta_{3i} - \mathrm{s}\vartheta_{1i}\, \mathrm{s}\vartheta_{2i}\, \mathrm{s}\vartheta_{3i}) + (\mathrm{s}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i})\, a_2 + \mathrm{s}\vartheta_{1i}\, a_1 \\
{}^{0i}p_{iz} &= [-\mathrm{c}\vartheta_{3i}(\mathrm{s}\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i} + \mathrm{c}\alpha_0\, \mathrm{s}\vartheta_{2i}) + \mathrm{s}\vartheta_{3i}(\mathrm{s}\alpha_0\, \mathrm{c}\vartheta_{1i}\, s\vartheta_{2i} - \mathrm{c}\alpha_0\, \mathrm{c}\vartheta_{2i})]\, a_3 \\
&\quad - (s\alpha_0\, \mathrm{c}\vartheta_{1i}\, \mathrm{c}\vartheta_{2i} - \mathrm{c}\alpha_0\, \mathrm{s}\vartheta_{2i})\, a_2 - s\alpha_0\, \mathrm{c}\vartheta_{1i}\, a_1
\end{aligned}
\tag{2}
$$

where $\vartheta_{1i}, \vartheta_{2i}$ and ϑ_{3i} are the variable joint angles of each leg mechanism $(i=1,\ldots,6)$, α_0 is the angle of the first joint axis with the axis z_{0i}, and a_1, a_2 and a_3 are the lengths of the coxa, femur and tibia links, respectively.

The inverse kinematic analysis of the leg mechanism is formulated through an algebraic approach. Thus, when the Cartesian components of the position vector $\mathbf{p}_i$ are given in the frame $O_{Fi}(x_{Fi}\ y_{Fi}\ z_{Fi})$, the variable joint angles $\vartheta_{1i}, \vartheta_{2i}$ and ϑ_{3i} $(i=1,\ldots,6)$ can be expressed as

$$\vartheta_{1i} = \text{atan2}\left({}^{Fi}p_{iy},\ {}^{Fi}p_{ix}\right), \tag{3}$$

$$\vartheta_{3i} = \text{atan2}\left(\text{s}\vartheta_{3i},\ \text{c}\vartheta_{3i}\right) \tag{4}$$

where

$$\begin{aligned} \text{c}\vartheta_{3i} &= \frac{({}^{Fi}p_{ix})^2 + ({}^{Fi}p_{iy})^2 + ({}^{Fi}p_{iz})^2 + a_1^2 - 2\,a_1\sqrt{({}^{Fi}p_{ix})^2 + ({}^{Fi}p_{iy})^2} - a_2^2 - a_3^2}{2\,a_2\,a_3} \\ \text{s}\vartheta_{3i} &= \pm\sqrt{1 - \text{c}^2\vartheta_{3i}} \end{aligned}, \tag{5}$$

and

$$\vartheta_{2i} = \text{atan2}\left(\text{s}\vartheta_{2i},\ \text{c}\vartheta_{2i}\right) \tag{6}$$

where

$$\begin{aligned} \text{s}\vartheta_{2i} &= -\frac{a_3\,\text{s}\vartheta_{3i}\left(\sqrt{({}^{Fi}p_{ix})^2 + ({}^{Fi}p_{iy})^2}\right) + {}^{Fi}p_{iz}\,(a_2 + a_3\,\text{c}\vartheta_{3i})}{a_2^2 + a_3^2 + 2a_2\,a_3\,\text{c}\vartheta_{3i}} \\ \text{c}\vartheta_{2i} &= -\frac{{}^{Fi}p_{iz} + \text{s}\vartheta_{2i}\,(a_2 + a_3\,\text{c}\vartheta_{3i})}{a_3\,\text{s}\vartheta_{3i}} \end{aligned} \tag{7}$$

Therefore, the [1–7] are useful for the overall kinematic model.

4 Kinematic Model of the Six-Legged Walking Robot

Referring to Figs. 2 and 3, the kinematic model of a six-legged walking robot is formulated through a direct kinematic analysis between the moving frame $O_{Ti}(x_{Ti}\ y_{Ti}\ z_{Ti})$ of the tibia link and the inertia frame $O(X\ Y\ Z)$. In general, a six-legged walking robot has 24 d.o.f.s, where 18 d.o.f.s are given by $\vartheta_{1i}, \vartheta_{2i}$ and $\vartheta_{3i} (i = 1, \ldots, 6)$ for the six 3R leg mechanisms and 6 d.o.f.s are given by the robot body, which are reduced in this case at only 1 d.o.f. that is given by X_G in order to consider the pure translation of the robot body along the X-axis. Thus, the equation of motion $X_G(t)$ of the robot body is assigned as input of the proposed algorithm, while $\vartheta_{1i}(t), \vartheta_{2i}(t)$ and $\vartheta_{3i}(t)$ for $i = 1, \ldots, 6$ are expressed through an inverse kinematic analysis of the six 3R leg mechanisms when the equation of motion of each tip leg is given by the algorithm proposed in [7] and the trajectory of each tip leg during the swing phase is assigned.

In particular, the transformation matrix $\mathbf{M}_G$ of the frame $G(x_G\, y_G\, z_G)$ on the robot body with respect to the inertia frame $O(XYZ)$ is expressed as

$$\mathbf{M}_G(X_G) = \begin{bmatrix} 1 & 0 & 0 & p_{GX} \\ 0 & 1 & 0 & p_{GY} \\ 0 & 0 & 1 & p_{GZ} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

where $p_{GX} = X_G, p_{GY} = 0$ and $p_{GZ} = h_G$.

The transformation matrix $\mathbf{M}_{Bi}^{G}$ of the frame $O_{Bi}(x_{Bi}\, y_{Bi}\, z_{Bi})$ on the robot body with respect to the frame $G(x_G y_G z_G)$ is expressed by

$$\mathbf{M}_{Bi}^{G} = \begin{cases} \begin{bmatrix} 0 & 1 & 0 & d_0 \\ -1 & 0 & 0 & l_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{for } i = 1, 2, 3 \\ \begin{bmatrix} 0 & -1 & 0 & d_0 \\ 1 & 0 & 0 & -l_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{for } i = 4, 5, 6 \end{cases} \tag{9}$$

where $l_1 = l_4 = -l_0, l_2 = l_5 = 0, l_3 = l_6 = l_0$.

Therefore, the direct kinematic function of the walking robot is given by

$$\mathbf{M}_{Ti}(X_G, \vartheta_{1i}, \vartheta_{2i}, \vartheta_{3i}) = \mathbf{M}_G(X_G)\ \ \mathbf{M}_{Bi}^{G}\ \ \mathbf{M}_{0i}^{Bi}\ \ \mathbf{M}_{Ti}^{0i}\,(\vartheta_{1i}, \vartheta_{2i}, \vartheta_{3i}) \tag{10}$$

where $\mathbf{M}_{0i}^{Bi} = \mathbf{I}$, being $\mathbf{I}$ the identity matrix.

The joint angles of the leg mechanisms are obtained through an inverse kinematic analysis. In particular, the position vector ${}^{Si}\mathbf{p}_i(t)$ of each tip leg in the frame $O_{Si}\,(x_{Si}\, y_{Si}\, z_{Si})$, as shown in Fig. 2, is expressed as

$${}^{Si}\mathbf{p}_i(t) = \left[\, {}^{Si}p_{ix}\ {}^{Si}p_{iy}\ {}^{Si}p_{iz}\ \ 1 \,\right]^T \tag{11}$$

where ${}^{Si}p_{ix}$ is reported in [7], ${}^{Si}p_{iy} = 0$ and ${}^{Si}p_{iz}$ is given by

$${}^{Si}p_{iz}(t) = \begin{cases} 0 & \text{for } s_i(t) = 1 \\ h_T\ \sin\left(\pi\ \dfrac{t - t_0^i}{t_f^i - t_0^i}\right) & \text{for } s_i(t) = 0 \end{cases} \tag{12}$$

where t is the general instant, t_0^i and t_f^i are the starting and ending times of the swing phase, respectively. Transformation matrix $\mathbf{M}_{Si}^{Bi}$ is given by

$$\mathbf{M}_{Si}^{Bi} = \begin{cases} \begin{bmatrix} 0 & -1 & 0 & L\,i \\ 1 & 0 & 0 & d_i \\ 0 & 0 & 1 & -h_i \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{for } i = 1, 2, 3 \\ \begin{bmatrix} 0 & 1 & 0 & L_{i-3} \\ -1 & 0 & 0 & d_{i-3} \\ 0 & 0 & 1 & -h_i \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{for } i = 4, 5, 6 \end{cases} \tag{13}$$

where $L_1 = l_1 - l_0, L_2 = 0$ and $L_3 = l_3 - l_0$ with L_i shown in Fig. 2.

The position of each tip leg in the frame O_{Fi} $(x_{Fi} y_{Fi} z_{Fi})$ is given by

$${}^{Fi}\mathbf{p}_i(t) = \mathbf{M}_{0i}^{Fi}\ \mathbf{M}_{Bi}^{0i}\ \mathbf{M}_{Si}^{Bi}\ {}^{Si}\mathbf{p}_i(t) \tag{14}$$

where the matrix $\mathbf{M}_{0i}^{Fi}$ can be easily obtained by knowing the angle α_0.

Thus, substituting the Cartesian components of ${}^{Fi}\mathbf{p}_i(t)$ in (3), (5) and (7), the joint angles $\vartheta_{1i}, \vartheta_{2i}$ and $\vartheta_{3i} (i = 1, \ldots, 6)$ can be obtained.

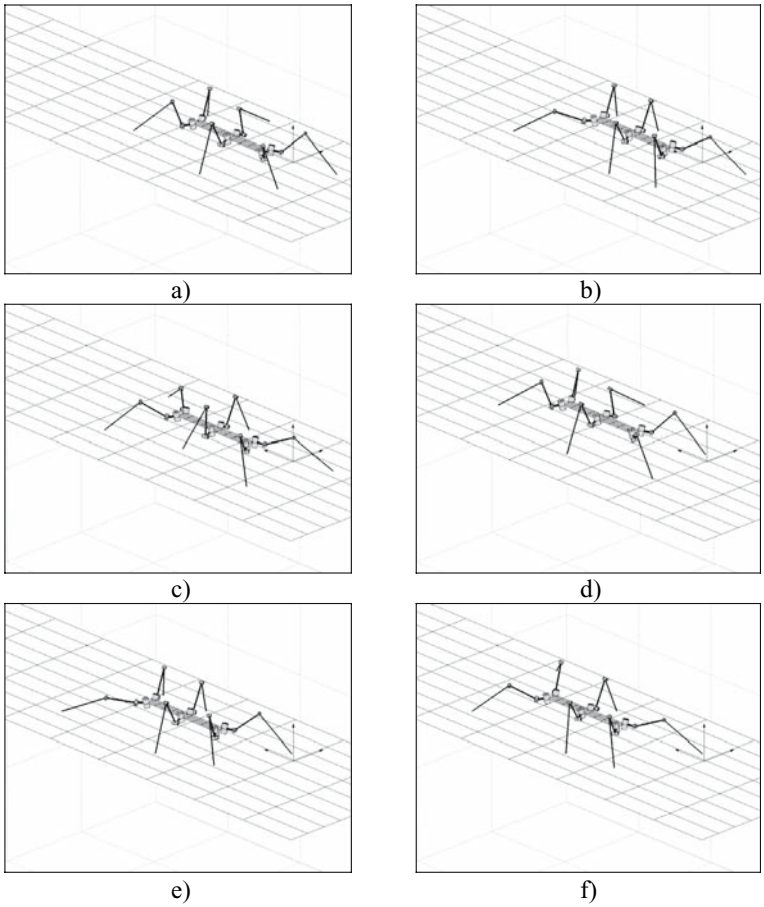


Fig. 4. Six frames of a Matlab animation that shows the absolute gait of a six-legged walking robot during its walking along the X-axis

5 Absolute Gait Simulation

This formulation has been implemented in a Matlab program in order to analyze the performances of a six-legged walking robot during the absolute gait along the X-axis of the inertia frame $O(XYZ)$.

Six frames of a significant simulation are shown in Fig. 4, where the inertia frame is shown on the right side of each configuration in light line. The six-legged walking robot is moving toward the left side by performing a transient motion at constant acceleration before to reach the steady-state condition with a constant translating velocity along the X-axis.

6 Conclusions

A kinematic model and absolute gait simulation of a six-legged walking robot that mimics the locomotion of the stick insect has been formulated.

In particular, the direct kinematic analysis between the moving frame of the tibia link and the inertia frame that is fixed to the ground has been formulated for the six 3R leg mechanisms, where the joint angles have been expressed through an inverse kinematic analysis when the trajectory of each tip leg is given.

At moment, the pure translation of the robot body along a straight line has been considered in order to validate the proposed algorithm for a stable walking of the robot.

The proposed kinematic model can be extended for analyzing the performances of the six-legged walking robot in other operating conditions, as turn right and left, avoid obstacles, climb and descend stairs.

Acknowledgements

The authors wish to thank Prof. Holk Cruse of the Department of Biological Cybernetics, Faculty of Biology, University of Bielefeld, Germany, to have kindly given some paper of the references about his research on this topic.

References

1. Song SM, Waldron KJ (1989) Machines that Walk. MIT Press Cambridge, Massachusset, London
2. Xiding Q, Yimin G, Jide Z (1995) Analysis of the dynamics of a six-legged vehicle. The International Journal of Robotics Research, 14 (1):1–8
3. Dean J (1991) A model of leg coordination in the stick insect, Carausius morosus: II. Description of the kinematic model and simulation of normal step patterns. Biological Cybernetics, 64:403–411
4. Dean J (1992) A model of leg coordination in the stick insect, Carausius morosus: IV. Comparisons of different form of coordinating mechanism. Biological Cybernetics, 66:345–355
5. Müeller-Wilm U, Dean J, Cruse H, Weidermann HJ, Eltze J, Pfeiffer F (1992) Kinematic model of a stick insect as an example of a six-legged walking system. Adaptive Behavior, 1 (2):155–169
6. Pfeiffer F, Loffler K, Gienger M, (2000) Design aspects of walking machines. 3rd International Conference on Climbing and Walking Robots, Eds. M Armada and P Gonzalez de Santos, Professional Engineering Publishing, London, pp. 17–38
7. Cymbalyuk GS, Borisyuk RM, Müeller-Wilm U, Cruse H (1998) Oscillatory network controlling six-legged locomotion. Optimization of model parameters. Neural Networks, 11:1449–1460

Part VIII

Climbing and Navigation

Simulation of Climbing Robots Using Underpressure for Adhesion

C. Hillenbrand, J. Wettach, and K. Berns

Robotics Research Lab, University of Kaiserslautern, Germany
cahillen@informatik.uni-kl.de
j_wettac@informatik.uni-kl.de
berns@informatik.uni-kl.de

1 Introduction

Applications of climbing robots using suction cups are examined in several projects worldwide. Examples are cleaning robots for windows, painting robots, inspection robots for concrete walls or climbing machine operating on steel tanks. The reliability of the used adhesion mechanisms is far away of reaching 100%. In previous research work it was shown that the concept of a wheel-driven climbing robot with a suction cup is adequate to move on vertical walls or overarm. In [3, 5] and [4] the adhesion system, the sensor system and the navigation strategy for such a robot is introduced. Still an open problem is the closed-loop control of the underpressure system because the effects of the surface conditions on the pressure in the suction cup can not be completely determined. Therefore a simulation system is developed in [9] which allows based on a thermodynamical model to test the adhesion reliability under arbitrary surface conditions.

This paper describes first the simulation based on the 1st fundamental theorem of thermodynamics. Then the results of the comparison of a simple climbing robot with one suction cup to its simulation are presented.

2 Simulation of the Underpressure System

The first part of this chapter introduces the fundamental thermodynamic equation that describes the pressure change in a control volume (i. e. underpressure chamber) because of air flow through lekages or valves from or to the outside of this volume (i.e. environment or other chamber). Furthermore the

modelling of sealing leakages due to cracks in the underground surface is described. In the second part the results are applied to a seven-working-chamber climbing robot.

2.1 Thermodynamic Model of Pressure Equalisation and Sealing Leakages

From the 1st fundamental theorem of thermodynamics and BERNOULLI's equation for the steady state flow of an ideal fluid (see [1]) the change of pressure $\dot{p}$ in a control volume is given by (1) (for a detailed derivation see [2], Chap. 3.3 and [9], Chap. 3).

$$\dot{p} = \frac{\kappa R T}{V} \sum_{k} \left(\mathrm{sgn}(p_{k_0} - p_{k_1}) * A_{k_{in}} \sqrt{2\rho \Delta p_k} \right) \tag{1}$$

The parameters in this equation are as follows:

- κ – adiabatic exponent of air
- R – gas constant of air
- T – air temperature
- ρ – air density
- V – control volume
- p_k – pressure in control volume k
- Δp_k – pressure difference between two adjacent control volumes (with pressures p_{k_0} and p_{k_1})
- $A_{k_{in}}$ – area of air flow between the two volumes

The *sgn*-function delivers the sign of its argument and the sum is calculated for all volumes that are connected with the considered one through air flows.

Figure 1 shows the developed model for sealing leakages which arise from "deformations" of the underground surface: holes, grooves and cracks in a concrete wall for example. The effective leakage area A is orthogonal to the air flow paths and connects the control volume with the outside environment. For complete exactness the hatched volume must be added to the control volume. As it is unknown, it is set approximately to 0 in the simulation.

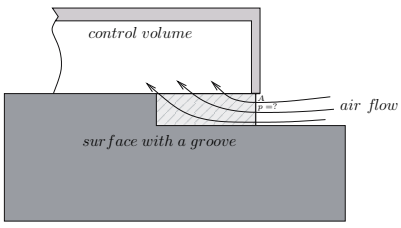


Fig. 1. Model of air flow through a leakage and corresponding flow paths. A is the effective leakage area, p is the pressure in A

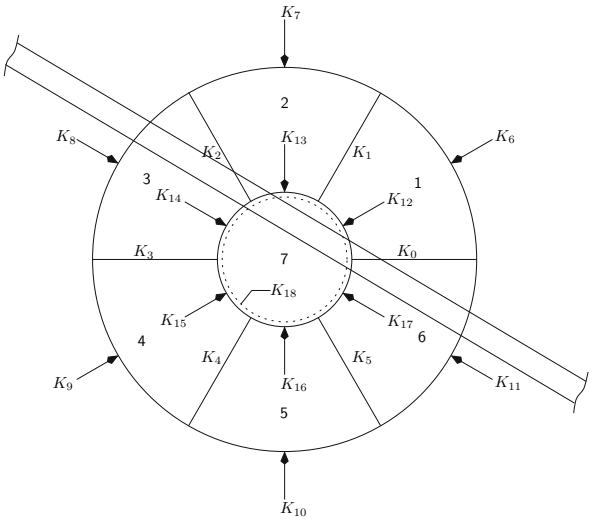


Fig. 2. Robots with numbered chambers and chamber edges K_i and an exemplary crack

Therefore the leakage area A is the product of crack depth and length of the affected sealing.

2.2 Determination of the Underpressure in One Chamber

Basis of the simulation is the model of a circular climbing robot with seven vacuum chambers for sucking to the wall and a reservoir chamber for "absorbing" fast vacuum losses in the chambers (see Fig. 2). The reservoir is evacuated constantly by a suction engine and has an exterior valve to the environment for adjusting its pressure. Each working chamber is connected to the reservoir by a valve for regulating the chamber pressure. According to (1) the pressure in each chamber depends on the air flow – driven by pressure differences – through the valve area (reservoir) and the leakage areas (adjacent chambers). For a given crack the sealing leakages have to be calculated separately for each edge $K_i, i = 0, \ldots, 17$ between two chambers or a chamber and the environment. Due to the leakage modelling introduced in Chap. 2.1 the central working chamber can never have a connection to the environment. Application of (1) to the robot model leads to (2) for the pressure change in the first chamber and to (3) for the one in the reservoir.

$$\begin{aligned}\dot{p}_1 = \tfrac{\kappa \cdot R \cdot T}{V_1} \cdot \sqrt{2\rho} \cdot \\ \Big[&\operatorname{sgn}(p_6 - p_1) \cdot A_{L,0} \cdot \sqrt{|p_0 - p_1|} \\ &+ \operatorname{sgn}(p_2 - p_1) \cdot A_{L,1} \cdot \sqrt{|p_2 - p_1|} \\ &+ \operatorname{sgn}(p_7 - p_1) \cdot A_{L,12} \cdot \sqrt{|p_7 - p_1|} \\ &+ \operatorname{sgn}(p_a - p_1) \cdot A_{L,6} \cdot \sqrt{|p_a - p_1|} \\ &- \operatorname{sgn}(p_1 - p_R) \cdot A_{V,1} \cdot \sqrt{|p_1 - p_R|} \Big] .\end{aligned} \tag{2}$$

A_{L_i} is the leakage area and p_i is the chamber pressure according to the mentioned numbering scheme, $A_{V,1}$ is the opening area of valve 1, p_R is the reservoir pressure and p_a is the ambient air pressure.

$$\begin{aligned}\dot{p}_R = \frac{\kappa \cdot R \cdot T}{V_R} \cdot \Bigg\{ \sqrt{2\rho} \cdot \Bigg[&\operatorname{sgn}(p_a - p_R) \cdot A_{V,a} \cdot \sqrt{|p_a - p_R|} \\ &+ \sum_{i=1}^{7} \left(\operatorname{sgn}(p_i - p_R) \cdot A_{V,i} \cdot \sqrt{|p_i - p_R|} \right) \Bigg] - \rho \cdot \dot{V}_g \Bigg\} .\end{aligned} \tag{3}$$

$A_{V,a}$ is the opening area of the exterior valve and $\dot{V}_g$ is the volume flow taken away by the suction engine.

3 Simulation User Interface

The simulation model has been implemented in C++ using the MCA framework (see [6] and [8]). Figure 3 shows the user interface (GUI) which is built with MCAGUI and Qt (see [7]). On the left hand the robot is shown while moving over a crack (parallel blue lines) along a given trajectory (line with crosses). The color of the chambers corresponds to their pressure and the

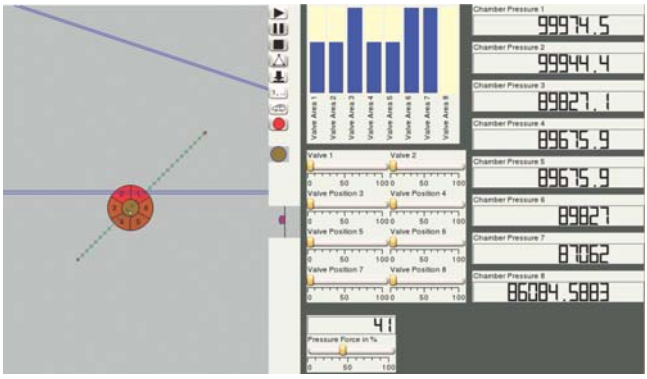


Fig. 3. Simulation GUI as screen shot

yellow circle within the robot shows the working point and value of the actual total pressure force that pushes it onto the wall. On the right there are monitors for the valve areas and chamber pressures and sliders for manual valve adjustment or determination of a desired pressure force. If this option is used an additionally developed control system computes the necessary pressure for each chamber – taking into account failed chambers due to too big leakages – and regulates correspondingly the valves by pid controllers. So the simulation allows a judgement which crack dimensions are realistic to be coped by the robot.

4 Validation of the Simulation System

For the validation of the correspondance between simulation results and reality a simple prototype has been built. With this robot measurements of the parameters of the suction engine, determination of the maximal underpressure on different surfaces and driving experiments on a concrete building wall have been performed. The resulting parameters have been used for a simulation of the prototype which allows a good comparison between model and reality.

4.1 Realized Prototype

It consists of a reservoir (volume: 10 l) and one working chamber (volume: 2.2 l) connected by an adjustable valve, a differenzial drive with a castor wheel and a stationary suction engine. Figure 4 shows a foto of the robot and the engine test rig. The test rig consists of the engine (encapsulated by blue plastic cover at the left side of the rig) and a suction channel with two pressure sensors (long grey tube with changing diameter). The robot is connected by a suction tube attached at the right end of the rig. The weight of the robot is 6.5 kg and the suction relevant area of the chamber is 0.0491 m^2. The sealing mechanism consists of an air hose of about 25 cm in diameter (wrapped with textile material to reduce sealing friction) whose pressure is surveyed by an sensor and adjustable via a manual valve. During operation of

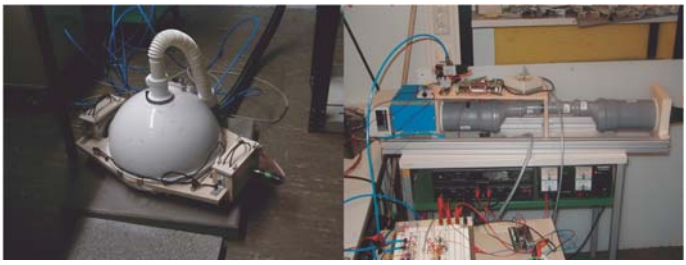


Fig. 4. Robot prototype and engine test rig

the robot the underpressure in the reservoir and working chamber is measured by two sensors and recorded by an oszilloscope.

4.2 Experiments

Determination of the Suction Engine Parameters

The maximal underpressure in the working chamber depends on the surface of the underground, the effectivity of the sealing and the suction power of the engine. The characteristic of the suction engine describes the relation between air flow through the motor and underpressure in the suction channel: when the underpressure is minimal the flow rate gets its maximum, when the suction orifice is sealed the underpressure gets maximal and the flow decreases to zero. The air flow can be measured with a pitostatic tube that is realised in the smaller middle of the grey-colored suction channel (see Fig. 4). Figure 5 shows a schematic sketch of the experimental setup.

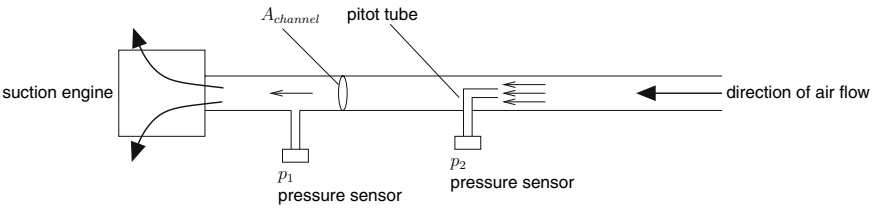


Fig. 5. Schematic sketch of the air flow measurement

According to Bernoulli's equation for the steady state flow of an ideal fluid the following holds:

$$p + \frac{\rho}{2} \cdot v^2 = p_0 = constant \tag{4}$$

where p is the static and $p_{dyn} = \frac{\rho}{2} \cdot v^2$ is the dynamic pressure of the flow (as the suction channel is horizontal the gravity pressure is neglected). The first pressure sensor measures the static pressure and the second one the total pressure p_0 by the perpendicular bent pitot tube. From these values the air flow $\dot{V}$ can be calculated as $\dot{V} = A_{channel} \cdot v_{air} = A_{channel} \cdot \sqrt{\frac{2 \cdot (p_0 - p)}{\rho}}$ (the diameter of the channel is 3.6 cm). The corresponding underpressure is the static pressure p in the tube. The characteristic of the engine is computed from measurements at different revolutions by variation of the suction orifice. Figure 6 shows the resulting curve and its linear regression line. As the sensor signals show a strong noise – they are analyzed by a prototype circuit with op-amp – the engine charateristic is taken from the regression line. This linear relation between air flow and pressure is also used in the simulation: the air flow $\dot{V}_g$ in (3) is computed from the reservoir pressure (equal to suction channel pressure) via $\dot{V}_g = 78.3 - (p_a - p_R) * \frac{78.3}{22632}$.

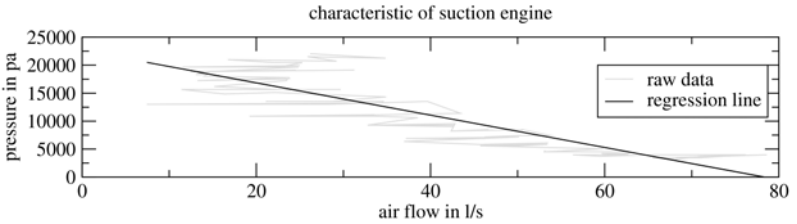


Fig. 6. Relation between air flow and underpressure of the suction engine

Measurement of the Maximal Underpressure on Different Surfaces

Figure 7 shows the recorded pressure values in the reservoir and working chambers with the prototype put on different surfaces on the floor. In each case the suction engine is operated from zero to maximal revolutions and back to zero at the following valve orifice areas: 0 cm^2, 0.375 cm^2, 0.75 cm^2, 1.125 cm^2 and 1.5 cm^2 (marked with 1.) to 5.)). With closed valve, the negative reservoir pressure is about 20000 Pa which is little bit below the maximum of the suction engine due to pressure losses in the suction tube. On surfaces where the sealing works well the chamber pressure reaches almost the reservoir pressure at completely opened valve (acrylic/wood plate, rough/smooth concrete plate) with a maximum of 17000 Pa on acrylic and a minimum of 10000 Pa on rough concrete. The corresponding pressure forces are between 490 N and 830 N – very comfortable for a robot of 6.5 kg. The grooves in the third concrete plate are 5 mm wide and deep and form a grid with 5 cm spacing. On this surface no suction forces can be achieved. A very interesting phenomena is shown in the last diagram (marked with arrows). In this situation the robot is put half on the floor and on the acrylic plate (thickness: 5mm). With at least $\frac{3}{4}$ opened valve the sealing suddenly makes a "jump" and seals up when the air flow through the leakages is big enough and its dynamic pressure sucks the sealing to the surface.

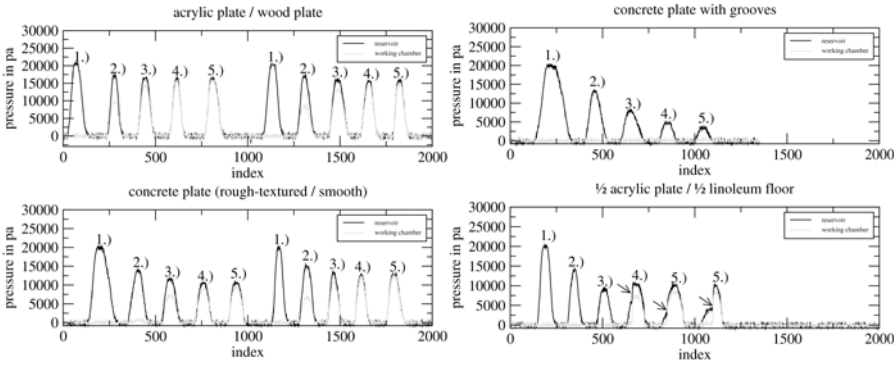


Fig. 7. Relation between air flow and underpressure of the suction engine

Driving Experiments on a Concrete Building Wall

First wall-driving experiments have been performed on the smooth concrete plate put in a vertical position. As the above results anticipated the sucking was no problem, but a movement was difficult due to the sealing friction. A wrapping with sellotape reduced the friction sufficiently and the robot moved quite well on the plate (sealing hose pressure: 17000 Pa, chamber pressure: 4000 Pa). The building wall used for further tests had a smoother surface because of its coat of paint. Therefore the robot could move very well without a taped sealing and "obstacles" of 2–3 mm height were no problem.

4.3 Comparison with Simulation Results

In order to validate the simulation system the relevant parameters of the suction engine and the robot have been used for a simulated test run where the prototyp had to cross nine cracks of different dimensions (shown in Table 1) by an orthogonal trajectory. The robot model was adopted so that only the central working chamber (7) and the reservoir were active. The latent sealing leakages were adjusted for a surface as the one of the smooth concrete plate: a leakage depth of 0.14 mm along the length of the sealing (785.4 mm) leads to a maximal negative chamber pressure of about 11800 Pa. Figure 8 shows the pressure force during the test run. The desired pressure force was set to 17.3% of its maximum (in correspondance to a chamber pressure of 4000 Pa used during wall-driving tests). Up to a $0.5 \times 1\,\mathrm{cm}^2$ crack the pressure can be kept constant while crossing it and up to a $1.3 \times 1.3\,\mathrm{cm}^2$ crack the pressure force is at least big enough to compensate the gravity force of the robot (even though the fricton of the driving wheels then will not be big enough for moving the robot). The last crack was only used because of its "standard" dimensions found on common concrete building wall: it is much to big for the prototype and will be a challenge for the seven-chamber-robot.

The test shows that the real pressure ratios can be simulated very well with the developed software system. So the thermodynamic model of the vacuum system is exact enough for generating predictions how the seven-chamber-robot will behave in reality.

Table 1. Crack dimensions used for the test run

Crack Number	Width	Depth	Crack Number	Width	Depth
1	0.5 cm	0.5 cm	6	1.2 cm	1.2 cm
2	0.5 cm	1.0 cm	7	1.3 cm	1.3 cm
3	1.0 cm	0.5 cm	8	1.4 cm	1.4 cm
4	1.0 cm	1.0 cm	9	4.0 cm	3.5 cm
5	1.1 cm	1.1 cm			

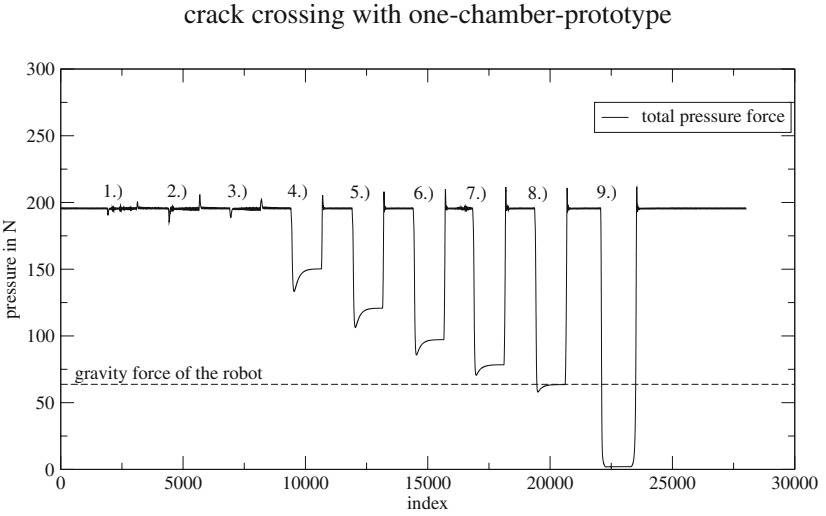


Fig. 8. Pressure force of the prototype during the crossing of nine cracks with different dimensions (refer to Table 1)

5 Summary

This paper introduced the thermodynamic model of the seven chamber vacuum system used for a new type of climbing robot. This model has been realised in a simulation software which was also outlined. The focus was on experiments performed with a simple prototype in order to get measurement results of the suction engine parameters and of the pressure ratios on different surfaces. Driving tests on a building wall have been re-simulated with the software system and the results have showed that the simulation model is realistic enough to predict which kind of surface irregularities the robot will manage.

References

1. Bernoulli's law. `http://scienceworld.wolfram.com/physics/BernoullisLaw.html`, 2004.
2. H. Hetzler. Physikalische modellierung und simulationsuntersuchungen zur regelung eines radgetriebenen, sich mit unterdruckkammern haltenden kletterroboters. Studienarbeit, FZI – Forschungszentrum an der Universität Karlsruhe (TH), März 2002.
3. C. Hillenbrand and K. Berns. Navigation and localization devices and the concept for a mobile robot. In *IEEE – International Conference on Mechatronics and Machine Vision in Practice (M2VIP) Chiang Mai /Thailand*, 2002.
4. C. Hillenbrand and K. Berns. A climbing robot based on under pressure adhesion for the inspection of concrete walls. In *35th International Symposium on Robotics (ISR), Paris / France*, p. 119, March 2004.

5. C. Hillenbrand, K. Berns, F. Weise, and J. Koehnen. Development of a climbing robot system for non-destructive testing of bridges. In *IEEE – International Conference on Mechatronics and Machine Vision in Practice (M2VIP), HongKong*, 2001.
6. Mca at sourceforge.net. `http://sourceforge.net/projects/mca2`, 2004.
7. Trolltech homepage. `http://www.trolltech.com`, 2004.
8. K.-U. Scholl, B. G. J. Albiez, and J. Zöllner. Mca – modular controller architecture. In *Robotik 2002, VDI-Bericht 1679*, 2002.
9. J. Wettach. Simulation und regelungskonzept für den unterdruckbasierten haltemechanismus eines radgetriebenen kletterroboters. Diplomarbeit, Technische Universität Kaiserslautern, 2004.

Technique for a Six-Legged Walker Climbing a High Shelf by Using a Vertical Column

Yu. F. Golubev and V.V. Korianov

Lomonosov Moscow State University, Russia

Abstract. The paper develops a subject of [1]. The walker's ability to overcome the vertical column is used for overcoming the combination of obstacles. This combination consists of two objects. One of them is the vertical column and the other is a lofty horizontal shelf with a vertical wall. A motion of a six legged insectomorphic walking robot is designed so that the walker could climb on the shelf by using the vertical column as a support. Robot can use only Coulomb's friction forces to realize its motion along obstacles. Different algorithms of climbing are considered in dependence on combinations of the column's and the shelf's altitudes. Results of 3D computer simulations of the full dynamic of the robot are presented to demonstrate the practicability of the proposed robot motion control.

1 Introduction

The ability of a walker to overcome a terrain with a conglomeration of obstacles can be formed by teaching the robot to overcome isolated obstacles as well as reasonable combinations of obstacles step by step. This approach to training is widely used in a sport like mountain climbing, competitions of firemen and so on. Some examples of overcoming a terrain with small roughness are given in [2]. Overcoming isolated obstacles by means of walker's jump is considered in [3, 4]. Methods for overcoming of vertical column by means of friction forces are presented in [1]. The attempt to design insectomorphic walker motion for surmount combination of two obstacles is made in this paper. This combination consists of a vertical column and a lofty horizontal shelf with a vertical wall. It is supposed that the height of a shelf doesn't permit to robot to climb on it if the shelf is the isolated obstacle. At the same time in combination with vertical column this obstacle can be overcame if the walker can climb the column. Motion design for climbing the column is adopted from [1]. In reality heights of the shelf and the column may differ. Since there is need to invent appropriate variants of robot's behavior in dependence on correlation of these heights. Appropriate variants are considered below. As in [1] the computer simulation of robot motion is

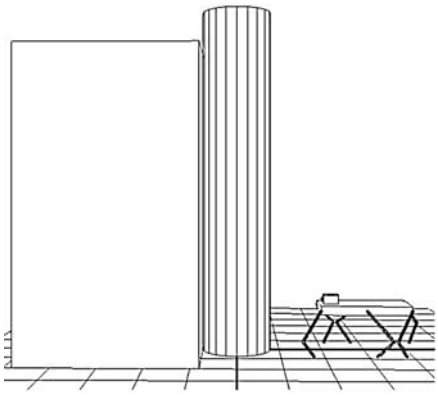


Fig. 1. Statement of the problem

accomplished to prove the robustness of algorithms developed in this paper for motion design.

2 Statement of the Problem

The vertical column is posed right against the shelf. The robot knows heights of the column and of the shelf. Initially it goes along the horizontal supporting plane in the direction perpendicular to the shelf and has an intention to arrive on the upper horizontal surface of the shelf. The shelf has the vertical wall and its height is so large that the walker couldn't climb it without some additional support (Fig. 1). The column and the walker are made like in [1]. So the walker can use algorithms for climbing the column presented in [1]. The focuses of this paper are methods for climbing from the column to the upper surface of the shelf. We suppose that the motion of the robot should be comfortable. It means that the property of static stability is fulfilled for all the time and there are no collisions of any part of the robot with a supporting surface. If the height of the column is more than the height of the shelf plus the length of the robot there are two evident variants of robot's behavior. In according with one of them the walker reaches the top of the column then it goes down across the top, reaches the upper surface of the shelf and transforms its motion for going along the shelf. This kind of motion is just like in [1]. Another variant provides for walker to go up the column until its rear edge turns out to be above the upper surface of the shelf. Then robot goes just across the column to opposite side of the column. After that the walker goes down back to front, transforming its motion for going along the shelf. At the end of this maneuver the robot turns out to be oriented so that its front will look at the column. Robot will have to do one hundred eighty degree turn to go along the shelf.

The last variant of the walker's behavior includes only one new element in comparison with algorithms of [1]. This element is the motion just across the column.

More complicated variants of walker's behavior appear when the difference between heights of the column and the shelf is not very large. Robot has to go to the top of the column in all that cases. Then robot should go either down to the shelf, or up to the shelf in accordance with static stability requirement [5]. If the difference of heights is small enough the walker may go to the shelf having the body horizontal oriented. But at large scale of these differences the body of the walker has to be inclined by appropriate manner to transfer robot to the upper surface of the shelf. The motion design for robot in these cases is confronted with difficulties because inconvenient initial position of the robot due to small enough circle top of the column. This circumstance provides the problem of neighboring legs intersection.

3 Sketch of a Robot

As in [1] we assume that the robot consist of rigid body (parallelepiped of sides a, b, c) and six identical insectomorphic legs. Each leg consists of two links (hip and shank). Denote l_1 the length of a hip and l_2 the length of a shank. Cartesian reference frame $Oxyz$ is connected with a body so that the point O is the body's center, axis Oz is oriented along the structural vertical, axis Ox is length-wise oriented, axis Oy forms the right reference frame. Coordinates of leg's attachment points to the body are as follows

$$(x_i, y_i, z_i) = (([(i-1)/2] - 1)a/2, (-1)^i b/2, 0) \; .$$

Index i defines a leg's number and symbol [ ] means the integer part of a real number. The structural vertical at a point of a leg attachment and two links of a leg belong to the same plane. This plane can rotate around the structural vertical. It is supposed that the walker can contact with the column and supporting plane by means of foots only, intersections of legs should be absent during the motion, sufficient supporting reactions exist for realization of prescribed motion.

4 Motion Design

General approach to the motion design is as follows. At each moment of time we prescribe the position of the robot's body and positions of foots standing on the support. So we have program dependences on time of supporting leg's angles if position of a body depends on time by prescribed manner. These

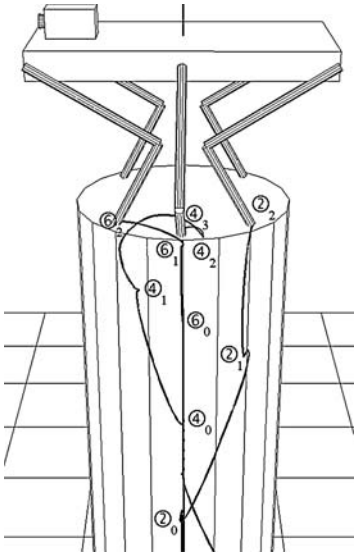


Fig. 2. Legs' transfer

program dependences of leg's angles should be realized by leg's angles servo-drives with PID control. All leg's transfers over the supporting surface are realized with according to the formula

$$\mathbf{r} = (1 - \lambda)\mathbf{r}_1 + \lambda\mathbf{r}_2 + \hat{y}\rho_n \ ,$$

where $\mathbf{r}_1$, $\mathbf{r}_2$ are radius-vectors of foot at initial and at final moments of transfer stage, $\boldsymbol{\rho} = \mathbf{r}_2 - \mathbf{r}_1$, $\boldsymbol{\rho}_n = \boldsymbol{\rho} \times (\mathbf{e} \times \boldsymbol{\rho})$, $\mathbf{e}$ is a vector perpendicular to a plane of a transfer trajectory, $\hat{y} = k(y' + y_s)$ is an ordinate of a point moving along the modified step cycle [1], k gives a convexity of transfer trajectory. Coefficient λ is a function of time which synchronizes body's and legs' motions.

The Fig. 2 illustrates the consecution of legs' transfers when robot climbs to the top of the column. Marks on the figure are legs' numbers. Their indexes correspond to the order of legs' transfers (legs are transferred successively 6_0 to $6_1, 4_0$ to $4_1, 2_0$ to $2_1, 6_1$ to $6_2, 4_1$ to $4_2, 2_1$ to $2_2, 4_2$ to 4_3). Supporting points $2_0, 4_0, 6_0$ appear due to the regular gallop gait and correspond to the initial position of the robot before the maneuver of climbing to the top. Legs with numbers 5, 3, 1 are transferred symmetrically on the invisible side of the column.

The motion design for robot going along the horizontal supporting plane, climbing from horizontal plain to the column, going up and down the column, climbing from the high column to the horizontal supporting plane are described in details in [1]. In this paper for the most part we will consider cases, when heights of the shelf and of the column differ not very much. For all these

cases robot should go to the top of the column firstly and only after that it can go to the shelf. The main difficult obstacle for these cases is the narrow top of the column having the form of a circle. So there is danger to have intersections of legs during the maneuver of going to the shelf. From here on we suppose that the initial position of the robot looks like position represented in Fig. 2 and that the robot try to go to the shelf by the gallop gait. For motion design we will use the criterion of static stability [5] which means that the vertical projection of the robot's center of mass to the supporting plane should belong to the inner part of the supporting polygon. Robot's center of masses does not coincide with point O due to non-zero legs' masses. According to this criterion robot can't lift up front legs as well as rear legs at initial position. It can't move middle legs due to danger of legs' intersection afterwards. So the climbing maneuver should begin with the body motion. We adopt the robot's and column's geometric relations as follows

$$a : b : l_1 : l_2 : r = 1 : 0.5 : 0.5 : 0.33 : 0.4 \, ,$$

where r is the radius of the column. Let us begin with the simplest case. Denote δh the difference between the heights of the shelf and the column.

1. The height of the shelf is equal to the height of the column ($\delta h = 0$). Robot may go to the shelf keeping its body horizontal. We represent the robot's motion as a succession of next stages.

1.a The robot's body moves diagonally in backward and upward directions. All foots are staying at its supporting points. This stage prepares the future transfer of front legs. The body backward motion creates the stability margin enough to transfer front legs. The body upward motion provides the possibility to move front legs without its intersection with middle legs.

1.b Foots of front legs are transferred to the edge of the shelf. Middle and rear legs' foots are staying at its supporting points. Robot's body is immovable.

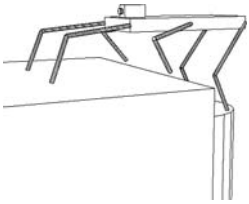
1.c Front and rear legs are at a supporting stage. Foots of middle legs are transferred to initial supporting points of front legs on the top of the column. Robot's body motion toward to the shelf begins. Robot's center of mass projection belongs to the polygon formed by middle and rear legs.

1.d Foots of front legs are transferred to next supporting points as far as possible from the edge of the shelf. Projection of the center of masses belongs to the supporting polygon of rear and middle legs.

1.e Body's center of mass projection goes to the supporting polygon of front and middle legs. Then foots of rear legs are transferred to initial supporting points of middle legs on the top of the column.

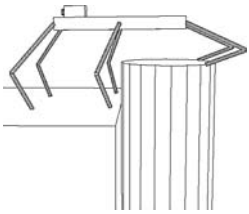
1.f Foots of middle legs goes to the supporting points between the edge of the shelf and current supporting points of front legs.

1.g Foots of rear legs goes to the edge of the shelf.



During this maneuver the height of the robot's body does not change after the item 1.a.

2. The difference δh satisfies the condition: $-0.2a \leq \delta h \leq 0.35a$. Orientation of the body is keeping horizontal and the succession of events looks like at the case $\delta h = 0$.
3. The difference δh belongs to the segment: $-0.35a \leq \delta h < -0.2a$. Orientation of the body is keeping horizontal.
   - 3.a The robot performs movement according to the item 1.a.
   - 3.b Foots of front legs are transferred to points placed under their attachment points. Final position of these foots are not correspond to a contact with any supporting surface. From the middle of this stage the body goes somewhat back inside to supporting polygon of middle and rear legs. Item 1.a provides a possibility to move front legs without intersection front and middle legs and without loss of static stability. But the motion forward front legs decreases static stability margin. Additional body's backward motion restores the static stability margin when front legs can not intersect middle legs by that time.
   - 3.c The robot's body goes down so that foots of front legs can be placed on the surface of the shelf and create a firm support. Simultaneously foots of front legs are transferred to the edge of the shelf. Middle and rear legs' foots are staying at its supporting points.
   - 3.d Items 1.c–1.g are accomplished one after another.

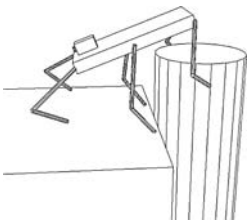


During this maneuver the height of the robot's body does not change after the item 3.c.

The requirement to provide the comfortable motion of the robot without collisions foots with the supporting surface constrains the value of δh.

It is clear that necessity to have enough static stability margin allows to descent the front edge of the robot a little lower than the top of the column. So we have $\delta h > -(l_1 + l_2)$. In reality supporting legs should be bended at knee for all the time. So the pointed restriction should be stronger. We adopt $\delta h \geq -0.6a$.

4. The difference δh belongs to the segment: $-0.6a \leq \delta h < -0.35a$
   - 4.a The robot performs movement according to the items 3.a, 3.b.
   - 4.b The robot's body goes down so that foots of front legs can reach the level $-0.35a$ relatively the top of the column. Simultaneously foots of fronts legs are placed to the vertical surface of the column under its previous supporting points.
   - 4.c Robot's body inclines so that center of the body goes down vertically, distance between attachment point and the foot of rear legs is keeping. The front edge of the robot's body becomes close to the top of the column. Static stability margin do not change practically. Foots of all legs are keeping their positions.
   - 4.d Foots of middle legs are transferred to the supporting points placed on the vertical surface of the column somewhat under the initial supporting points of front legs. This choice of middle legs' supporting points is performed due to the requirement that legs should not intersect the column. Robot's body does not move.
   - 4.e Foots of front legs are transferred to supporting points placed at $0.5a$ from the edge of the shelf.
   - 4.f Foots of middle legs are transferred to supporting points placed on the vertical surface on the column at $0.4a$ lower than initial supporting point of front legs. Robot's body goes forward keeping its inclination so that projection of body's center arrives to the supporting polygon of front and middle legs.
   - 4.g 4.g Foots of rear legs are transferred to supporting points placed at $0.2a$ lower than initial supporting points of middle legs. The body does not move.
   - 4.h Foots of middle legs are transferred to the edge of the shelf. The body goes along the line of its inclination. This motion of the body continues until item 4.k.
   - 4.i Foots of front legs are transferred as far as possible.



4.j Foots of middle legs are transferred to supporting points between the edge of the column and current supporting points of front legs.
4.k Foots of rear legs are transferred to the edge of the shelf.
5. Maneuver "across the column": $\delta h < -1.5a$. Robot should not care about going to the top of the column. It may go to the shelf when difference between height of its rear edge and the height of the shelf is equal to $0.5a$.
5.a The robot takes the symmetrical position on the column. It means that planes of all legs are perpendicular to the axis Ox.
5.b The robot goes by triple gait across the column with supporting point on vertical surface of the column. The robot's body is keeping the vertical position.
5.c When robot arrives to the opposite side of the column it can realize the maneuver of transferring to the shelf described in [1].

Adopted in this paper combination of values l_1, l_2 does not allow to consider $\delta h > 0.35a$ because in this cases knees of legs become lower than foots. This difficulty may be overcame by choice of broad track for footsteps, but it does not increase δh significantly.

The maneuver for preparation of regular motion along the shelf by triple gait is necessary after all these stages.

5 Realization of Prescribed Motion Computer Simulation

Computer simulation of robot dynamics was performed to evaluate efficiency and robustness of control algorithms of a robot. This simulation was performed by the program package "Universal Mechanism" [6]. Let us list general stages of data calculation at each step of computer simulation.

The Fig. 3 illustrates essential stages of computer simulation.

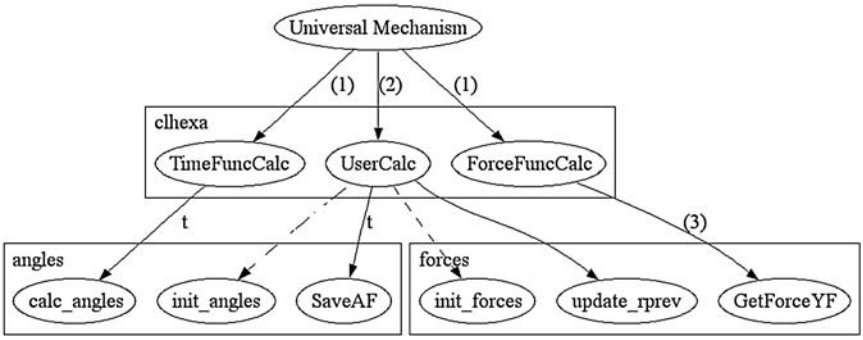


Fig. 3. Diagram of robot dynamics computer simulation

Adopted indications:

(1) Time and pointers to arrays of coordinates and velocities;
(2) Pointers to arrays of coordinates, velocities and accelerations, message index;
(3) Leg's number.

1. Desired positions of foots and of legs' attachment points are found. Joint angles are determined according to these positions (`calc_angles`).
2. Components of forces are calculated (`GetForceYF`).
3. Control voltages for joint drives are determined (in unit "angles").
4. The message about completion of these calculations is sending to the function for events simulation processing. Testing of process stages termination is fulfilled at that time (`SaveAF, update_rprev`).

The beginning and ending of simulation are processed individually (`init_angles`, `init_forces`).

The results of the simulations show effectiveness of developed methods of robot control and realizability of proposed scheme of robot motion.

Acknowledgement

This work was fulfilled under the financial support of Russian Fund of Fundamental Researches (04-01-00065).

References

1. Golubev Yu.F., Korianov V.V. (2003) Motion design for six-legged robot overcoming the vertical column by means of friction forces. Proc. of the 6-th International Conference CLAWAR-2003, pp. 609–616
2. Pugh D.R., Ribble E.A., Vohnout V.J., Bihari T.E., Walliser T.M., Patterson M.R., Waldron K.J., (1990) Technical Description of the Adaptive Suspension Vehicle. International Journal of Robotics Research, vol.9, No. 2, pp. 24–42
3. Wong H.C., Orin D.E., (1985) Control of a Quadruped Standing Jump and Running Jump Over Irregular Terrain Obstacles. Autonomous Robots, vol. 1, pp. 111–129
4. De Man H., Lefeber D., Vermeulen J. (1998) Design and Control of a One-Legged Robot Hopping on Irregular Terrain. In Proc. Euromech 375: Biology and Technology of Walking, pp. 173–180
5. Okhotsimsky D.E., Golubev Yu. F. (1984) Mechanics and Motion Control for Self-Acting Walking Vehicle. – M.: Nauka. The main publishing house of physico-mathematical literature, 1984. – 312 p
6. Pogorelov D.Yu. (1999) On numerical methods of modeling large multibody systems. Mech. and Mash. Theory 34, pp. 791–800. http://www.umlab.ru

Inverse Kinematic/Dynamic Analysis of a New 4-DOF Hybrid (Serial-Parallel) Pole Climbing Robot Manipulator

M.R. Zakerzadeh[1], M. Tavakoli[2], G.R. Vossoughi[3], and S. Bagheri[4]

[1] Graduate Student in Mechanical Engineering, Sharif University of Technology, Tehran, Iran
[2] Graduate Student in Mechanical Engineering, Sharif University of Technology
[3] Associate Professor of Mechanical Engineering, Sharif University of Technology
[4] Assistance Professor of Computer Science, Sharif University of Technology
m_zakerzadeh@mehr.sharif.edu

Abstract. This paper studies inverse kinematics and dynamics of a new 4-DOF hybrid (serial-parallel) manipulator. This kind of hybrid manipulator is specially designed for pole/column climbing applications with spatially bent and branching poles/columns. The proposed hybrid manipulator consists of a one-DOF rotary mechanism in series with a 3-DOF planar 3-R PR parallel mechanism. This combination provides 2 translational and 2 rotational degrees of freedom for the pole climbing robot. The inverse kinematic solution is firstly presented for the manipulator in closed forms. Then the inverse dynamic formulation is presented by the Newton-Euler approach for the proposed robot. The minimum force and moment equilibrium equations, that are essential for obtaining the actuator forces in the parallel mechanism legs and actuating moment in serial mechanism, are presented.

1 Introduction

Most robot manipulators designed till now are of the form of either serial or parallel kinematic chains. Serial manipulators are mechanisms which consist of a series of a single DOF active revolute or prismatic joints connecting the base plate to the moving platform. These manipulators have good operating characteristics, such as large workspace and high flexibility and have the disadvantages such as low precision, low stiffness and high vibration and deflection. To overcome the aforementioned drawbacks of the serial manipulators, there have been considerable developments in the field of manipulating structures, now known as the parallel manipulators. The parallel manipulators have their origin in the tire testing machine designed by Gough and Whitehall [1], and the flight simulator platform devised by Stewart [2]. The Stewart-Gough platform consists of two rigid bodies (referred to as the base and the platform)

connected through six extensible legs, each with spherical joints at both ends or with spherical joints at one end and universal joints at the other. Later, Hunt [3] suggested the use of the parallel actuated manipulators as robot manipulators.

Hybrid manipulator systems were the next generation of manipulator structures. Design of such manipulators by combining serial and parallel manipulator presented a compromise between high rigidity of fully parallel manipulators and extended workspace of serial manipulators. Currently, there has been an increasing interest in the hybrid manipulators although comparatively little literature on these manipulators is available because in contrast to the open-chain serial manipulators, the kinematics and dynamics analysis of parallel (or hybrid) manipulators present an inherent complexity due to presence of kinematics constraints and singularities. Shahinpoor [4] presented a kinematics analysis of a hybrid manipulation system which consists of two serially connected parallel manipulators.

The dynamics model for parallel and hybrid manipulators is very important especially if we target high speed and high acceleration applications. Euler-Lagrange and Newton-Euler formulations are the two broadly adopted approaches for dynamics analysis of robot manipulators. In the Euler-Lagrange approach, some times referred to as *the analytic approach,* the complete physical description of the manipulator is first incorporated in the Lagrangian in terms of a set of generalized coordinates and velocities and then a systematic procedures is followed to develop the Lagrangian equation of motion. On the other hand, in the Newton-Euler approach, also referred to as *the synthetic approach*, Newton's law and Euler's equation for linear and angular motion are directly applied to individuals bodies. Recently, the principle of virtual work is applied more frequently in parallel manipulators. Merlet [5] and Tsai [6] provide a detailed literature overview of different approaches.

Although the kinematic analysis of the parallel and hybrid manipulator have been investigated to some extent, works on their dynamics are relatively few. Dasgupta and Mruthyunjava [7] used the Newton-Euler approach to develop the closed-form dynamic equations of Stewart platform, considering all dynamic and gravity effect as well as the viscous friction at joints.

This paper presents an inverse kinematics and dynamics analysis of a new 4-DOF hybrid (serial-parallel) manipulator. This hybrid manipulator is specially designed for pole/column climbing applications with spatially bent and branching poles/columns. The proposed hybrid manipulator consists of a one-DOF serial mechanism and a 3-DOF planar 3-RPR parallel mechanism. This combination provides 2-translational and 2-rotational degrees of freedom for the pole climbing robot. The design, modeling and prototyping of this manipulator are discussed in [8] by Tavakoli et al. In that paper the reasons of requirement to these degrees of freedom for locomotion along tabular structures, with bends and branches, are discussed in detail. The mechanism and the coordinate systems of the manipulator are described in the next section. Then for inverse kinematics, the position analysis is performed and the velocity and acceleration formulae are derived in closed form. Based on the

Fig. 1. The proposed 4-DOF hybrid manipulator

force and moment equilibriums, the inverse dynamics of the manipulator is developed in the fourth section. The kinematic and dynamic analysis presented in this paper is fundamental in obtaining the optimal design and control of such robot.

2 Structural Description of the Manipulator

The considered hybrid manipulator consists of a serial rotating mechanism in series with a 3-DOF planar 3-R$\underline{\text{P}}$R parallel mechanism. The overall degrees of freedom for the manipulator are four which are two translations and two rotations. This manipulator consists of a fixed base plate and two platforms, as shown in Fig. 1. The first platform, called the mid platform has one degree of rotation with respect to the base plate. The second platform, called the top or moving platform, is linked to the mid platform by three legs in parallel. Using three prismatic joints in parallel results in three degrees of freedom (two translations and one rotation) for the moving platform with respect to the mid platform. The two translational degrees of freedom are restricted to a plane and the one rotational degree of platform is perpendicular to this plane. Therefore, the moving platform has four degrees of freedom (two translations and one rotation) with respect to the base plate. The hybrid manipulator with the relevant geometric parameters is shown in Fig. 2. Vertices of the fixed base plate are denoted as $a_i (i = 1, 2, 3)$ and vertices of the mid platform and moving platform are denoted as $b_i (i = 1, 2, 3)$ and $P_i (i = 1, 2, 3)$, respectively. A fixed references frame $U : O - XYZ$ is located at the center of the side a_1a_2 with Z-axis normal to the base plate and Y-axis directed along vector $\mathbf{a_1a_2}$ and X-axis perpendicular to Y and Z axes. Another reference frame $R : o - xyz$ is located on the mid platform as shown in Fig. 2 with x, y and z axes configuration as same as $U : O - XYZ$. Finally the moving reference frame $R' : o' - x'y'z'$ is located at the center of the side P_1P_2 with z'-axis normal to the moving platform and y'-axis directed along vector $\mathbf{P_1P_2}$ and x'-axis perpendicular to y' and z' axes (i.e. directed along vector $\mathbf{P_3o'}$). The rotation of the mid platform frame (i.e. R) with respect to the base platform

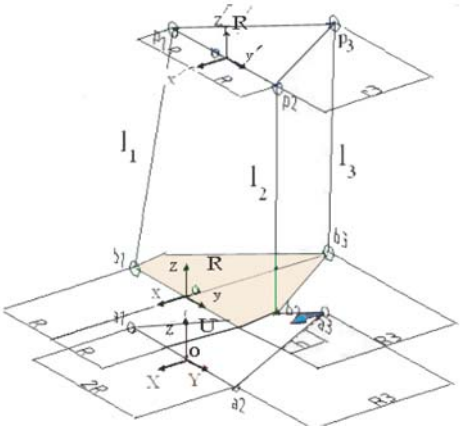


Fig. 2. The geometric parameter of the considered manipulator

frame (i.e. U) is denoted as ϕ. The length of each leg $P_i b_i$ is denoted as $l_i (i = 1, 2, 3)$ which are the length of prismatic actuators. (Note: The distance between the fixed plate and mid platform is exaggerated in these figures and it is negligible in kinematic equations).

3 Inverse Kinematics of the Manipulator

The study of manipulator kinematics is divided into two parts, inverse (or reverse) kinematics and forward (or direct) kinematics. The inverse kinematics problem involves mapping a known pose (position and orientation) of the moving platform of the manipulator to a set of input joint variables that will achieve that pose. The forward kinematics problem involves the mapping from a known set of input joint variable to a pose of the moving platform that results from those given inputs.

3.1 Position Analysis

For the inverse kinematics analysis, the position and orientation (and their time derivations) of the moving frame (in the fixed reference frame U) are considered known. In this paper, first we obtain the rotation matrices between U, R and R' frames. The rotation matrix between the fixed base frame U and the mid platform frame R, denoted as ${}^{U}_{R}\mathbf{T}$ is obtained as:

$$ {}^{U}_{R}\mathbf{T} = Rot(Z, \phi) = \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1} $$

where $s\phi$ and $c\phi$ indicate $\sin\phi$ and $\cos\phi$, respectively. The rotation matrix between the mid platform frame R and the top platform frame R' denoted as

${}^{R}_{R'}\mathbf{T}$ is only a rotation about y axis with magnitude θ, and expressed as:

$$ {}^{R}_{R'}\mathbf{T} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \tag{2} $$

where $s\theta$ and $c\theta$ indicate $\sin\theta$ and $\cos\theta$, respectively. The rotation matrix between the moving platform frame R' and the fixed base frame U, denoted as ${}^{U}_{R'}\mathbf{T}$ is obtained as:

$$ {}^{U}_{R'}\mathbf{T} = {}^{U}_{R}\mathbf{T}\,{}^{R}_{R'}\mathbf{T} \tag{3} $$

By substituting (1) and (2) in (3), one can obtain:

$$ {}^{U}_{R'}\mathbf{T} = \begin{bmatrix} c\phi\, c\theta & -s\phi & c\phi\, s\theta \\ s\phi\, c\theta & c\phi & s\phi\, s\theta \\ -s\theta & 0 & c\theta \end{bmatrix} \tag{4} $$

In the inverse kinematics analysis, the position of the moving frame R'(in the fixed reference frame U) is considered known and denoted by the vector $\mathbf{t}$ and expressed as

$$ \boldsymbol{t} = [x\ y\ z]^T \tag{5} $$

Vector $\boldsymbol{P}_i(i = 1,2,3)$, the coordinate of the moving platform point $P_i(i = 1,2,3)$ in the moving platform frame R', and vector $\boldsymbol{b}_i(i = 1,2,3)$, the coordinate of the mid platform point $\boldsymbol{b}_i(i = 1,2,3)$ in the mid platform frame R, are expressed as:

$$ (\mathbf{P}_1)_{R'} = \begin{bmatrix} 0 \\ -R \\ 0 \end{bmatrix}, \quad (\mathbf{P}_2)_{R'} = \begin{bmatrix} 0 \\ R \\ 0 \end{bmatrix}, \quad (\mathbf{P}_3)_{R'} = \begin{bmatrix} -r_3 \\ 0 \\ 0 \end{bmatrix} \tag{6} $$

$$ (\mathbf{b}_1)_R = \begin{bmatrix} 0 \\ -R \\ 0 \end{bmatrix}, \quad (\mathbf{b}_2)_R = \begin{bmatrix} -h \\ R \\ 0 \end{bmatrix}, \quad (\mathbf{b}_3)_R = \begin{bmatrix} -R_3 \\ 0 \\ 0 \end{bmatrix} \tag{7} $$

Coordinate of the above points in the fixed base frame U can be obtained by following transformations:

$$ (\mathbf{P_i})_U = {}^{U}_{R'}\mathbf{T}\,(\mathbf{P}_i)_{R'} + \mathbf{t} \tag{8} $$

$$ (\mathbf{b_i})_U = {}^{U}_{R}\mathbf{T}(\mathbf{b}_i)_R \tag{9} $$

By substituting (4), (5), (6) in (8) one can obtain:

$$ (\mathbf{P}_1)_U = \begin{bmatrix} Rs\phi + x \\ -Rc\phi + y \\ z \end{bmatrix}, \ (\mathbf{P}_2)_U = \begin{bmatrix} -Rs\phi + x \\ Rc\phi + y \\ z \end{bmatrix}, \ (\mathbf{P}_3)_U = \begin{bmatrix} -r_3 c\phi\, c\theta + x \\ -r_3 s\phi\, c\theta + y \\ r_3 s\theta + z \end{bmatrix} \tag{10} $$

and by substituting (1) and (7) into (9), one can obtain:

$$(\mathbf{b_1})_U = \begin{bmatrix} Rs\phi \\ -Rc\phi \\ 0 \end{bmatrix}, \quad (\mathbf{b}_2)_U = \begin{bmatrix} -hc\phi - Rs\phi \\ -hs\phi + Rc\phi \\ 0 \end{bmatrix}, \quad (\mathbf{b}_3)_U = \begin{bmatrix} -R_3c\phi \\ -R_3s\phi \\ 0 \end{bmatrix} \tag{11}$$

Letting $\mathbf{S}_i (i = 1, 2, 3)$ be the vector from points b_i to P_i (in the fixed frame U) as

$$\mathbf{S}_i = (\mathbf{P}_i) - (\mathbf{b}_i) \quad (i = 1, 2, 3) \tag{12}$$

then three kinematics equations of the manipulator can be obtained by writing following constraint equations

$$\|\mathbf{S}_i\| = l_i \quad (i = 1, 2, 3) \tag{13.1}$$

$$\hat{\mathbf{s}}_i = \frac{\mathbf{S}_i}{l_i} \quad (i = 1, 2, 3) \tag{13.2}$$

where $\hat{\mathbf{s}}_i$ is the unit vector along the leg's direction.

By substitution (10) and (11) into (12) and using (13.1), yields:

$$x^2 + y^2 + z^2 = l_1^2 \tag{14}$$

$$(x + hc\phi)^2 + (y + hs\phi)^2 + z^2 = l_2^2 \tag{15}$$

$$(x - r_3c\phi\, c\theta + R_3c\phi)^2 + (y - r_3s\phi\, c\theta + R_3s\phi)^2 + (z + r_3s\theta)^2 = l_3^2 \tag{16}$$

The last constraint is obtained from the fact that vector $\mathbf{P}_1\mathbf{P}_2$ is always perpendicular to the vector position of point o' in the fixed frame U(i.e. $\mathbf{t}$) and can be written as

$$\mathbf{P}_1\mathbf{P}_2 \cdot \mathbf{t} = 0 \tag{17}$$

By substituting (10) and (5) in (17) the last kinematics equation is obtained

$$xs\phi - yc\phi = 0 \tag{18}$$

The (14)–(16) and (18) are the governing kinematics equations of the proposed hybrid manipulator.

For obtaining the inverse kinematic equations of the manipulator, we first solve (18) for ϕ and obtain two solutions for ϕ as:

$$\phi_1 = A\tan 2(y, x) \tag{19}$$

$$\phi_2 = A\tan 2(-y, -x) \tag{20}$$

but only the first solution lead to the position (x, y, z) and is acceptable. To obtain the other inverse kinematics equations, we first compute $s\phi$ and $c\phi$ from (18)

$$s\phi = \frac{y}{\sqrt{x^2 + y^2}} \quad , \quad c\phi = \frac{x}{\sqrt{x^2 + y^2}} \tag{21}$$

By substituting (21) in (14), (15) and (16), the other inverse kinematic equations are obtained as

$$l_1 = \pm\sqrt{x^2 + y^2 + z^2} \tag{22}$$

$$l_2 = \pm\sqrt{(x^2 + y^2)\left(1 + \frac{h}{\sqrt{x^2 + y^2}}\right)^2 + z^2} \tag{23}$$

$$l_3 = \pm\sqrt{(x^2 + y^2)\left(1 - \frac{r_3}{\sqrt{x^2 + y^2}}c\theta + \frac{R_3}{\sqrt{x^2 + y^2}}\right)^2 + (z + r_3 s\theta)^2} \tag{24}$$

Since the negative solutions for $l_i (i = 1, 2, 3)$ haven't any geometric interpretation, only the positive solutions of (22)–(24) are acceptable. From (19) and (22)–(24) we realize that there are one inverse kinematics solutions for any given pose (position and rotation). For the forward kinematics solutions of this hybrid robot you can refer to [9].

3.2 Velocity and Acceleration Analysis

Because all the legs of the planar parallel mechanism have one rotational degree of freedom about the y axis of the R frame, the angular velocity and acceleration of these legs with respect to the R frame, expressed as $(\boldsymbol{\omega}_i)_R$ and $(\boldsymbol{\alpha}_i)_R$, can be written as:

$$(\boldsymbol{\omega}_i)_R = [0 \quad \omega_i \quad 0]^T, (\boldsymbol{\alpha}_i)_R = [0 \quad \alpha_i \quad 0]^T \tag{25}$$

Due to rotation of the mid platform the angular velocity of these legs in the global frame U can be obtained by the following equation:

$$\boldsymbol{\omega}_i = {}^U_R\mathbf{T}\,(\boldsymbol{\omega}_i)_R + [0 \quad 0 \quad \dot{\phi}]^T \tag{26}$$

that results to:

$$\boldsymbol{\omega}_i = [-\omega_i s\phi \quad \omega_i c\phi \quad \dot{\phi}]^T \tag{27}$$

and angular acceleration of these legs in the global U frame can be obtained by following equation:

$$\boldsymbol{\alpha}_i = {}^U_R\mathbf{T}\,(\boldsymbol{\alpha}_i)_R + [0 \;\; 0 \;\; \ddot{\phi}]^T + [0 \;\; 0 \;\; \dot{\phi}]^T \times [-\omega_i s\phi \quad \omega_i c\phi \quad 0]^T \tag{28}$$

that results to

$$\boldsymbol{\alpha}_i = [-\alpha_i s\phi - \dot{\phi}\omega_i c\phi \quad \alpha_i\, c\phi - \dot{\phi}\omega_i s\phi \quad \ddot{\phi}]^T \tag{29}$$

The velocity of point P_i can be written as:

$$(\dot{\mathbf{P}}_i)_U = (\dot{\mathbf{b}}_i)_U + \boldsymbol{\omega}_i \times (\mathbf{P}_i - \mathbf{b}_i)_U + \dot{l}_i \widehat{\mathbf{s}}_i \tag{30}$$

where $(\dot{\mathbf{P}}_i)_U$ and $(\dot{\mathbf{b}}_i)_U$ can be obtained from the time derivative of (10) and (11), respectively. By substituting each expression into (30) angular velocity of the legs, velocity of sliders and $\dot{\phi}$ can be computed in closed form as

$$\omega_1 = \frac{1}{q_1}\left(\frac{\dot{l}_1}{l_1}z - \dot{z}\right)\ ,\ \dot{l}_1 = \frac{1}{l_1}(x\dot{x} + y\dot{y} + z\dot{z}) \tag{31}$$

$$\omega_2 = \frac{1}{q_2}\left(\frac{\dot{l}_2}{l_2}z - \dot{z}\right)\ ,\ \dot{l}_2 = \frac{1}{l_1}[(x + hc\phi)\dot{x} + (y + hs\phi)\dot{y} + z\dot{z}] \tag{32}$$

$$\begin{aligned}\omega_3 = \frac{1}{q_3}\left[\frac{\dot{l}_3}{l_3}(z + r_3 s\theta) - (\dot{z} + r_3\dot{\theta}\, c\theta)\right]\ ,\ \dot{l}_3 = \frac{1}{l_3}[x\dot{x} + y\dot{y} + z\dot{z} + R_3 r_3\dot{\theta}\, c\theta \\ + q_1(r_3\dot{\theta}\, s\theta) + (R_3 - r_3 c\theta)(\dot{x}c\phi - x\dot{\phi}\, s\phi + \dot{y}s\phi + \dot{\phi}y\, c\phi) \\ + r_3(\dot{z}s\theta + z\dot{\theta}\, c\theta)]\end{aligned} \tag{33}$$

$$\dot{\phi} = \frac{1}{q_1}(\dot{y}c\phi - \dot{x}s\phi) \tag{34}$$

where

$$q_1 = ys\phi + xc\phi \quad q_2 = q_1 + h \quad q_3 = q_1 + R_3 - r_3 c\theta$$

Equations (31) to (34) give the closed-form formulation for the velocity analysis of this hybrid manipulator.

Again the acceleration of point P_i can be written as

$$(\ddot{\mathbf{P}}_i)_U = (\ddot{\mathbf{b}}_i)_U + \boldsymbol{\alpha}_i \times (\mathbf{P}_i - \mathbf{b}_i)_U + \boldsymbol{\omega}_i \times [\boldsymbol{\omega}_i \times (\mathbf{P}_i - \mathbf{b}_i)_U] + 2\boldsymbol{\omega}_i \times \dot{l}_i \widehat{\mathbf{s}}_i + \ddot{l}_i \widehat{\mathbf{s}}_i \tag{35}$$

where $(\ddot{\mathbf{P}}_i)_U$ and $(\ddot{\mathbf{b}}_i)_U$ can be obtained from the time derivative of $(\dot{\mathbf{P}}_i)_U$ and $(\dot{\mathbf{b}}_i)_U$, respectively. By substituting each expression into (35) angular acceleration of the legs, acceleration of sliders and can be computed in closed form as

$$\ddot{l}_1 = \frac{1}{l_1}(x\ddot{x} + y\ddot{y} + z\ddot{z} + \dot{x}^2 + \dot{y}^2 + \dot{z}^2 - \dot{l}_1^2)\,,$$

$$\alpha_1 = \frac{1}{q_1}\left[\frac{\ddot{l}_1}{l_1}z - \ddot{z} - 2\omega_1(\dot{y}s\phi + \dot{x}c\phi) + z\omega_1^2\right] \tag{36}$$

$$\ddot{l}_2 = \frac{1}{l_2}(x\ddot{x} + y\ddot{y} + z\ddot{z} + \dot{x}^2 + \dot{y}^2 + \dot{z}^2 - \dot{l}_2^2 + h[\ddot{x}c\phi + \ddot{y}s\phi + \dot{\phi}(\dot{y}c\phi - \dot{x}s\phi)]\,)\,,$$

$$\alpha_2 = \frac{1}{q_2}\left[\frac{\ddot{l}_2}{l_2}z - \ddot{z} - 2\omega_2(\dot{y}s\phi + \dot{x}c\phi) + z\omega_2^2\right] \tag{37}$$

$$\ddot{l}_3 = \frac{1}{l_3}(x\ddot{x} + y\ddot{y} + z\ddot{z} + \dot{x}^2 + \dot{y}^2 + \dot{z}^2 - \dot{l}_3^2 + \ldots\ldots)\,,$$

$$\begin{aligned}\alpha_3 = \frac{1}{q_3}\left[\frac{\ddot{l}_3}{l_3}(z + r_3 s\theta) - \ddot{z} - r_3\ddot{\theta}\, c\theta + r_3\dot{\theta}^2 s\theta - 2\omega_3(\dot{y}s\phi + \dot{x}c\phi + r_3\dot{\theta}^2 s\theta)\right. \\ \left. + (z + r_3 s\theta)\omega_3^2\right]\end{aligned} \tag{38}$$

$$\ddot{\phi} = \frac{1}{q_1}[\ddot{y}c\phi - \ddot{x}s\phi - 2\dot{\phi}(\dot{y}s\phi + \dot{x}c\phi)] \tag{39}$$

Equations (36) to (39) give the closed-form formulation for the acceleration analysis of this hybrid manipulator.

Because the mid-platform have only one rotational degree of freedom about the Z axis of the fixed reference frame, then the angular velocity and acceleration of this plate, expressed as $\boldsymbol{\omega}_{mid}$ and $\boldsymbol{\alpha}_{mid}$ respectively, can be written as

$$\boldsymbol{\omega}_{mid} = [0 \;\; 0 \;\; \dot{\phi}]^T, \quad \boldsymbol{\alpha}_{mid} = [0 \;\; 0 \;\; \ddot{\phi}]^T \tag{40}$$

Finally the moving platform angular velocity, expressed as $\boldsymbol{\omega}_p$, and the moving platform angular acceleration, expressed as $\boldsymbol{\alpha}_p$, with respect to the fixed base frame U can easily written as

$$\boldsymbol{\omega}_p = [-\dot{\theta}\, s\phi \quad \dot{\theta}\, c\phi \quad 0]^T, \quad \boldsymbol{\alpha}_p = [-\ddot{\theta}\, s\phi \quad \ddot{\theta}\, c\phi \quad 0]^T \tag{41}$$

4 Inverse Dynamic Analysis

In the inverse dynamic algorithm all positions, velocities and accelerations are determined from the known position, velocity and acceleration of the end-effector (as obtained in the previous sections) and then the vectors of joint forces required to generate the desired trajectory of moving platform are obtained.

4.1 Mid Platform Acceleration and Inertia

If $\mathbf{r}_{\text{mido}}$ be the position vector of the center of gravity of the mid-platform plate (with mass m_{mid}) in its local frame of reference (i.e. R), the same vector expressed in the fixed-base frame will be

$$\mathbf{r}_{\text{mid}} = {}^{U}_{R}\mathbf{T}\, \mathbf{r}_{\text{mido}} \tag{42}$$

Therefore, the acceleration of its center of gravity is

$$\mathbf{a}_{\text{mid}} = \boldsymbol{\alpha}_{\text{mid}} \times \mathbf{r}_{\text{mid}} + \boldsymbol{\omega}_{\text{mid}} \times (\boldsymbol{\omega}_{\text{mid}} \times \mathbf{r}_{\text{mid}}) \tag{43}$$

also the moment of inertia of the mid-platform can be transformed to the base frame U as

$$\mathbf{I}_{\text{mid}} = {}^{U}_{R}\mathbf{T}\, \mathbf{I}_{\text{mido}}\, {}^{U}_{R}\mathbf{T}^T \tag{44}$$

in which $\mathbf{I}_{\text{mido}}$ is the mid-platform moment of inertia in the coordinate frame attached to its center of gravity parallel to the mid-platform frame R.

4.2 Leg Acceleration and Inertia

A frame of reference (shown as frame $\{b_i\, x_i\, y_i\, z_i\}\, (i = 1, 2, 3)$ in Fig. 3) is attached to lower part of the leg with its origin at the mid platform-point, x axis along the leg, y axis along the rotating axis of the revolute joint

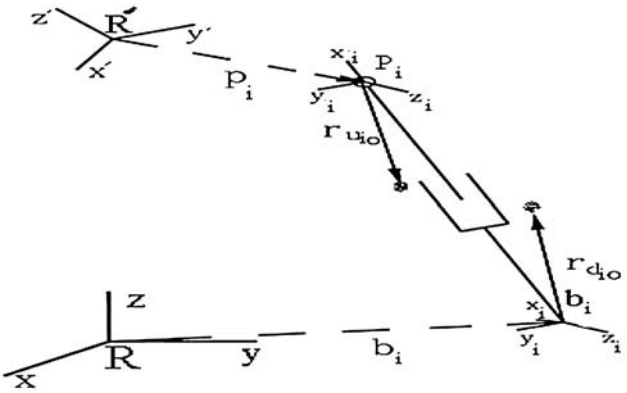


Fig. 3. The i-th local leg frame

(along y axis of frame R) and z axis perpendicular to the x axis and y axis according to the right hand rule. Another frame of reference (frame $\{P_i\, x_i\, y_i\, z_i\}\,(i = 1, 2, 3)$ in Fig. 3) with the same orientation is attached to the upper part of the leg with the origin at the platform-point. The kinematic and dynamic parameters of the leg are transformed to a fixed leg frame of reference (not shown separately in the Fig. 3) parallel to the fixed base frame U at the mid platform frame. The x, y and z axes of the moving lower frame $\{b_i\, x_i\, y_i\, z_i\}$ in reference frame are as

x-axis $\hat{\mathbf{x}}_i = \hat{\mathbf{s}}_\mathrm{i}$
y-axis $\hat{\mathbf{y}}_i = [-s\phi \;\; -c\phi \;\; 0]^T$
z-axis $\hat{\mathbf{z}}_i = \hat{\mathbf{x}}_i \times \hat{\mathbf{y}}_i$

so the transformation from the moving lower frame to the fixed leg frame is as

$$\mathbf{T}_i = [\hat{\mathbf{x}} \;\; \hat{\mathbf{y}} \;\; \hat{\mathbf{z}}] \tag{45}$$

If $\mathbf{r}_{\mathrm{doi}}$ and $\mathbf{r}_{uoi}$ denote the position vectors of the centers of gravity of the lower and upper parts of each leg (with mass m_{di} and m_{ui}, respectively) in their respective frames of reference, then they can be transformed to the fixed leg frame as

$$\mathbf{r}_{di} = \mathbf{T}_i\, \mathbf{r}_{\mathrm{doi}} \tag{46}$$

$$\mathbf{r}_{ui} = \mathbf{T}_i\, (\mathbf{r}_{uoi} + [l_i \;\; 0 \;\; 0]^T) \tag{47}$$

The acceleration of the center of gravity of the two parts are then

$$\mathbf{a}_{di} = \boldsymbol{\alpha}_i \times \mathbf{r}_{di} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{\mathrm{di}}) \tag{48}$$

$$\mathbf{a}_{ui} = \boldsymbol{\alpha}_i \times \mathbf{r}_{ui} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{ui}) + 2\boldsymbol{\omega}_i \times \dot{l}_i\, \hat{\mathbf{s}}_i + \ddot{l}_i\, \hat{\mathbf{s}}_i \tag{49}$$

The moment of inertia of the lower part $\mathbf{I}_{di}$ and of the upper part $\mathbf{I}_{ui}$ in the fixed leg frame can be obtained as

$$\mathbf{I}_{di} = \mathbf{T_i}\ \mathbf{I_{doi}}\ \mathbf{T_i}^T \tag{50}$$

$$\mathbf{I}_{ui} = \mathbf{T_i}\ \mathbf{I}_{uoi}\ \mathbf{T_i}^T \tag{51}$$

in which $\mathbf{I}_{\text{doi}}$ and $\mathbf{I}_{uoi}$ denote the leg's lower and upper part moment of inertia in the coordinate frame attached to the center of gravity of each part. These coordinates are parallel to the moving lower frame $\{b_i\, x_i\, y_i\, z_i\}$.

4.3 Moving Platform Acceleration and Inertia

If $\mathbf{r}_{po}$ represent the position vector of the center of gravity of the moving platform plate (with mass m_p) in its local frame of reference (i.e. R'), the same vector expressed in the fixed-base frame will be

$$\mathbf{r}_p = {}^{U}_{R'}\mathbf{T}\ \mathbf{r}_{po} \tag{52}$$

Therefore, the acceleration of the moving platform's center of gravity is

$$\mathbf{a}_p = \boldsymbol{\alpha}_p \times \mathbf{r}_p + \boldsymbol{\omega}_p \times (\boldsymbol{\omega}_p \times \mathbf{r}_p) + \ddot{\mathbf{t}} \tag{53}$$

also the moment of inertia of the moving platform can be transformed to the base frame U as

$$\mathbf{I}_p = {}^{U}_{R'}\mathbf{T}\ \mathbf{I}_{po}\ {}^{U}_{R}\mathbf{T}^T \tag{54}$$

in which $\mathbf{I}_{po}$ is the moving platform's moment of inertia in the coordinate frame attached to its center of gravity parallel to the moving platform frame R'.

4.4 Dynamic Equations

In this section by writing the Newton-Euler equation of some bodies the actuator forces in prismatic joints and actuator moment in rotary joint are obtained.

It is assumed that external forces and moments (if any) acting on the moving platform is known and given as $\mathbf{F}_{\text{ext}}$ and $\mathbf{M}_{\text{ext}}$ in the local frame of reference (i.e. R'). The position vector of the force point is $\mathbf{r}_{\text{ext}}$ in the local frame of reference. These vectors can be transformed to the global fixed reference frame by the appropriate transformation matrices.

Hence, Euler equation for all the system (taking the moment about the global reference point) can be written as

$$\mathbf{M}_{\text{base}} = \mathbf{q}_4 \tag{55}$$

where

$$\mathbf{q}_4 = -{}^{U}_{R'}\mathbf{T}\,\mathbf{M}_{\text{ext}} - \left({}^{U}_{R'}\mathbf{T}\,\mathbf{r}_{\text{ext}} + \mathbf{t}\right) \times {}^{U}_{R'}\mathbf{T}\,\mathbf{F}_{\text{ext}} + \mathbf{I}_p\alpha_p + \mathbf{I}_{\text{mid}}\alpha_{\text{mid}} + \boldsymbol{\omega}_p \times \mathbf{I}_p\omega_p$$
$$+ \boldsymbol{\omega}_{\text{mid}} \times \mathbf{I}_{\text{mid}}\boldsymbol{\omega}_{\text{mid}} + \sum_{i=1}^{3}\left[\left(\mathbf{I}_{ui} + \mathbf{I}_{di}\right)\boldsymbol{\alpha}_i + \boldsymbol{\omega}_i \times \left(\mathbf{I}_{ui} + \mathrm{I}_{di}\right)\boldsymbol{\omega}_i\right]$$
$$+ \sum_{i=1}^{3}\left[\mathbf{r}_{ui} + (\mathbf{b}_i)_u\right] \times m_{ui}(\mathbf{a}_{ui} - \mathbf{g}) + \left[\mathbf{r}_{di} + (\mathbf{b}_i)_u\right] \times m_{di}(\mathbf{a}_{di} - \mathbf{g})]$$
$$+ (\mathbf{r}_p + \mathbf{t}) \times m_p(\mathbf{a}_p - \mathbf{g}) + (\mathbf{r}_{\text{mid}}) \times m_{\text{mid}}(\mathbf{a}_{\text{mid}} - \mathbf{g}) = 0$$

then the actuator moment in the serial mechanism, expressed as τ can be obtained as

$$\tau = \mathbf{M}_{\text{base}}.\mathbf{K} \tag{56}$$

Letting $(\mathbf{F}_{\text{ri}})_{\mathbf{R}} = [\mathrm{F}_{\text{ri}_x}\ \mathrm{F}_{\text{ri}_y}\ \mathrm{F}_{\text{ri}_z}]^T$ be the constraint forces at the revolute joints connecting the upper part of the legs in parallel mechanism to moving platform in mid platform frame (i.e. R), then their corresponding value in the global frame will be;

$$\mathbf{F}_{\text{ri}} = {}^{U}_{R}\,\mathbf{T}\ (\mathbf{F}_{\text{ri}})_{\text{R}} = \mathbf{F}_{\text{ri}} = \begin{bmatrix} F_{\text{ri}_x}c\phi - F_{\text{ri}_y}s\phi \\ F_{\text{ri}_x}s\phi + F_{\text{ri}_y}c\phi \\ F_{\text{ri}_z} \end{bmatrix} \tag{57}$$

By taking the moment of the complete leg (including the lower and upper parts) about point b_i we can obtain

$$\mathbf{M}_{\text{ri}} + \mathbf{S}_i \times \mathbf{F}_{\text{ri}} - \mathbf{M}_{\text{midi}} = \mathbf{C}_i \tag{58}$$

where

$$C_i = -(\mathbf{I}_{ui} + \mathbf{I}_{di})\,\boldsymbol{\alpha}_i - \boldsymbol{\omega}_i \times (\mathbf{I}_{ui} + \mathbf{I}_{di})\,\boldsymbol{\omega}_i - \mathbf{r}_{ui} \times m_{ui}(\mathbf{a}_{ui} - \mathbf{g}) - \mathbf{r}_{di} \times m_{di}(\mathbf{a}_{di} - \mathbf{g})$$

and $\mathbf{M}_{\text{ri}}$ is the constraining moment at the upper revolute joints exerted by the leg's upper part on the moving platform and $\mathbf{M}_{\text{midi}}$ is the constraining moment at the lower revolute joints exerted by the mid platform on the lower part of the legs. As the revolute joints don't exert any moment along the revolute axis (here it is the y axis of R frame) we can eliminate $\mathbf{M}_{\text{ri}}$ and $\mathbf{M}_{\text{midi}}$ by taking dot product of both sides of (59) with this direction. Noting that this direction in global reference frame is as ${}^{U}_{R}\mathbf{T}\,[0\ \ 1\ \ 0]^T$, we have

$$(\mathbf{y})_U = [-s\phi\ \ c\phi\ \ 0]^T$$

By performing the dot product and substitute (57) we obtain these three scalar equations

$$-q_1\,F_{r1z} + z\,F_{r1x} = \mathbf{C}_1 \cdot (\mathbf{y})_U \tag{59}$$

$$-q_2\,F_{r2z} + z\,F_{r2x} = \mathbf{C}_2 \cdot (\mathbf{y})_U \tag{60}$$

$$-q_3\, F_{r3z} + (z + r_3 s\theta)\, F_{r3x} = \mathbf{C}_3 \cdot (\mathbf{y})_U \tag{61}$$

now taking the moments on the moving platform about the platform reference point, and using the Euler's equation for the moving platform gives

$$\sum_{i=1}^{3} [({}^{U}_{R'}\mathbf{T}\ (\mathbf{P}_i)_{R'}) \times \mathbf{F}_{\mathrm{ri}}] + \sum_{i=1}^{3} \mathbf{M}_{\mathrm{ri}} = \mathbf{q}_5 \tag{62}$$

where

$$\mathbf{q}_5 = m_p \mathbf{r}_p \times (\mathbf{a}_p - \mathbf{g}) + \mathbf{I}_p \alpha_p + \boldsymbol{\omega}_p \times \mathbf{I}_p \boldsymbol{\omega}_p - {}^{U}_{R'}\mathbf{T}\ \mathbf{M}_{\mathrm{ext}} - {}^{U}_{R'}\mathbf{T}\, \mathbf{r}_{\mathrm{ext}} \times {}^{U}_{R'}\mathbf{T}\, \mathbf{F}_{\mathrm{ext}}$$

by taking dot product of both side with $(\mathbf{y})_U$ we obtain another scalar equation as

$$r_3(F_{r3z} c\theta + F_{r3x} s\theta) = \mathbf{q}_5 \cdot (\mathbf{y})_U \tag{63}$$

by solving (61) and (63) together we obtain

$$F_{r3x} = \frac{\mathbf{C}_3 \cdot (\mathbf{y})_U + \frac{\mathbf{q}_5 \cdot (\mathbf{y})_U}{r_3\, c\theta}}{q_3 \tan\theta + (z + r_3 s\theta)} \tag{64}$$

$$F_{r3z} = \frac{\mathbf{q}_5 \cdot (\mathbf{y})_U}{r_3\, c\theta} - F_{r3x} \tan\theta \tag{65}$$

when $\theta = 0$ we can easily find these answers by solving (61) and (63).

By writing the Newton equation of the moving platform we obtain:

$$\sum_{i=1}^{3} \mathbf{F}_{\mathrm{ri}} = \mathbf{q}_6 \tag{66}$$

where

$$\mathbf{q}_6 = m_p(\mathbf{a}_p - \mathbf{g}) - {}^{U}_{R'}\mathbf{T}\mathbf{F}_{\mathrm{ext}} \tag{67}$$

this equation yields three scalar equations and by taking dot product in two directions (first: $(\mathbf{x})_U = [c\phi\ \ s\phi\ \ 0]^T$ and second: $(\mathbf{z})_U = [0\ \ 0\ \ 1]^T$) we obtain:

$$F_{r1x} + F_{r2x} + F_{r3x} = \mathbf{q}_6 \cdot (\mathbf{x})_U \tag{68}$$

$$F_{r1z} + F_{r2z} + F_{r3z} = \mathbf{q}_6 \cdot (\mathbf{z})_U \tag{69}$$

solving (59), (60), (68) and (69) gives

$$F_{r1z} = \frac{z[\mathbf{q}_6 \cdot (\mathbf{x})_U - F_{r3x}] - [(\mathbf{C}_1 + \mathbf{C}_2) \cdot (\mathbf{y})_U] - q_2[\mathbf{q}_6 \cdot (\mathbf{z})_U - F_{r3z}]}{q_1 - q_2} \tag{70}$$

$$F_{r2z} = (\mathbf{q}_6 \cdot (\mathbf{z})_U - F_{r3z}) - F_{r1z} \tag{71}$$

$$F_{r1x} = \frac{q_1 F_{r1z} + \mathbf{C}_1 \cdot (\mathbf{y})_U}{z} \tag{72}$$

$$F_{r2x} = \frac{q_2 F_{r2z} + \mathbf{C}_2 \cdot (\mathbf{y})_U}{z} \tag{73}$$

The reason that we only obtain these unknown and not more can be found in following.

Finally, considering the upper part of each leg and Newton's equation gives

$$\mathbf{F}_{Pi} - \mathbf{F}_{\mathrm{ri}} = m_{ui}(\mathbf{a}_{ui} - \mathbf{g}) \tag{74}$$

in which $\mathbf{F}_{Pi}$ is the vector force at the prismatic joint exerted by the lower part on the upper part.

As we are interested in the axial component of $\mathbf{F}_{Pi}$ i.e. the actuating force, we can take the component of the above equation in the direction of the leg by taking the dot product with $\hat{\mathbf{s}}_i$ obtaining the following equations

$$F_{act1} = q_1\, F_{r1x} + z\, F_{r1z} + m_{u1}(\mathbf{a}_{u1} - \mathbf{g})\,.\,\hat{\mathbf{s}}_1 \tag{75}$$

$$F_{act2} = q_2\, F_{r2x} + z\, F_{r2z} + m_{u2}(\mathbf{a}_{u2} - \mathbf{g})\,.\,\hat{\mathbf{s}}_2 \tag{76}$$

$$F_{act3} = q_3\, F_{r3x} + (z + r_3 s\theta)\, F_{r3z} + m_{u3}(\mathbf{a}_{u3} - \mathbf{g})\,.\,\hat{\mathbf{s}}_3 \tag{77}$$

Equations (56) and (75)–(77) give the inverse dynamic solution of this hybrid manipulator.

5 Numerical Example

Firstly the kinematic parameter of the parallel mechanism are given as

$$R_3 = 0.2\,(\mathrm{m}), \quad \mathrm{r}_3 = 0.14\,(\mathrm{m}), \quad \mathrm{h} = 0.02\,(\mathrm{m})$$

and the dynamic parameters of the mid platform and moving platform are

$$\mathbf{r}_{mido} = [-0.05\ \ 0\ \ 0]^T\,(m), \mathbf{I}_{mido} = \begin{bmatrix} 0.004 & 0 & 0 \\ 0 & 0.005 & 0 \\ 0 & 0 & 0.02 \end{bmatrix} (\mathrm{kg.m}^2), m_{\mathrm{mid}} = 2\,(\mathrm{kg})$$

$$\mathbf{r}_{po} = [0.02\ \ 0\ \ 0]^T\,(m), \mathbf{I}_{po} = \begin{bmatrix} 0.005 & 0 & 0 \\ 0 & 0.006 & 0 \\ 0 & 0 & 0.04 \end{bmatrix} (\mathrm{kg.m}^2)\,, m_{\mathrm{p}} = 5\,(\mathrm{kg})$$

The dynamic parameters of the legs are

$$\mathbf{r}_{doi} = [0.075\ 0\ 0]^T\,(mm),$$

$$\mathbf{I}_{doi} = \begin{bmatrix} 0.00012 & 0 & 0 \\ 0 & 0.002 & 0 \\ 0 & 0 & 0.002 \end{bmatrix} (\mathrm{kg.m}^2)\ , m_{di} = 0.6\,(\mathrm{kg})$$

$$\mathbf{r}_{uoi} = [-0.12\ 0\ 0]^T(mm),$$

$$\mathbf{I}_{uoi} = \begin{bmatrix} 0.00018 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{bmatrix} (\mathrm{kg.m}^2)\ , m_{ui} = 0.4\,(\mathrm{kg})$$

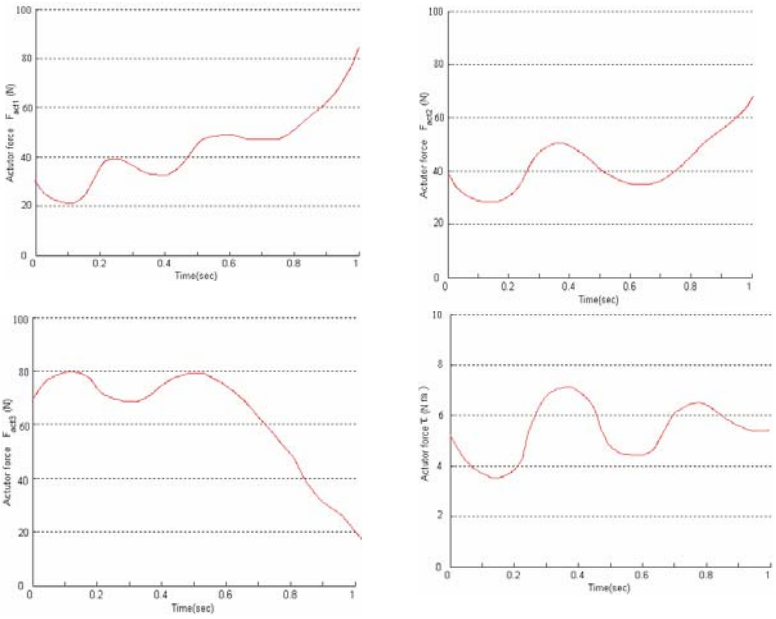


Fig. 4. The actuator forces for the sample numerical example

The acceleration due to gravity is

$$\mathbf{g} = [0 \quad 0 \quad -9.806]^T (\mathrm{m/s^2})$$

In this numerical example, the moving platform moves for one second and the trajectory of this platform is given as

$$\begin{aligned} x &= -0.1\,t^4 + 0.02\,t^2\,(\mathrm{m}) \\ y &= -0.05\,t^4 + 0.04\,t^2 (\mathrm{m}) \\ z &= -0.2\,t^4 - 0.04\,t^2 (\mathrm{m}) \\ \theta &= 0.4\,\sin(2\pi\,t)\,(\mathrm{rad}) \end{aligned}$$

so the acceleration quantities are made continues.

Assuming that there is no external force and moment acting on the moving platform, the actuator forces are shown in Fig. 4.

References

1. Gough V.E., Whitehall S.G. (1962) Universal tyre test machine", in: proceedings of the 9th international technical congress, FISITA, p. 177
2. Stewart D. (1966) A platform with six degrees of freedom", proc. Inst. Mech. Eng., part I, vol. 180, no. 15, pp. 371–386

3. Hunt K.H. (1978) Kinematic Geometry of mechanism", Clarendon press, Oxford
4. Shahinpoor M. (1992) Kinematics of parallel-serial (hybrid) manipulator. Journal of Robotic System, vol. 9, no. 1, pp. 17–36
5. Merlet J.P. (1997) Les robots paralleles Collection robotique", Hermes, Paris, 2nd edition
6. Tsai L.W. (1999) Robot analysis- The Mechanics of Serial and Parallel Manipulators, John Wiley & Sons, 1st edition
7. Dasgupta B., Mruthyunjava T.S. (1998) A Newton-Euler formulation for the inverse dynamics of the stewart platform manipulator", Mechanism and Machine Theory, vol. 33, pp. 1135–1152
8. Tavakoli M., Zakerzadeh M.R., Vossoughi G.R., Bagheri S. (2004) Design, Modeling and Prototyping of a Serial/Parallel Pole Climbing and Manipulating Robot with Minimum DOFs for Construction Works. Submitted to 7th international on climbing and walking robots and the support technology for mobile machines, Madrid, September 2004
9. Zakerzadeh M.R., Vossoughi G.R., Bagheri S., Tavakoli M. (2004) Kinematic Analysis of a New 4-DOF Hybrid Pole Climbing Manipulator. 12th Mediterranean conference on control and automation MED'04, 6–9 June 2004, Turkey

Climbing Without a Vacuum Pump

Werner Brockmann and Florian Mösch

University of Lübeck, Institute of Computer Engineering, Ratzeburger Allee 160, 23562 Lübeck, Germany
{brockmann|moesch}@iti.uni-luebeck.de

Summary. The most crucial issue with climbing robots it the question of how to achieve sufficient adhesive forces. Normally suction cups are used in combination with at least one vacuum pump. Unfortunately, weight, cost, and complexity of the robot are determined heavily by that. This paper describes how so-called passive suction cups can be used for climbing robots. They are evacuated simply by pressing them to the surface and thus simple, robust, and cheap. A simple mechanism is also sufficient to release them. Hence, climbing robots can be built simpler and cheaper as the example of the climbing robot DEXTER illustrates.

1 Introduction

Climbing on an arbitrarily inclined, e.g. vertical or hanging over surfaces is commonly done by adhering to the surface through the suction principle. Most machines use either a single large suction cup [1–4] or multiple small suction cups on each foot [5, 6]. But there also exist climbing robots which are driven by multiple small suction cups that are arranged on a (large) turning wheel, like HYDRA V, or on a rope [7], thus allowing continuous locomotion. Alternatively, the body of a climbing robot is used as a large suction cup with a wheeled drive-system underneath [8, 9]. In general, these suction cups are evacuated actively by at least one vacuum pump which is mounted on the robot. This is called "active suction", or "active suction cups" respectively, hereinafter.

Vacuum pumps make climbing robots not only noisy. They also increase the weight and the costs of a robot. Additionally, the design complexity and the weight are increased due to additional vacuum tubes, muffles, valves and so forth. Hence, it is desirable to avoid a separate installation for vacuum generation and transportation. There are only a few machines [10] using "passive ouction (cups)" (see for instance Fig. 1a). This means machines that adhere to the surface by suction cups made of elastic material. These cups are evacuated by simply pressing them to the surface. Hence, vacuum is caused

(a) Suction cups without straps. (b) A 75 mm cup with a strap.

Fig. 1. Different types of suction cups

without any vacuum pump by utilizing the robot's locomotion system which is already there.

On the other hand, releasing a passive suction cup must normally be done by tearing it off the surface. Therefore large forces are needed which cause mechanical stress and thus finally a rigid and heavy mechanical construction. In order to keep the forces low which are needed for tearing off, multiple small suction cups are arranged in a row and only a single cup is torn off at a time. The standard construction of passive suction cups furthermore allows only small angles in which the cups can be placed on the surface. Otherwise an edge of the suction cup may get folded and no vacuum is built. This results in severe kinematic restrictions if compared to a design with active suction cups. Thus e.g. chains with passive suction cups [10] or wheels with a large diameter (HYDRA V) are used for locomotion.

2 New Approach to Passive Suction

2.1 Basic Approach

The aim of our approach is to overcome above limitations and to use passive suction cups as flexible as active suction cups. We therefore utilize passive suction cups with a strap, as Fig. 1b shows. The cup is pressed to the surface without pulling the strap. By that, it is evacuated and produces hence an adhesive force. The strap is pulled to release the vacuum before a suction cup is to be lifted in order to avoid any remaining adhesive force. This pulling may be done e.g. by a small and light-weight radio control servo or solenoid. According to that, no vacuum pump, muffles, tubes etc. are needed. Instead, an adhesive force is produced without steady energy consumption. A climbing robot is thus operated very power-efficiently and has a reduced weight.

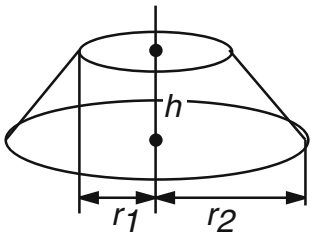


Fig. 2. Schematic of a passive suction cup

2.2 Behavior of Passive Suction Cups

In order to analyze the behavior of a passive suction cup, the volume V underneath it may simply be approximated by a frustum as illustrated in Fig. 2. Here radius r_1 is close to the fixation of the cup. In contrast, r_2 is the radius of the effective area which adheres to the surface. Due to the elasticity of the cup material, it may vary depending on the pressing and pulling forces. The effective volume is determined by (1).

$$V = \frac{h}{3}\pi \left(r_1^2 + r_1 r_2 + r_2^2\right) \tag{1}$$

Without an external force, the suction force and the reset force caused by the elasticity of the suction cup are in an equilibrium state. An external pulling force then increases the effective height h, and thus the volume V. Because there is no exchange of air, the vacuum P is increased as well. Based on the assumption that the form remains that of a frustum, this can be expressed by (1) and (2) where indices 1 and 2 refer to two different pulling forces.

$$P_1 V_1 = P_2 V_2 \tag{2}$$

In practice, r_1 remains nearly the same when an external force is applied due to a rigid coupling to the fixation (see. Fig. 1). But r_2 increases due to elasticity of the cup material. Hence, the effective area as well as the volume V and the vacuum P are increased as well. In total this leads to a super-proportional increase of the adhesive force, until it reaches the external force. Nevertheless, the height h of a cup increases with the external force, but less than proportional. But the variation of height h is smaller the stronger the vacuum P is after pressing on.

The limit of adhesion is reached when the adhesive-force, or the pulling force, respectively, gets equal to the tear-off force $F_{\max}$ (3).

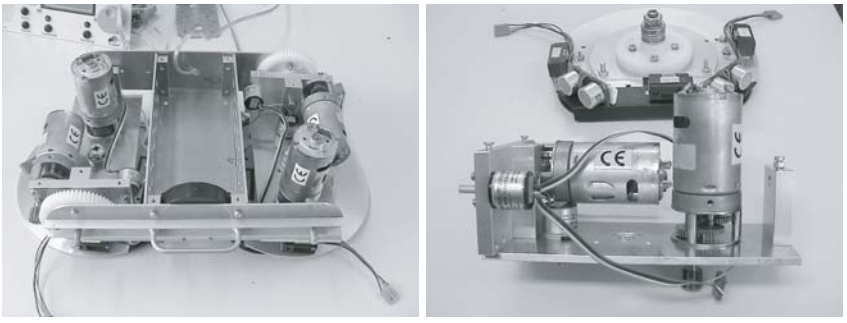
$$F_{\max} = \pi r_{2_{\max}}^2 P_{\max} \tag{3}$$

The maximum of r_2 is given by the suction cup itself and little bit smaller than half of the outer diameter. In contrast, the maximum vacuum $P_{\max}$

depends on the vacuum which is achieved when pressing the cup on. Hence a cup should be pressed on as strong as possible in order to obtain the smallest volume possible underneath the suction cup. Than not only the largest tear-off force $F_{\max}$ is obtained, but also the smallest variation of height h as well as the largest safety concerning leakage because it lasts the longest time until a hold force gets too small.

3 Design Example

The climbing robot DEXTER (DEXTerous Electric Rover) built at our institute shall be described as a first practical example. It shows that this kind of passive suction can be directly combined with kinematic structures known from climbing robots using active suction and allows the same dexterity. Its locomotion principle is similar to e.g. [4], except that the body is not articulated (Fig. 3a). It has four degrees of freedom, i.e. each foot is capable of rotating around two axes (Fig. 3b). This enables DEXTER to move on an arbitrarily inclined surfaces in two modes, i.e. by turning over and by swiveling around the stemming foot. By that, it is also capable of crossing from the floor to a wall and from a wall to another wall or to the ceiling.



(a) Assembly of DEXTER. (b) Two orthogonal axes per foot.

Fig. 3. Kinematics of DEXTER

Besides position sensors on each axis, two inclination sensors are used to determine the absolute orientation of the body. Based on this, the movement of the robot is controlled. The inclination information is additionally used to simplify the axis controllers by precompensating the gravity force. DEXTER is also equipped with three ultrasonic and two infrared range sensors on each foot. In order to perceive the environment in a wide range with suitable resolution, a swinging foot is rotated close to the surface in order to perform a scanning movement.

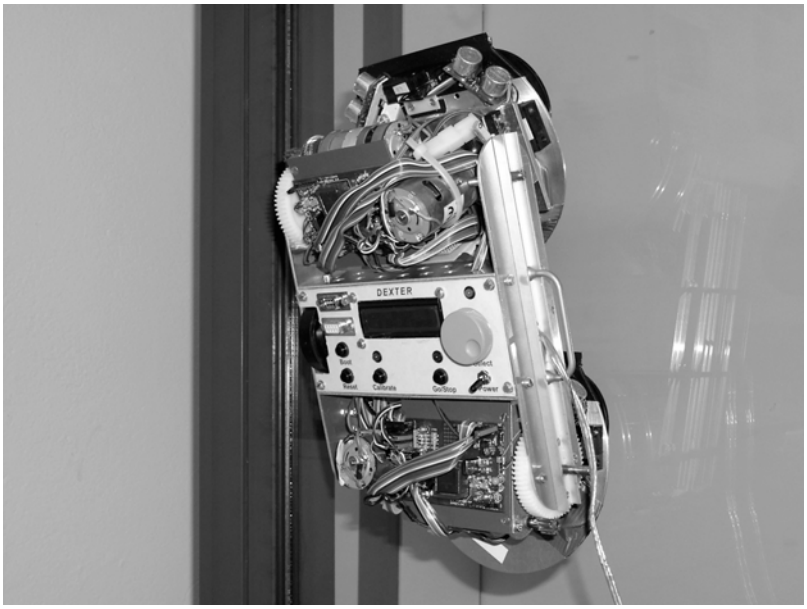


Fig. 4. DEXTER climbing at a window

Due to passive suction cups which are released by an ordinary miniature RC-servo, DEXTER only needs a 12 VDC electric energy supply (see cable in Fig. 4). It has a total size of $36.5 \times 22 \times 13\,\mathrm{cm}^3$ and a weight of about 3 kg.

4 General Design Considerations

Necessarily, passive suction cups are made of elastic material. This is needed for evacuating the volume underneath the cup. Due to that, elasticity is introduced into the robot's kinematics by standard passive suction cups in two ways. First, an elastic suction cup is prone to bending. Second, a cup increases its height depending on the pulling force. Hence, the robot has no rigid foothold which introduces elasticity into the kinematic structure and may cause control problems. But a simple constructive method eliminates the first problem and reduces the second one, namely to arrange multiple suction cups on a single foot in a distance larger than the diameter of a single suction cup (see Fig. 5). This distance acts as a rigid coupling which avoids bending and also like a lever which reduces the effective variation of suction cup height. Because severe bending is avoided, the danger of damaging a suction cup is also significantly reduced such that large torques and forces are compensated very effectively.

Three passive suction cups (Fig. 5a) may be connected by short straps (black in Fig. 5). Hence, only a short stroke is needed to release the vacuum.

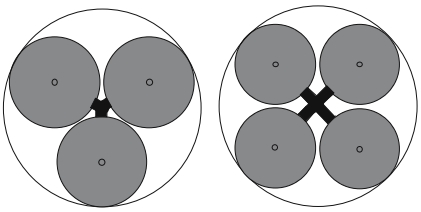


Fig. 5. Suggested suction cup arrangements on a foot

In contrast, if four suction cups (Fig. 5b) are connected at one central point in order to simplify the lifting mechanism, the straps and hence the stroke have to be longer.

The advantage of using four cups are an increased total adhesive force and stability against bending. Multiple suction cups are also useful to improve the safety of the robot by redundancy as well as by increasing the safety factor of the achievable adhesive force. Due to the simplicity and the inexpensiveness of passive suction cups, this is nearly free of additional costs and weight. For instance, a single suction cup used by DEXTER is capable of holding eight times the weight of the robot and has a weight of a few grams. And DEXTER has three of them on each foot (in an arrangement according to Fig. 5a).

Of course, the problem of passive suction cups loosening by the time, only occurs when the robot is not in steady movement. Fortunately, climbing robots use a dexterous locomotion system. This allows a foot to be kept on the same place for a longer time by simply pressing it again and again to the surface in order to reevacuate the cup without moving the robot.

An ordinary, light-weight RC-Servo plus a simple tearing mechanism, e.g. a rope, replace at least one vacuum pump, valves, tubes, muffles etc. Thus not only costs are reduced but also the weight of the robot. Hence lower static as well as dynamic forces and torques occur. The robot thus can be built less rigid and less massive. This reduces weight as well as costs further. Additionally, the design may also be simplified.

Last but not least, it should be noticed that pulling forces act against the adhesive force. Thus also the friction force holding the robot at its position on an inclined surface is reuced. Due to a friction coefficient being always smaller than 1, the friction force is always smaller than the adhesive force. Consequently, the robot may slip, although a suction cup is not torn off. The safety factor should hence be designed such that slippage does not occur even in the worst case. Otherwise the control system has to compensate it which tends to get more complicated than increasing the safety factor by applying more and/or larger suction cups.

5 Conclusions

Passive suction has several advantages over active suction. First, no vacuum pump, tubes and so forth are needed. Instead, passive suction cups are simple, robust, light-weight, energy-efficient, and cheap. Large adhesive forces are achieved easily by increasing the diameter and/or the number of a suction cups. With the principles described in this paper, this allows to build climbing robots having the same dexterity as robots using active suction cups, but significantly reduced weight, complexity, and costs. It is very simple to design climbing robots with large adhesive forces. Furthermore, no energy is needed for keeping the vacuum, and the robot moves with lower noise. Hence, passive suction cups seem to be important to build fully autonomous climbing robots, because only electric power is needed which is easily delivered by accumulators. On the other hand, even a very small gap between a passive suction cup and the surface will cause the vacuum to break down by the time. The robots are thus not allowed to stand still for a longer period of time. Additionally, requirements concerning flatness and cleanness of the surface, on which the robot is climbing, may be a little bit harder than for active suction.

References

1. Prieto M, Armada M (1998) Experimental issues on wall climbing gait generation for a six-legged robot. In: Proc. 1st Int. Symp. Mobile, climbing and walking robots. CLAWAR'98, Brussels: 125–130
2. White TS, Hewer N et al. (1998) SADIE, A climbing robot for weld inspection in hazardous environments. In: Proc. 1st Int. Symp. Mobile, climbing and walking robots. CLAWAR'98, Brussels: 385–389
3. Yoneda K, Ota Y et al. (2001) Design of light-weight wall climbing quadruped with reduced degrees of freedom. In: Berns K, Dillmann R (eds) Proc. 4th Int. Conf. Climbing and walking robots. CLAWAR 2001, Prof. Engineering Publ., Bury St. Edmunds, London: 907–912
4. Lal Tummala R, Mukherjee R et al. (2002) Climbing the walls. IEEE Robotics & Automation Magazine, Vol. 9, No. 4: 10–19
5. Hirose S, Kawabe K (1998) Ceiling walk of quadruped wall climbing robot NINJA-II. In: Proc. 1st Int. Symp. Mobile, climbing and walking robots. CLAWAR'98, Brussels: 143–147
6. Gimenez A, Abderrahim M, Balaguer C (2001) Lessons from the ROMA I inspection robot development experience. In: Berns K, Dillmann R (eds) Proc. 4th Int. Conf. Climbing and walking robots. CLAWAR 2001, Prof. Engineering Publ., Bury St. Edmunds, London: 913–920
7. Schmid D, Maeule B, Merrig H (1998) Tracked robot goes up the wall. In: Proc. 1st Int. Symp. Mobile, climbing and walking robots. CLAWAR'98, Brussels: 33–34
8. Xu D, Liu S et al. (1999) Design of a wall cleaning robot with a single suction cup. In: Virk GS, Randall M, Howard D (eds) Proc. 2nd Int. Conf. Climbing

and walking robots. CLAWAR'99, Prof. Engineering Publ., Bury St. Edmunds, London: 405–411
9. Weise F, Köhnen J et al. (2001) Non-destructive sensors for inspection of concrete structures with a climbing robot. In: Berns K, Dillmann R (eds) Proc. 4th Int. Conf. Climbing and walking robots. CLAWAR 2001, Prof. Engineering Publ., Bury St. Edmunds, London: 945–952
10. Household robot RACCOON – autonomous window cleaner. (2002) Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Germany. http://www.ipa.fhg.de/Arbeitsgebiete/BereichC/320/leistungsangebote/mechatronik/

ROBOCLIMBER: Control System Architecture

S. Nabulsi, H. Montes, and M. Armada

Control Department, Industrial Automation Institute, Madrid, Spain
snabulsa@iai.csic.es

Summary. The paper shows the complete system architecture of a walking and climbing robot developed to make slope consolidation, where heavy duty equipment is employed for drilling up to 20 m holes. ROBOCLIMBER a quadruped robot whose development in founded under a Growth/Craft RTD project, and must operate on harsh conditions in order to make positioning and have to be capable of making the drilling tasks from a remote and safe place. The system architecture used to coordinate the manoeuvrability of the positioning, the drilling process, and the remote operation of the machine are explained.

1 Introduction

The development of walking and climbing robots for a broad spectrum of applications has been recently reported on literature. In particular the application of climbing machines to construction industries is a very active area of research. ROBOCLIMBER is a quadruped walking and climbing robot developed to make automated operations for the positioning and the drilling of rocky slopes or vertical walls, with the purpose of firming up the terrain where landslides can affect different kind of infrastructures such as railroads, buildings, houses or roads aside mountains [1–3].

The whole concept of the robot is based on a mechanical structure powered by a hydraulic oil pump, which has all the equipment on board including the drilling device, the oil pump, the electric and electronic control and the complete set of actuators. This structure is supported by 4 legs were a cylindrical configuration was chosen. Each of the legs is able to support a total weight of 1500 Kg and they are able to carry all the equipment on board that which weighs more than 3000 Kg (see Fig. 1).

To develop a stable control, capable of running a big robot with high reasonable security, a proper solution was to use hydraulic powered cylinders as actuators rather than electrical motors. The main advantages of the hydraulic servo-actuators are the high pressure they can support, transmitting

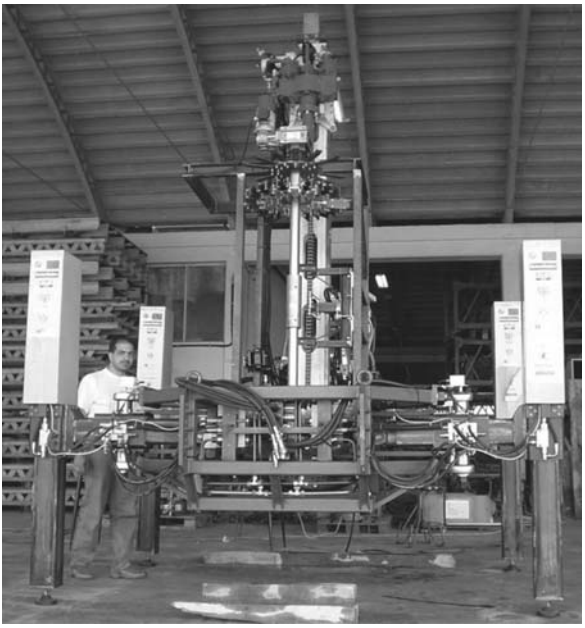


Fig. 1. ROBOCLIMBER four legged walking and climbing robot

movements with high payloads, also they can be controlled with different types of feedbacks like pressure, flow, position and/or force sensors; finally and because this application does not need high velocity, the hydraulic actuators develop soft movements, this property can be very useful in the generation of gaits where the synchronicity of the different actuators is critical.

The robot is made up of two main systems, the first one is the legged system which, as well, is made up of the four legs each of them with a 3 DOF cylindrical configuration, where the movements are developed by hydraulic servo cylinders. The final task made by this system is the generation of walking and climbing gaits necessary to move the robot to a specified position on a flat terrain or on a slope, were is possible to find irregular terrain.

For the drilling process, the second principal system of the robot, the generation of sequences of different movements are needed for the semi-autonomous drilling tasks. These set of tasks has to be supervised by an expert on the area, because there is not enough information to develop an expert system with the new type of drilling process used in this procedure. The drilling system is made of a hydraulic motor for the rotation of the drill, another hydraulic motor used to pierce the surface, and a set of jacks and electric motors to supply the extensions to achieve up to 20 m holes.

In the real application of the machine the two systems will never work at the same time. After the positioning of the robot is made the legs have to be in a static position while the drilling tasks are made. Is very difficult to make

a simultaneous control, because the hydraulic pump is not powerful enough to feed both of the systems. A single control unit set made of a CPU, control cards, data acquisition cards, electric power supplies and power conversion is used for the entire system. The control architecture must deal with all the data of in order to get the necessary information about the behaviour machine processes. For this reason QNX 6.2 a real-time based operation system is used to make the reading and the data processing of all sensors information keeping the capacity and the accuracy of controlling the different actuators to react in real time. The whole system will be detailed and the results of some experimental practices on gait generations will be analysed.

2 Joint Movement

As said before, the legs are powered by hydraulic jacks where in total the complete robot has to control 12 joints. Each of the legs configured with one rotational and two prismatic joints have to be controlled for speed and for position, that is why for each joint there is a position transducer, were in the case of a the rotation joint is used a regular encoder to measure the angle of the leg respect the axis (see Fig. 2(a)) and, instead, for the prismatic joints a linear encoder is used (see Fig. 2(b)).

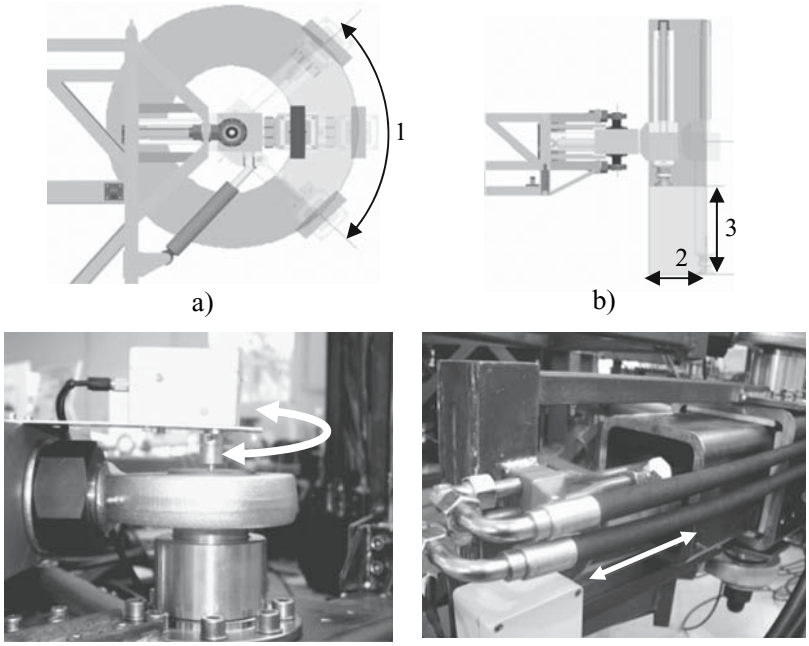


Fig. 2. Legs degrees of freedom (DOF), (**a**) Rotational joint; and (**b**) Prismatic joints

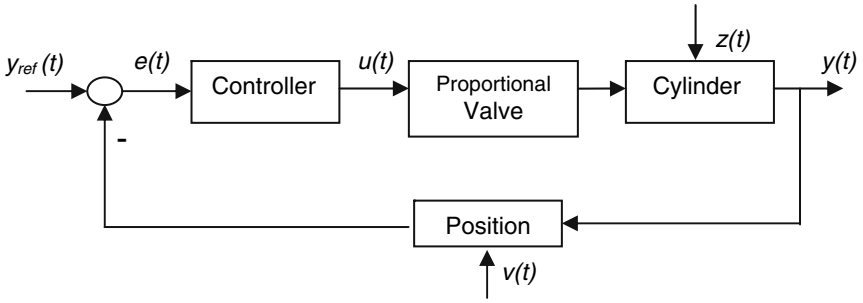


Fig. 3. Joint Movement: Block diagram of the servo-hydraulic system with position feedback for each joint

According to the requirements of the system a proportional valve controls the fluid direction and flow from a hydraulic power supply to an actuation device, in this case, a double-acting cylinder that allow the hydraulic force to be applied in both directions where the velocity and the position can be monitored to close the control loop [4, 5] (see Fig. 3).

Where the hydraulic pump is part of an auxiliary unit that can be considered as a source of the hydraulic power. The controller as indicated is the central CPU that controls a power drive unit that in addition generates a PWM signal for the position of the aperture of flow in the proportional valve on the desired direction of the cylinder [6].

When simultaneous movements of the legs' joints are made the control loop must be closed by a velocity feedback, waiting the computed time that each joint need to reach a calculated position.

3 Gait Generation System

The hierarchical structure of the control system for the gait generation has been defined in three main elements [7]:

– Level A: navigation planning
– Level B: gait control
– Level C: basic motion regulation

The positioning of the robot is not autonomous; an operator should give the necessary commands and decide in which direction to move and where to stop according to the position feedback generated by the control unit. Predefined gaits are programmed on the CPU; these gaits are enough complex for the positioning of the robot, on flat or on a sloped terrain, where to maintain a position parallel to the surface is critical for the stability and safety of the machine.

The climbing and walking gaits have different movements, so depending on the application the user must decide between the two options. For the walking

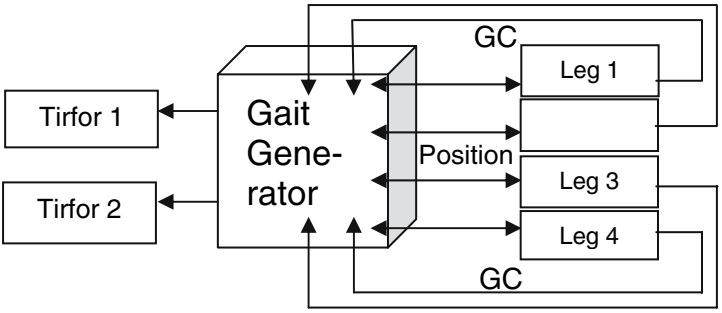


Fig. 4. Gait generation diagram

process a two face discontinuous gait is applied and for the climbing process a one phase discontinuous gait is generated [4]. Once the choice is made the robot will generate the gait according to the commands received. The control system is designed to generate from the kinematics the coordination of the joints to move the robot.

To achieve parallelism on an irregular terrain two types of ground contact (GC) control have been developed. The first one is an on/off detection that when the foot touches the ground a mechanism provokes the activation of a proximity sensor and stops. The second option is to sense the force of the leg touching the terrain by a sensor arrangement made with strain gauges on the feet [8]. The second option gives better results in the sense of a better distribution of the forces in each of the legs, but implies much more computational power (see Fig. 4).

For climbing two hydraulic mechanisms called Tirfor[a], attached to the front edges of the robot, helps to pull the machine up or down through a steel rope attached from the top of the mountain. Each of the Tirfors is powered by an on/off electro valve that turns on when the body moves and the feet are touching the ground. The climbing speed of the Tirfors and the movement of the body of the robot must be simultaneous, so the speed of the Tirfor must be determined before starting the climbing procedures and then the speed of the body can be set.

4 Drilling Process Controller

Some of the drilling processes are on an open loop control. The semi-autonomous drilling operation consist on a hydraulic motor (see Fig. 5(1)) that introduces a hammer drill (2) in the rocky surface. To introduce it deeper an extension should be placed, so the motor has to be detached from the hammer, and then an extension (3) is placed between motor and the hammer. This process must keep going until a certain depth is reached.

[a]TRACTEL Group, "http://www.tirfor.co.uk"

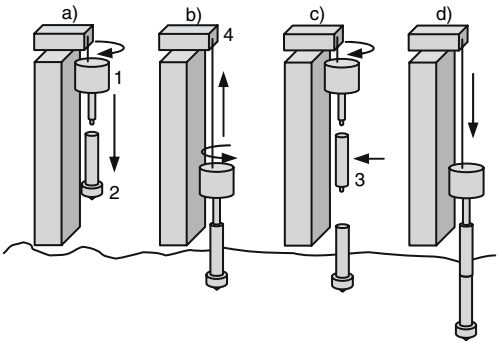


Fig. 5. Basic steps and components of the Drilling Process

While drilling some aspects have to be considered, off course, the deepness and the velocity of the thrust (a.) measured with an encoder (4), the force of the rotation of the drill known by a pressure sensor on the hydraulic circuit, and the thrust or the pressure made to pierce the surface also monitorized by another sensor. This information is considered as important in the process because, for example, if the hammer's shows any trouble while drilling, it can be a sing that an unwanted material is blocking the system inside the hole generating high pressure on the hydraulics; like this, there are other circumstances that can turn out during the rock perforation.

The operator can select the speed of rotation and the thrust manually, but if any of the sensors detects unusual high levels of pressure the drilling process must stop to prevent possible damage of the machine.

5 Control System Architecture and HMI (Human Machine Interface)

The control system for ROBOCLIMBER has been separated in two main subsystems: a) one on-board the legged platform and drilling systems; b) the other on the supervisor station and command control.

Both for generating gaits and for the working tools is required the use of many sensors and the coordination of all actions in real-time. So the control system has to process information from the sensors in order to maintain the system in the correct position and attitude while performing drilling and consolidating work. The main control systems are based on a control card designed by de Control Department of the Institute of Industrial Automatics (Spain).

PID with position feedback and with PWM (Pulse Width Modulation) output is the main control used by these cards, this type of controller is appropriate for many kinds of actuators like electric motors. In this case the

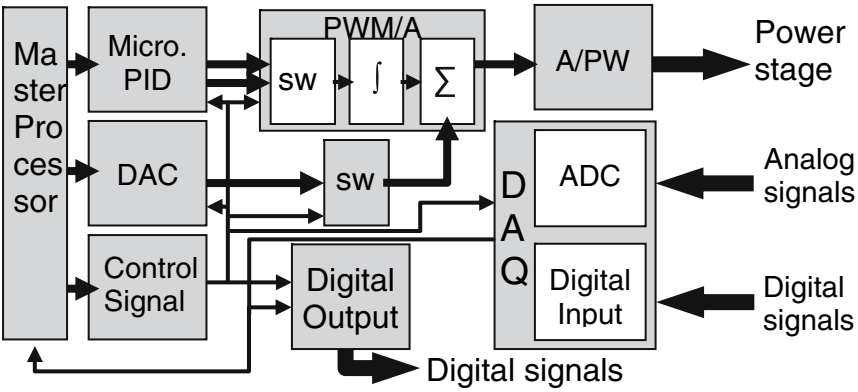


Fig. 6. Control Card design. Control Department of the Institute of Industrial Automatics (Spain)

PWM output is transformed in an analog signal of ±10 Volts necessary to control the hydraulic power units (see Fig. 6).

This kind of electronic cards have other advantages useful for the type of control needed for robots. The first advantage is that with only one command the card is able to control various actuators simultaneously and with autonomy; this properties reduces significantly computational power. Another advantage is the implementation of digital and analog inputs and outputs; in this case, this property is very useful for the sensoring process and for the control external devices.

Computational power is needed to compose the system controller for all the subsystem on a real-time application. Programmed on C language and running on QNX platform the application control running on an 400 MHz industrial CPU, is enough powerful to complete the sequences of the gait generation system, the drilling process through the control cards, and the monitoring system (see Fig. 7).

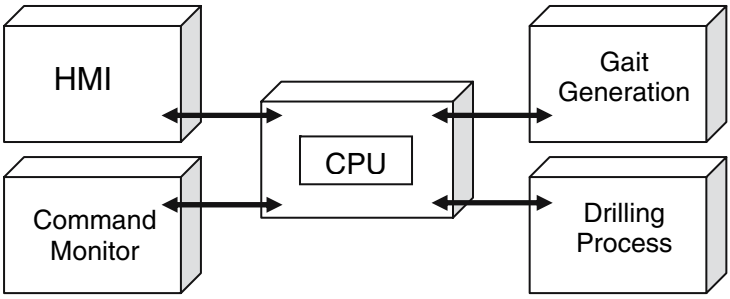
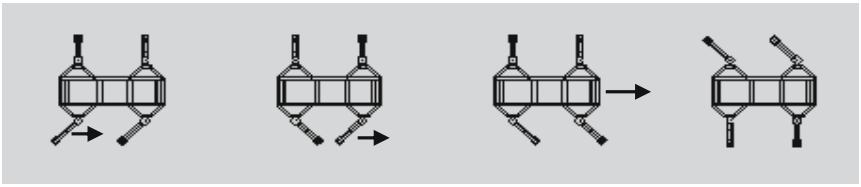
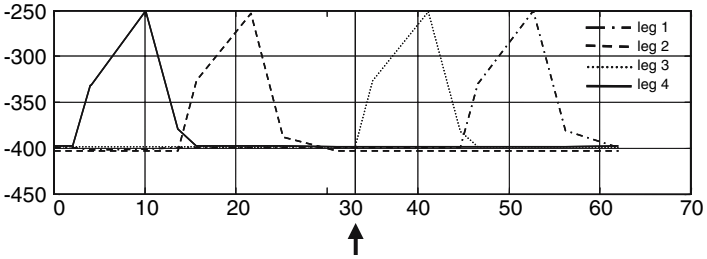


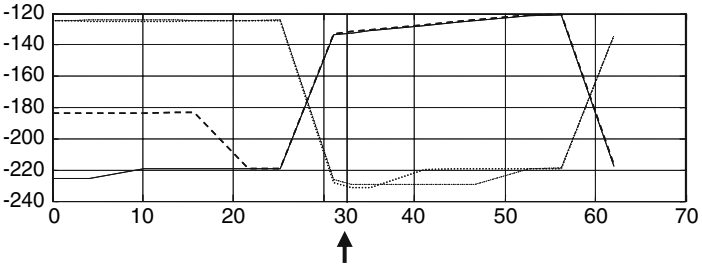
Fig. 7. Basic Control System Architecture



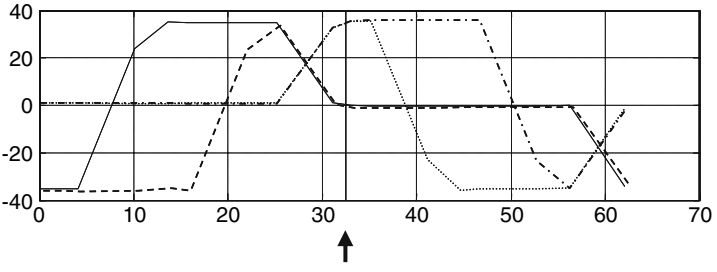
a. Walking gait movements (half cycle)



b. Vertical joints (mm)



c. Radial joints (mm)



d. Rotation joints (Degrees)

Fig. 8. Position of the joints in the Walking gait sequence

The monitoring system can be selected from two choices, the first one it to introduce the commands directly from a keyboard and the second to work with a HMI via TCP connection protocol. The second option is used to make the climbing process and drilling operations from a remote and safe position of the user. Both of the systems works the same way, a global command is selected by the operator and the CPU makes the necessary actions to make the to control the machine, while sending the different status variables like position, warnings or emergency signals as feedback to the operator.

6 Experimental Results

In the two-phase discontinuous gait generation for walking the control unit of Roboclimber must make many simultaneous actions (see Fig. 8).

The Fig. a) shows halve of a complete walking gait cycle. The arrow on the graphics b) to d) shows where the first halve of the complete cycle finishes. First the two legs of one of the sides should be placed on a position forward to the direction of the movement (a. and b.), followed by the movement of the body (c), made possible rotating of the four legs at the same time, meaning the coordination between eight joints controlled with velocity feedback. The graphics shows the position of the every joint in each legs.

7 Conclusions

The paper explains the control architecture of a four-legged robot intended for the automation of the firming-up of rocky slopes and walls, to grant safeguard of peopled areas, highways, private residences or public sites. The prospected solution looks after a goal-oriented robotic equipment, Roboclimber, for tethered wall climbing and endowed with devices for churn drill, boring and cast-in-situ piling. The architecture of the developed control system for the gait management and legs-ropes co-ordination was introduced, with focus on its two-level structure, which leaves the possibility of autonomous climbing or manual driving through a remote console.

Acknowledgements

ROBCLIMBER project is funded by the EC under Contract N°: G1ST-CT-2002-50160. The project partnership is as follows: ICOP S.p.a., Space Applications Services (SAS), Otto Natter Prazisionenmechanik GmbH, Comacchio SRL, Te.Ve. Sas di Zannini Roberto & Co. (TEVE), MACLYSA, D'Appolonia S.p.a., University of Genoa-PMAR Laboratory, and CSIC-IAI.

References

1. Armada M, Molfino RM (2002) Improving Working Conditions and Safety for Landslide Consolidation and Monitoring. In: Workshop on the role of CLAWAR in education, training, working conditions and safety, on CD-Rom, Madrid
2. Anthoine P, Armada M, Carosio S, Comacchio P, Cepolina F, González P, Klopf T, Martin F, Michelini RC, Molfino RM, Nabulsi S, Razzoli RP, Rizzi E, Steinicke L, Zannini R, Zoppi M (2003) ROBOCLIMBER. In: ASER03, 1st International Workshop on Advances in Service Robotics, March 13–15, Bardolino, Italy
3. Competitive and Sustainable Growth Programme CRAF-1999-70796 (2002) Sixth month report, Basiliano, Udine
4. Nabulsi S, Armada M (2004) Climbing strategies for remote maneuverability of ROBOCLIMBER. In: 35th Inteernational Symposium on Robotics, 23–26 March, Paris-Nord Villepinte, France
5. Nabulsi S, Armada M, González de Santos P (2003) Control Architecture for a Four-Legged Hydraulically Actuated Robot. Measurement and Control in Robotics (ISMCR 2003), pp. 291–295, Madrid
6. Jelali M, Kroll A (2003) Hydrulic Servo-systems: Modelling, Identification and Control. Springer, London
7. Hirose S (1984) A Study design and Control of a Quadruped Walking Vehicle. The International Journal of Robotics Research, 31(2): 113–133
8. Montes H, Pedraza L, Aramda M, Akinfiev T, Caballero R (2004) Adding extra sensitivity to the smart non_linear actuator using sensor fusion. Industrial Robot: An International Journal, 31(2): 179–188

Navigation of Walking Robots: Localisation by Odometry

B. Gaßmann, J.M. Zöllner, and R. Dillmann

Forschungszentrum Informatik, Haid-und-Neu-Str. 10–14, 76131 Karlsruhe, Germany
{gassmann,zoellner}@fzi.de,
dillmann@ira.uka.de

Proper navigation of walking machines in unstructured terrain requires the knowledge of the spatial position and orientation of the robot. There are many approaches for localisation of mobile robots in outdoor environment, but their application to walking robots is rather rare. In particular, middle sized robots like LAURON III don't provide the possibility to carry large or heavy sensors. Due to many degrees of freedom of walking robots the localisation task becomes a even more complex challenge. This paper discusses the problem, presents a method of resolution and describes the first steps towards a localisation system for the six-legged walking robot LAURON III.

1 Introduction

Navigation in unstructured terrain is a difficult challenge. Compared to structured environments, the geometry of unstructured areas cannot be described in detail, because there is no complete a-priori knowledge available on the surface and the ground conditions on which the walking machine has to operate. Legged machines are very suitable for navigating in this sort of terrain due to the great flexibility in their motion and their ability to cope with very uneven environment. In the last decades an enormous technical progress could be observed in the fields of walking robots in all their variation all over the world. The main focus of outdoor walking in unstructured and rough terrain lies on the build up of the mechanical systems and the basic control algorithms including gait generation, leg coordination and basic adaption to the terrain (e.g. Genghis [3], Tarry [4]). Recent research in the area of two legged walking is mainly restricted to structured and well-defined environments. Higher level of control which enable autonomy to the greatest possible extend in terms of navigation in unstructured terrain could hardly be found. In the context of this paper navigation covers the fields of:

Localisation: determine the spatial position and orientation of the robot

Map building build up of a geometrical and/or topological model of the environment

Planning and action selection plan the global/local path and detailed footholds of the robot and select general parameters for the walking movements (e.g. target values relating to the body posture, the gait or the step height)

Because the localisation task forms the basis of the map building process and thus as well of the planning and the action selection techniques, the overall performance of the navigation system depends on the accuracy of the robot localisation.

2 Previous Work

Since only few walking robots give attention to the navigation task there are only few published articles concerning the navigation of walking robots. The early huge walking robots like the Adaptive Suspension Vehicle (ASV) [2], the Ambler [10] and the Dante II [1] were able to carry a huge and heavy 3D distance laser for supporting the build up of a local 2D elevation map of the environment. For merging the local 2D elevation maps the position of the ASV was determined by the use of an inertial navigation system (INS). For this purpose in [7] an approach for the dead reckoning navigation for the Ambler was introduced.

Recent projects on middle-sized walking robots start to cover navigation aspects again. The dynamical hexapod robot RHex [8] follows a high-contrast line on the ground by the use of an INS and a camera system. The Scorpion project [9] aims to navigate an eight legged robot through a 25 miles desert environment. The four legged JROB-2 [6] is a research platform for the investigation of image processing algorithms on a walking robot for object tracking and obstacle avoidance.

3 Test Platform Lauron III

Lauron III is a stick insect like walking machine built as a universal platform for locomotion in rough terrain It consists of a main body, a head and six equal legs, each providing three degrees of freedom ($\alpha \pm 80$, $\beta \pm 100$, $\gamma \pm 140$). With all components mounted on, the weight of Lauron III is about 18 kg. It has a length and width between the footsteps of about 70 cm and a maximal payload of 10 kg. The maximum distance between body and ground is 36 cm, the maximum velocity is given by $0.3\,\mathrm{ms}^{-1}$. Lauron III matches the requirements for autonomy, whereas the accumulators last for about 45 min (average power consumption 80 Watt). Its joint angles are measured by optical encoders, the load of the motors could be determined by the measured currents. Lauron III is equipped with foot force sensors mounted on the lower leg, allowing the

measurement of all three axial force components. This construction enables the determination of the contact forces including direction and quantity. The main body carries an inertial sensor box delivering information about the angular velocity and the axial acceleration, each in three dimensions. Furthermore, a laser distance sensors (range: 20–500 cm, resolution: 0.5–3 cm) is fixed in the middle of the head for scanning the near environment.

4 General Approach

Till this day 3D distance laser systems remain not suitable for small or middle-sized walking robots like LAURON III. So map building must be possible on the basis of successive sparse sensory information [5]. Therefore the robot localisation task must be performed with high accuracy to guarantee a consistent and thus a useable model of the environment. Because a single localisation method is not expected to provide the require precision in the rough outdoor environment, the combination of different localisation systems has to be investigated.

The first relative measuring system to be examined is the calculation of the robot odometry on base of the joint encoders of the legs. This system provides the general trend of the robot movement. The realization of the odometry calculation on the walking robot LAURON III is presented in the following sections. The use of an inertial navigation system including magnetic field sensors provides further information about the orientation of the robot. In addition it enables the detection of disturbances like slipping of the whole robot which could not be detected by the odometry system. To compensate the drift of the relative measuring systems a DGPS system is introduced. This enables the robot to obtain global position information if the mandatory satellites are available. In case the DGPS system is unsuitable the detection and tracking of natural landmarks with the help of a stereo camera system should be introduced.

5 Odometry System for Walking Robots

The use of dead reckoning as base of the localisation task is very common in mobile robotics (see Fig. 1 (left)). In contrary to wheeled robots, where dead reckoning implementation could be realized via the odometry of the wheel encoders, walking machines don't provide continuous contact to the ground. Each leg of a walking robot performs a stepping cycle. While contacting the ground the leg supports the main body (support phase). It is pushed backwards to propel the main body forward until a posterior extreme position (PEP) in respect of the joint configuration is reached. Then the leg is lifted off the ground and swung to an anterior extreme position (AEP) in the front again (swing phase) for starting the next support phase. Following that matter

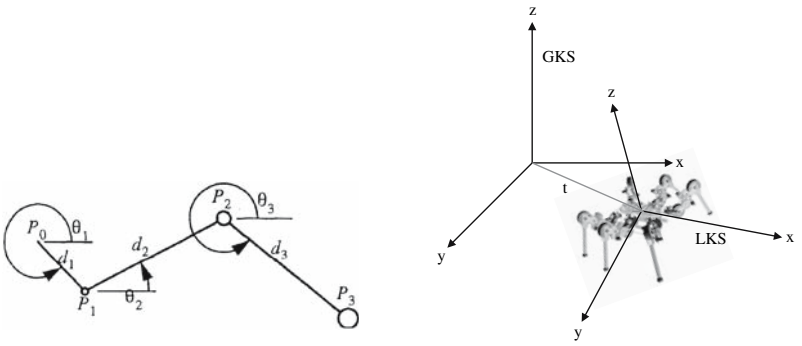


Fig. 1. (*Left*) Dead reckoning course of a robot starting in $P0$, travelling along the path segments (d_i, θ_i). The new positions P_i can be calculated from the course and distance travelled. (*Right*) The coordinate systems for calculating the odometry. The body frame is assigned to the robot. The world frame is assigned to a frame of reference

of fact only the movement of the supporting legs propel the rigid main body forward. Roston and Krotkov [7] introduced an approach for calculating the odometry of the Ambler walking robot. At this they take also in account that a leg could slip if the robot walks on uneven terrain. After the identification of slipping legs the movement of all remaining supporting legs are used as input for the calculation of the rigid body transformation (the matrix R and the vector T) which satisfies:

$$P_{w,j} = RP_{b,j} + T \tag{1}$$

where $P_{w,j}$ is the position of the foot j in the world frame (w) which remains constant in the support phase for a non slipping leg. $P_{b,j}$ is the position of foot j in the body frame (b) which can be calculated from the leg kinematics (see Fig. 1 (right)). Following [7] the solution of that equation could be obtained by minimising the square error $q(R, T)$, where:

$$q(R,T) = \sum_j w_j e_j^T e_j \tag{2}$$

with

$$e = P_{w,j} - RP_{b,j} - T \tag{3}$$

where w_j are weights with

$$w_j = \begin{cases} 1, & \text{foot slips or swings} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Unlike Ambler which moved only one leg at a particular time LAURON III in general swings multiple legs simultaneously. In addition the walking speed of LAURON III is in respect to the robot dimension much higher. This leads often

to partly slipping feet in support phase, so that the classification of *slipping feet* and *not slipping feet* is not obvious any more. When walking in rough terrain the feet could also collide with obstacles which also could affect the position of the main body by pushing it aside. All these disturbances strongly influence the preciseness of the odometry calculation.

5.1 Generation of the Fuzzy Weights

The main problem to be solved for the odometry calculation of LAURON III is the determination of the influence of each single leg on the movement of the main body. This is solved by introduction of a fuzzy reasoning process which generates a fine granular weight factor w_j for each leg. Thereby a weight of $w_j = 0$ represents a leg j that has no influence on the robot movement at all. A weight of $w_k = 1$ corresponds to a leg that has strong influence on the movement of the main body. In contrary to (4) the fuzzy reasoning allows weights in the whole range of $[0, 1]$.

At first it has to be defined what leg states are of interest and how these states could be quantified. The main leg states that could be identified are:

Ground Contact: A swinging leg which does not touch the ground at all should not influence the position of the main body in the world frame. On the other hand a leg with strong contact to the floor pushes the body forward.

Slipping: Intense slipping legs could also be treated as swinging legs without influence on the odometry of the robot. But legs that slip only slightly have to be taken in account partly.

Collision: A light collision of a leg with an obstacle won't affect the robot position at all, whereas a hard collision with a fixed object could push the main body aside.

Now the question arises why fuzzy reasoning is used in this approach. In most cases the above described leg states could not be retrieved from a single sensor value. They arise from the interpretation of different sensor information whereas their consistency could not be guaranteed. Furthermore there are smooth transitions within the states itself. The concept of fuzzy reasoning suits for solving these constraints.

Figure 2 sketches the proposed hierarchy for determination of the weights w_j for each leg of the robot. The sensor measurements *Contact Forces*, *Joint Angles* and *Motor Currents* build the basis of the system. In the following the fuzzy reasoning mechanism is described roughly by exemplary listing of the inputs, the outputs and examples for the rules used. On the first level the main leg states are computed:

Ground Contact

Input	absolute and relative contact forces of the foot: $F_{abs,j}$, $F_{rel,j}$
Output	weighting of ground contact $g_j \in [0, 1]$
Rules	IF $F_{abs,j}$ = high THEN g_j = high
	IF $F_{rel,j}$ = high THEN g_j = high

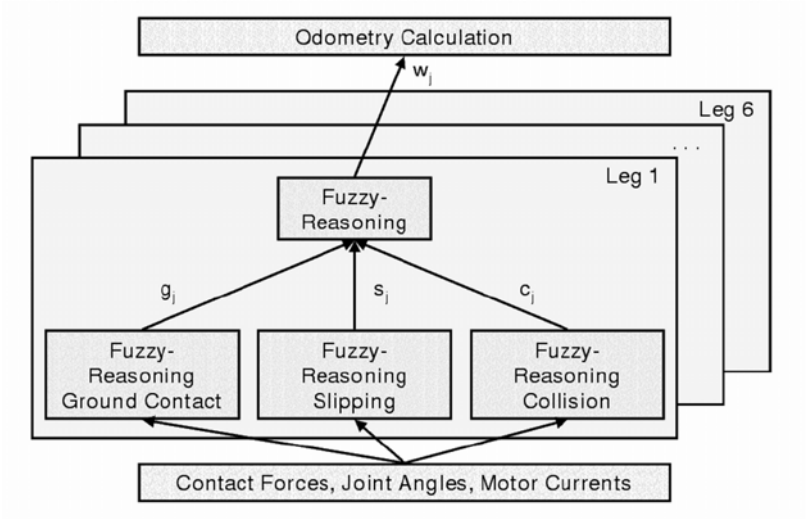


Fig. 2. On base of the measurement of *Contact Forces*, *Joint Angles* and *Motor Currents* the intensity of the leg states *Ground Contact* g_j, *Slipping* s_j and *Collision* c_j are evaluated. An overall fuzzy reasoning process fuses these into the leg weight w_j which is passed to the *odometry calculation*.

Slipping

Input	absolute contact forces $F_{abs,j}$, angular velocities ω_j, body frame foot position $P_{b,j}$
Output	weighting of slipping $s_j \in [0, 1]$
Rules	IF $F_{abs,j}(Z) =$ high THEN $s_j =$ low
	IF $F_{abs,j}(Z) =$ low AND $\omega_j =$ high THEN $s_j =$ high

Collision

Input	absolute contact forces $F_{abs,j}$, angular velocities ω_j, motor currents I_j
Output	weighting of collision $c_j \in [0, 1]$
Rules	IF $\omega_j =$ small AND $F_{abs,j} =$ high THEN
	IF $\omega_j =$ small AND $I_j =$ high THEN

The second level then fuses the output of these main leg states into the overall weight w_j characterising the influence of the leg movement:

Overall Weight

Input	ground contact g_j, slipping s_j, collision c_j
Output	overall weighting of the leg $w_j \in [0, 1]$
Rules	IF $g_j =$ low AND $s_j =$ high THEN $w_j =$ low
	IF $g_j =$ high AND $s_j =$ low THEN $w_j =$ high
	IF $g_j =$ high AND $c_j =$ low THEN $w_j =$ high

Table 1. Summary of the walking experiments on different terrain

Experimental Setup	Average Error
Flat ground straight slow (25 m)	5%
Flat ground straight fast (25 m)	10%
Flat ground turning (360°)	10%
Slope	10%
Gravel	12.5%
Deep sand	25%

The overall weight w_j is used as input for the the odometry calculation according to (2).

6 Test Results

To analyse the accuracy of the described method, several experiments have been done. Table 1 shows the summary of these experiments. To get a first impression of the odometry system LAURON III had to walk on flat ground. On straight slow walking for 25 m an average error of 5% could be observed. If the speed of the robot was increased for the same experimental setup this error grew up to 10% because of an increase of the disturbances, especially the irregular ground contact forces. Turning on flat ground showed also an error of about 10% which could be explained by the tensions between the feet. An error of 10% was also measured on walking up slopes (see Fig. 3 (right)) whereas an increase of slipping feet could be noticed. The more the ground became softer, the more the odometry error raised from about 12.5% on gravel (see Fig. 3 (left)) up to 25% on deep sand. This effect is up to the compliance of the terrain. The material beneath the feet compresses. Therefore the legs perform moves in the body frame while at the same time the measured forces in the feet are still high. This leads to high weights for the odometry calculation of these feet although they are slipping on or better into the ground.

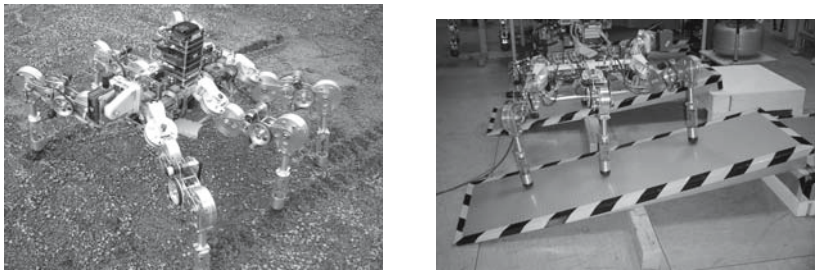


Fig. 3. (*Left*) LAURON III walking on gravel (*right*) LAURON III walking on a slope

7 Conclusion and Outlook

The implemented dead reckoning system based on the odometry calculation for the walking robot LAURON III is the first step towards a robust localisation system. Though the errors of the system vary between 5% and 25% the odometry gives a rough approximation of the travelled distance. By increasing the accuracy of the sensor system of LAURON III a more precise evaluation of the leg states and therefore a better performance of the overall calculation could be expected. Nevertheless it is strongly recommended to focus on additional localisation systems. Currently the coupling of the odometry with an inertial sensor system and a GPS receiver is under investigation.

References

1. John E. Bares and David S. Wettergreen. Dante ii: Technical description, results, and lessons learned. *The International Journal of Robotics Research*, 18(7):621–649, July 1999.
2. Thomas E. Bihari, Thomas M. Walliser, and Mark R. Patterson. Controlling the adaptive suspension vehicle. *Computer*, 22(6):59–65, June 1989.
3. Enric Celaya and Josep M. Porta. A control structure for the locomotion of a legged robot on difficult terrain. *IEEE Robotics & Automation Magazine*, 5(2):43–51, June 1998.
4. M. Frik, M. Guddat, M. Karatas, and C. D. Losch. A novel approach to autonomous control of walking machines. In *Proceedings of the 3th International Conference on Climbing and Walking Robots*, pp. 333–342, Portsmouth, UK, 1999.
5. B. Gaßmann, L. Frommberger, R. Dillmann, and K. Berns. Improving the walking behaviour of a legged robot using a three-dimensional environment model. In *Proceedings of the 6th International Conference on Climbing and Walking Robots*, pp. 319–326, Catania, Italy, September 2003.
6. Atsushi Konno, Noriyoshi Kato, Yusuke Mitsuya, and Masaru Uchiyama. Design and development of the quadrupedal research platform jrob-2. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 1056–1061, Washington, DC, USA, May 2002. IEEE.
7. Gerald P. Roston and Eric P. Krotkov. Dead reckoning navigation for walking robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 607–612, Raleigh, NC, USA, July 1992. IEEE/RSJ.
8. Sarjoun Skaff, George Kantor, David Maiwand, and Alfred A. Rizzi. Inertial navigation and visual line following for a dynamical hexapod robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pp. 1808–1813, Las Vegas, NV, USA, October 2003.
9. D. Spenneberger and F. Kirchner. Scorpion: A biomimetic walking robot. In *Robotik 2002*, volume 1679 of *VDI-Bericht*, pp. 677–682, 2002.
10. David Wettergreen, Hans Thomas, and Chuck Thorpe. Planning strategies for the ambler walking robot. In *Proceedings of IEEE International Conference on Systems Engineering*, pp. 198–203, August 1990.

Towards Penetration-based Clawed Climbing

William R. Provancher[1], Jonathan E. Clark[1], Bill Geisler[2], and Mark R. Cutkosky[1]

[1] Department of Mechanical Engineering, Stanford University, Stanford, CA 94305
wil@cdr.stanford.edu
[2] Department of Biology, Lewis and Clark College, Portland, OR 97129

1 Introduction

Despite significant research focused on running robots, very little progress has been made towards legged robots that are capable of climbing in natural environments. Unlike their running counterparts, climbing robots must generate hand or foot holds capable of pulling them toward the substrate. The majority of efforts to develop climbing robots have been for urban settings with smooth glass or metal surfaces where suction and magnetic approaches to generating adhesion are possible. Some examples of robots that have used a suction based approach include [8, 10, 15]; some magnetic based climbers include [2, 13]. A few robots have also addressed climbing on rough rock surfaces, employing strong grips capable of sustaining tensile and shear loads [3, 4]. This paper describes efforts towards the development of a penetration-based clawed climbing robot capable of climbing on rough or smooth inclines.

1.1 Climbing in Natural Environments

Developing successful climbing robots for unstructured terrain or natural environments requires developing methods of generating adhesion, selecting a suitable sensor suite, developing locomotion strategies, and computing desired robot trajectories. In this paper, we focus on the first of these challenges.

As a starting point for selecting a mode of adhesion suitable for unstructured terrain, we look to the numerous examples found in nature. The primary modes of adhesion used by animals include wet and dry adhesion as well as claws or barbs. Some smaller animals, including Asian weaver ants, palmetto beetles, and tree frogs use wet-adhesion to stick to branches or trunks [6, 7]. However, the forces that can be generated from surface tension and capillary action in wet adhesion are not great enough to support large animals and may not be sufficient for small to medium scale robots.

Geckos, on the other hand, use dry-adhesion to stick to virtually any surface, no matter how slippery or shear. Geckos implement their van der

Waals-based adhesion through millions of tiny hairs (setae) on their toes [1]. Recent experiments have shown that this mode of adhesion is capable of generating adhesive forces suitable for a small climbing robot [12]. The great potential and multitude of applications of this adhesive mechanism have motivated researchers to synthesize artificial dry adhesives [12]. However, practical artificial dry adhesives are not yet available.

Another approach to climbing, used by many larger mammals–including humans–utilizes cracks, ledges, and other types of hand holds to cling to a wall. Recently work has been done on developing robots [4] that use hand-hold and move like human rock climbers. These hand-hold based approaches, however, cannot be implemented in the absence of such surface irregularities. For a robot designed for rough terrain climbing, smooth impenetrable surfaces present the greatest challenge. On the other hand, a robot with claws designed to penetrate soft surfaces or engage an asperity on a rough surface does not need to rely on fortuitous hand or footholds to scale a wall. In fact, Mahendra [11] has shown that without claws even a gecko's ability to adhere to very rough surfaces is compromised.

1.2 Claw-Based Climbing

For climbing on both rough and smooth penetrable surfaces, a claw-based approach is promising. Animals as small as insects and as large as bears use spines or claws when climbing. For example, cockroaches can climb an impressive range of materials by using their claws in conjunction with sticky metatarsal pads. However, in order to use claws effectively on a climbing robot, we need to develop engineering models of claw behavior, supported by empirical results.

Dai, et al. have used a simple model for clawed adhesion to predict frictional forces based on relative claw tip radius and surface roughness [5]. However, this model only considers shear forces generated when the insect foot interacts with a non-penetrable textured surface. For climbing it is also necessary to consider adhesive normal forces.

Many biologist have also investigated evolutionary adaptations specific to climbing. Zani [14] has looked specifically at lizard claws and toe morphology for climbing, showing improved clinging performance of shorter, stiffer claws on rough surfaces.

Recent research focused on observing cockroaches while climbing (unpublished experiments of D. Goldman and R. Full, UC-Berkeley) has shown that the roach's ability to climb shear surfaces is clearly effected by surface roughness. It appears that they rely on small irregularities in the substrate as footholds and therefore as surface roughness decreases, their ability to climb degrades.

Consequently, for smooth, soft surfaces the best approach for a robotic climber may be to use claws to penetrate the substrate. To date, the primary

work done on penetration and attachment to smooth surfaces comes from engineering, specifically the nail and staple industry [9].

Whether penetration or surface irregularities are utilized, a claw-based climbing robot has a number of challenges to overcome. Some of these include determining proper path generation, angle of attachment, and retraction angle. Other issues include dealing with ankle rotation during stance and keeping the claws sharp. This study takes the first step towards developing a viable claw-based climber by examining how claw tip geometry and insertion angle affect the ability of a claw to penetrate and grip a smooth penetrable surface.

2 Experimental Setup and Testing

To begin to answer these questions, we performed tests to determine how the geometry of claws affects their ability to penetrate a substrate and generate shear and adhesive forces. As shown in Fig. 1 we have designed a set of experiments to test the effect of varying claw diameter (d), cone angle (ϕ), and tip radius (R), as well as claw orientation (α), approach angle (β), and detachment angle (γ) relative to the substrate normal vector.

A first set of experiments were conducted with approach (β) and retraction (γ) angles fixed at 45° and −45°, respectively. A second set of experiments

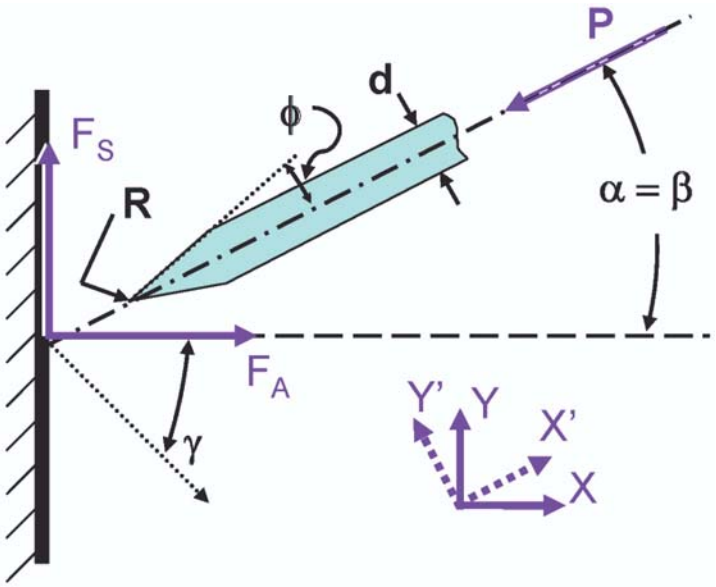


Fig. 1. Schematic showing test geometry, nomenclature, and conventions

investigated the effect of varying the approach angle (β) on adhesive forces. In both sets of experiments, the claw angle (α) and the approach angle (β) were identical. In both sets of experiments, the retraction angle, γ, of $-45°$ was chosen assuming that equal amounts of shear and adhesive forces would be required to scale a vertical wall.

Tests were performed using a Newport ESP-300 motion controller driving 850G and 850G/HS actuators on a 461XYZLM stage. In all tests, the barbs were inserted and retracted at a constant rate of 0.01 mm/s and 0.05 mm/s. Approximate preloads were prescribed by programming insertion displacement appropriately. Forces were measured in the X' and Y' axes (see Fig. 1) using a Kistler 9328A piezoelectric 3-axis force sensor and 5010B charge amplifiers. Drift in these force sensors was corrected in post-processing of data by assuming a constant drift rate and adjusting each force channel by a proportional offset, based on the no-load force readings at the end of each set of trials. Data were acquired with MacLab A/D converter on an Apple Powerbook G3 at 40 Hz.

2.1 Experimental Results

A typical claw adhesion test is shown in Fig. 2 (for barb no. 8). In each test the claw tip was driven into the substrate (soft wood) a finite displacement along the approach angle (β) and then, following a 2 second pause, was forcibly removed along the retraction path (γ). This test shows a peak preload of

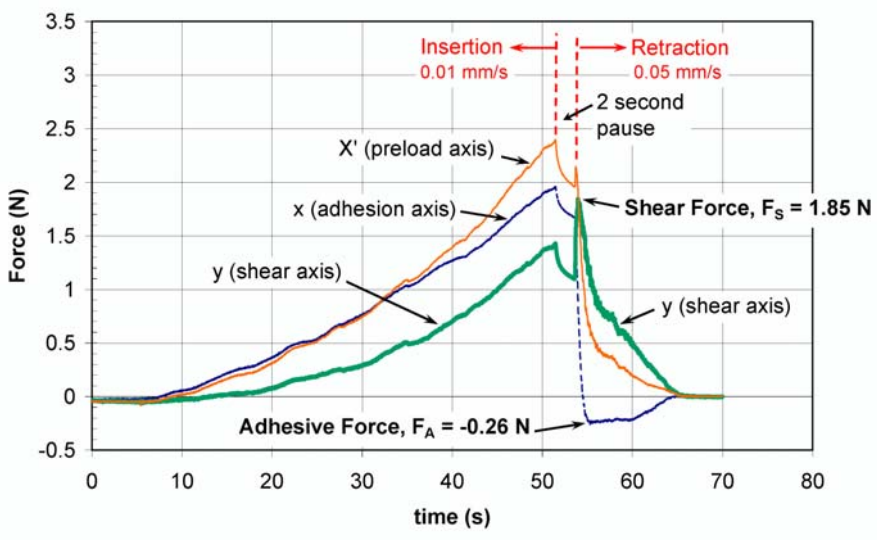


Fig. 2. A plot of a sample test run. This test was for a claw tip with diameter (d) of 1.5 mm, tip radius (R) of 30.5 µm, and cone angle of 18°

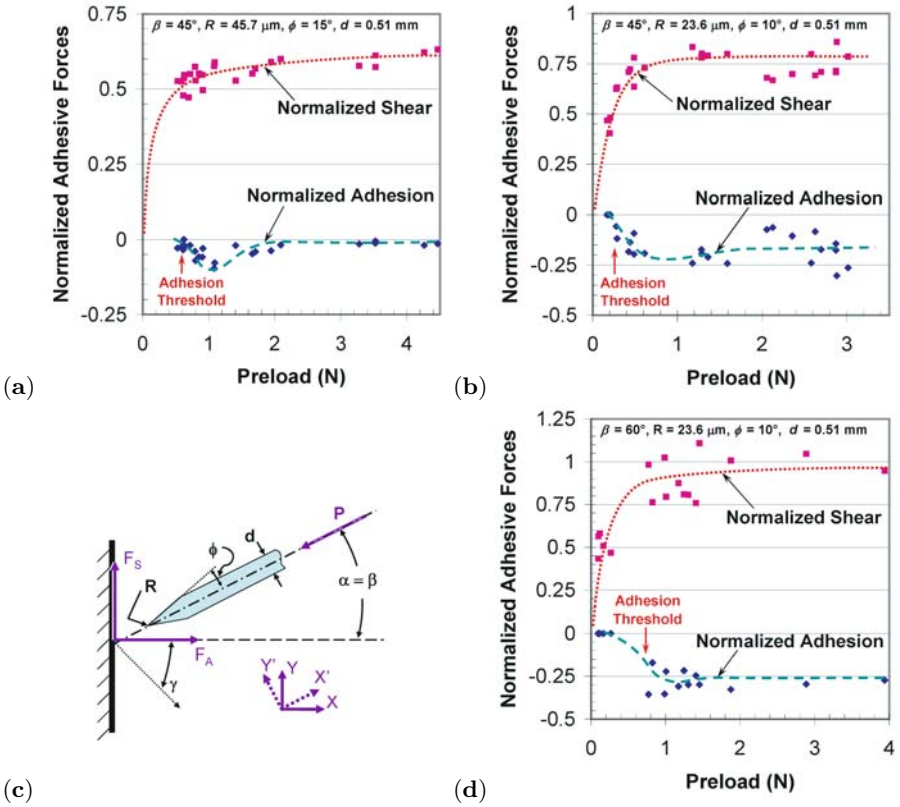


Fig. 3. A plot of results for $d = 0.51$ mm barbs. Shear and adhesion values are normalized by dividing by the applied preload, designated (F_S/P) and adhesion (F_A/P), respectively. Normalized adhesive forces are shown for (**a**) experiment 1, barb 1, (**b**) experiment 1, barb 2, and (**d**) experiment 2, barb 2, respectively. A simplified version of Fig. 1 is shown in (**c**) for reference

2.2 N on the X'-axis, and a maximum adhesive force of −0.26 N. The claw also generated a 1.85 N plowing (shear) force.

As shown in Table 1, barb geometry in the first set of experiments covered the following range: wire diameter, $d = 0.51$ to 1.50 mm diameter, cone angle, $\phi = 10$ to 21°, and tip radius, $R = 15.7$ to 121.9 μm. Data were collected in 25 trials for each barb with preloads ranging from 0.095 to 9.38 N (barb no. 8 was only tested in 21 trials). To make it easier to compare adhesive forces (on adhesion, x, and shear, y, axes) generated for various preloads, they are normalized by dividing adhesive forces by the preload for each respective trial. The average of normalized adhesive values for each set of trials is reported in Table 1 along with the standard deviation. Observe that in each case where two barbs of the same diameter, but with different tip radius, R (e.g., barbs 1 and 2), that the one with the smaller tip radius performs better. In addition,

Table 1. Results of first experiment, investigating effect of varying barb diameter (d), cone angle (ϕ), and tip radius (R). The mean of normalized shear and adhesion forces are reported, designated $\overline{(F_S/P)}$ and $\overline{(F_A/P)}$, respectively. The data were normalized by dividing by the applied preload for each respective trial. Mean values are listed with ± one standard deviation

barb no.	d (mm)	ϕ (°)	R (μm)	$\overline{(F_A/P)}$	$\overline{(F_S/P)}$
1	0.51	15	45.7	0.029 ± 0.029	0.555 ± 0.041
2	0.51	10	23.6	0.146 ± 0.085	0.700 ± 0.113
3	0.79	14	25.4	0.154 ± 0.038	0.873 ± 0.072
4	0.99	10	15.7	0.149 ± 0.050	0.746 ± 0.044
5	1.24	15	88.9	0.046 ± 0.051	0.558 ± 0.060
6	1.24	21	27.9	0.105 ± 0.045	0.733 ± 0.065
7	1.50	14	121.9	0.057 ± 0.061	0.598 ± 0.095
8	1.50	18	30.5	0.092 ± 0.068	0.662 ± 0.134

those with smaller tip radius also have a lower threshold for adhesion, as shown in a comparison of Figs. 3(a) and (b).

After establishing a good tip geometry, the goal of second experiment was to investigate what approach angle, β, is optimal for generating adhesive forces. Tests were completed for $\beta = 15$, 30, 45, 60, and 75°. 25 trials were completed for each approach angle (17 were completed for $\beta = 60°$). No trials were completed successfully for $\beta = 75°$, because the barb never penetrated the wood. Barb 2 was chosen for these experiments because of its relatively good performance in experiment 1. Results for the second set of experiments are reported in Table 2. Barbs tested at $\beta = 45°$ have the highest adhesion, while more shear force was produced at $\beta = 60°$. A comparison of Figs. 3(b) and (c) shows that $\beta = 45°$ has the lowest preload threshold for adhesion. No wear (dulling) was observed for either sets of experiments.

Table 2. Results of second experiment, investigating effect of varying the approach angle (β). These tests were conducted with the second barb from experiment 1 (d = 0.51mm, $\phi = 10°$, and $R = 23.6$ μm). As in Table 1, the mean of normalized shear and adhesion forces are reported, designated $\overline{(F_S/P)}$ and $\overline{(F_A/P)}$, respectively

β (°)	$\overline{(F_A/P)}$	$\overline{(F_S/P)}$
15	0.0393 ± 0.0437	0.2557 ± 0.1021
30	0.1092 ± 0.0771	0.5224 ± 0.1611
45	0.2328 ± 0.1570	0.6683 ± 0.1423
60	0.1981 ± 0.1399	0.7928 ± 0.2158

Results for $\beta = 75°$ experiments omitted – barb did not penetrate

3 Discussion and Application to Climbing Robots

Preliminary experiments show that tip radius (R) far outweighs the influence of cone angle (ϕ) on measured shear and normal adhesion forces. We also found that shear and adhesion forces were roughly proportional to preload. For example, with the sharpest claws, we found that we could generate shear forces of up to 103% of the applied preload, and adhesive forces up to 34% of the preload. This percentage, however, decreased as the preload increased if claws were not sharp (as seen in Fig. 3(a)). In fact, decreasing adhesion at higher preloads is also the primary cause of the high adhesion standard deviations reported for barb numbers 1, 5, and 7 in Table 1. Since barbs will tend to dull with wear, this suggests that distributing the preload over many independently sprung claws may be advantageous. We also observe that adhesion is more greatly affected by barb tip geometry than shear. The second experiment suggests that an approach angle and barb orientation of 45 to 60° is desirable for generating optimal adhesion.

These observations guided the design of our initial prototype foot. A CAD model of a climbing foot and individual toe from this prototype is shown in Fig. 4. Our prototype includes multiple claws and compliance as inspired by these findings and the cockroach foot design. Future prototypes will have multiple toes per foot and multiple pins per toe. Compliance on each toe will allow a maximum number of toes to engage on rough terrain.

The design in Fig. 4 has the nominal claw angle of $\alpha = 60°$ that rotates to 45° when the foot contacts the surface and compresses the toes. This design works relatively well, however, the data suggests that the claws should actually rotate the opposite direction when they contact the surface. Setting the nominal claw angle to $\alpha = 45°$ would allow the lowest possible preload threshold for adhesion. Once initial claw engagement has transpired, rotating the claw toward 60° will allow the maximum adhesion. A toe design as shown in Fig. 5, though slightly more complex than the initial prototype would accomplish this goal. This design could be purely passive, as suggested by the schematic 4-bar linkage shown in Fig. 5, with an instantaneous center of rotation at the wall surface. Note that in addition to the preferential barb

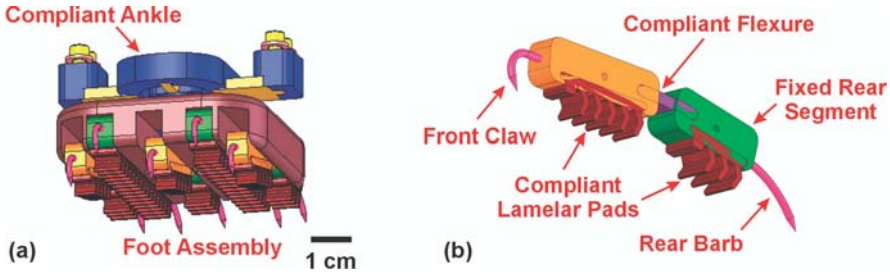


Fig. 4. A preliminary CAD model concept of a proposed robot (**a**) foot and (**b**) toe. A number of toe assemblies would be combined to form a foot

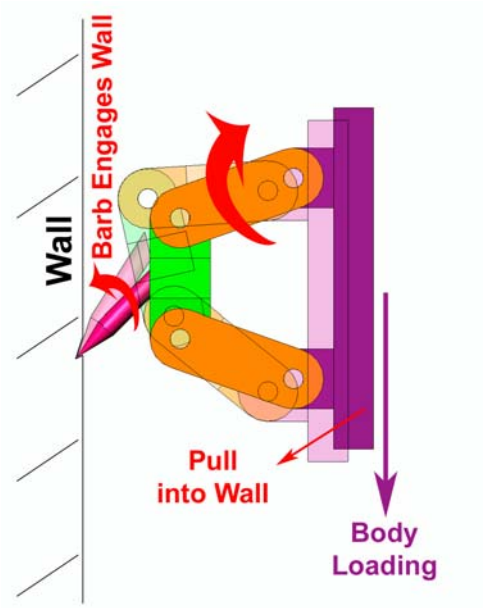


Fig. 5. Kinematic concept of passively engaging claw. The mass of the robot induces rotation of the linkage after initial attachment to the climbing substrate

orientation, this type of toe design also pulls in towards the surface as shearing load from the weight of the robot is applied to the claws.

4 Conclusions

Our results show that claws are promising for smooth surfaces. Our experiments show a preferred claw geometry (small tip radius, R) and insertion angles, $\beta = 45 - 60°$, for good adhesion and shear. The experiments also suggest having many claws to distribute the required adhesive forces and reduce loading on each claw. To accomplish this, claws will need to be independently sprung, necessitating more complex foot designs and raising the bar for mesoscale manufacturing.

This first set of experiments has just scratched the surface of this challenging problem. To better understand the sensitivity of claw orientation on foot attachment, future experiments will look at varying claw orientation and approach angles independently as well as varying retraction angles. We will also focus on the range between 45 and 60° for the optimum barb orientation. Once these effects are better understood, testing at higher impact velocities and testing at the toe level will ensue, to determine how to design toe compliance. For materials other than soft wood, the strain rate is may be important so that quasi-static tests as reported here will no longer suffice. In the future

we will investigate simple models of claw penetration and adhesion and their correlation with experiments.

Acknowledgements

This work was supported by the DCI Postdoctoral Fellow Program under grant number MNA501-03-12002 and by DARPA under grant number NC66001-03-C-8045. The authors also thank Kellar Autumn from Lewis and Clark College for access to lab facilities used to run these experiments.

References

1. K. Autumn and A. Peattie. Mechanisms of adhesion in geckos. *Integrative and Comparative Biology*, 42(6):1081–1090, 2002.
2. C. Balaguer, A. Gimenez, J.M. Pastor, V.M. Padron, and C. Abderrahim. A climbing autonomous robot for inspection applications in 3d complex environments. *Robotica*, 18:287–297, 2000.
3. D. Bevly, S. Dubowsky, and C. Mavroidis. A simplified cartesian-computed torque controller for highly geared systems and its application to an experimental climbing robot. *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control*, 122(1):27–32, 2000.
4. T. Bretl, S. Rock, and J.C. Latombe. Motion planning for a three-limbed climbing robot in vertical natural terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation*. Piscataway, NJ, USA : IEEE, 2003.
5. Z.D. Dai, S.N. Gorb, and U. Schwarz. Roughness-dependent friction force of the tarsal claw system in the beetle pachnoda marginata (coleoptera, scarabaeidae). *Journal Of Experimental Biology*, 205(16):2479–2488, 2002.
6. S.B. Emerson and D. Diehl. Toe pad morphology and mechanisms of sticking in frogs. *Biological Journal of the Linnean Society*, 13(3):199–216, 1980.
7. W. Federle, M. Riehle, A.S.G. Curtis, and R.J. Full. An integrative study of insect adhesion: mechanics and wet adhesion of pretarsal pads in ants. *Integrative and Comparative Biology*, 42(6):1100–1106, 2002.
8. G. La Rosa, M. Messina, G. Muscato, and R. Sinatra. A low-cost lightweight climbing robot for the inspection of vertical surfaces. *Mechatronics*, 12(1):71–96, 2002.
9. Forest Products Laboratory. Nail-withdrawal resistance of american woods. *U. S. Department of Agriculture - Forest Service*, 1965.
10. R. Lal Tummala, R. Mukherjee, N. Xi, D. Aslam, H. Dulimarta, Jizhong Xiao, M. Minor, and G. Dang. Climbing the walls [robots]. *IEEE Robotics and Automation Magazine*, 9(4):10–19, 2002.
11. B.C. Mahendra. Contributions to the bionomics, anatomy, reproduction and development of the indian house gecko hemidactylus flaviviridis ruppell. part ii, the problem of locomotion. volume 13, pp. 288–306. Proc. Indian Acad. Sci., Sec. B, 2003.

12. M. Sitti and R.S. Fearing. Synthetic gecko foot-hair micro/nano-structures for future wall-climbing robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pp. 1164–1170. Piscataway, NJ, USA : IEEE, 2003.
13. Z.L. Xu and P.S. Ma. A wall-climbing robot for labelling scale of oil tank's volume. *Robotica*, 20:209–212, 2002.
14. P.A. Zani. The comparative evolution of lizard claw and toe morphology and clinging performance. *Journal of Evolutionary Biology*, 13:316–325, 2000.
15. J. Zhu, D. Sun, and S.K. Tso. Development of a tracked climbing robot. *Journal Of Intelligent And Robotic Systems*, 35(4):427–444, 2002.

Motion Planning for a Legged Vehicle Based on Optical Sensor Information

Richard Bade[1], André Herms[1], and Thomas Ihme[2]

[1] University of Magdeburg, Unipl. 2, 39106 Magdeburg, Germany
{ribade,aherms}@ivs.cs.uni-magdeburg.de
[2] University of Applied Sciences Mannheim, Windeckstr. 110, 68163 Mannheim, Germany
t.ihme@fh-mannheim.de

Summary. This paper describes the motion planning for a walking robot based on environment information. The planning algorithm is based on random sampling. The environment information are generated by a stereo vision algorithm that has been modified to meet real-time requirements.

1 Introduction

Legged vehicles can help to get access to many parts of earth surface which wheeled vehicles can not cope. Different concepts are used to control legged locomotion in unstructured environment. Simple control concepts use gait pattern, which define a fixed sequence of footholds and are suitable for quite flat terrain. Reactive control components like the elevator reflex [8] are added to improve walking capabilities and terrain adaptation. But problems arise if the exception handling dominates the normal walking process. Path planning can help to minimize exception handling occurring. It uses some environment information to calculate movements avoiding exceptions. At the same it should result in good movements. This leads to the use of an optimization approach as described in Sect. 4.

For getting the needed environment information we use stereo vision. The usual algorithms do not meet our real-time constraints or require special hardware. So we developed an anytime algorithm [3] that returns imprecise but sufficient results if aborted. It is described in Chap. 3.

2 Related Work

There exist many control systems for walking robots. Gait pattern as used in [7, 17] are stable but not reliable on uneven ground. Even reactive control

components like used in [8] have the mentioned drawbacks. Motion planning in [12] is only used to control gait parameters like stroke height. Planning as described in [20] switches between proper gait pattern, but these are limited. In [5] the whole motion of the robot is planned. Genetic algorithms and backtracking is used for this. But there are no real-time assertions. The ordinal optimization approach used in [2] cannot guaranty that a solution is found.

The real-time stereo vision solutions used in [13, 14, 16] require special hardware or set constraints to the target environment [4]. As we focus on a more general approach, we developed an algorithmic solution.

3 Stereo Vision as Anytime Algorithm

Stereo vision algorithms are generally time consuming. To use them within a real-time system, an implementation with a predictable runtime is required and so an anytime algorithm is preferred. This class of algorithms requires that a result is available no matter when the algorithm is stopped.

To get an anytime algorithm we first have to choose a stereo vision method, which we can implement as such an algorithm. Our choice for this was the block-matching method. It is a representative for area based stereo [4]. We decided to use area based stereo, because it can be used either for all pixels of an image or for selected areas [21]. This provides the opportunity to extract features in the image and compute the displacement vectors for these features only. In the future, one can use this opportunity to decrease computation time, if only parts of the image are of interest.

The main idea of the block matching algorithm is a similarity measure between two equal sized blocks ($n \times m$-matrices) in the stereo images. This means, a block in the left image will be compared with all possible blocks of the right image, which have the same size. A standard method to describe the similarity is the mean square error (MSE) of the pixel values inside the respective blocks. For further details on Stereo Vision using the standard block matching method see [6].

But the standard block-matching method can only provide results after the algorithm is finished. To transform it into an anytime algorithm means, changing it to get a first result as soon as possible, which can be gradually improved afterwards. We have solved this problem by using a pyramid model approach.

3.1 Hierarchical Stereo Vision

Tanimoto and Pavlidis [18] introduced the use of pyramid models for image processing. We use a four–level pyramid to enhance the block matching method.

The correspondence problem will be solved on each level of the image pyramid starting with the coarsest level. The results of the previous level

are used to initialize the disparity computation of the more detailed level. Therefore, the search for corresponding blocks starts at the disparities, which were found in the coarse levels. This allows to decrease the search space in horizontal direction. A minimal *MSE* can be found fast. For further details on the algorithm see [11].

The main idea is to be able to use the results of every level as computation result, if the algorithm is stopped in between because of timing constraints. Missing results of one level are completed with results of the previous level. This ensures that the result improves steadily. Therefore, a first result, though imprecise, is available in short time. Because imprecise results are better than no results they can still be used, e.g. to avoid collisions.

Now we show how to speed up our method to faster get the highly detailed results. A serious problem of the block-matching method is the size of the search space. If the epipolar constraint (see [6]) can not be fulfilled the search space must be spread to several rows. Another assumption is made about the order of the pixels in both images. It is assumed that pixels of one epipolar line in one of the stereo images can be found in the same order in the corresponding epipolar line in the other image. But this assumption is violated, if the depth distance between objects in the observed scene is too large. We considered that, as we do not want to restrict the environment. So the block-matching method was changed to allow for a search of corresponding blocks in horizontal and vertical direction. But to limit the search space we use an initialized disparity value (from previous pyramid level) so that definite false results will not be considered. Additionally, the search space in horizontal direction can be limited due to the vision parameters of the cameras. The closer an object to the cameras is, the larger the disparity values of the corresponding pixels will be. If one assumes a minimum distance, e.g. by the stereo camera configuration on the robot, the maximum horizontal search space can be defined exactly.

However, not only the search space defines speed and quality of the results. An important parameter is the block size. We have tested different block sizes and concluded that the ratio between block height and block width should be 2:3, if the epipolar constraint is used further on. In our tests, a block size of 12 $\times$ 8 pixels yields the best results. The hierarchical structure of the algorithm and the available provisional results lead to an anytime algorithm. With this, we have a method that provides depth information from stereo images in real-time.

4 Walking as Optimization Problem

The goal of motion planning is to calculate a good movement based on environment information and application demands. By the claim for a *good* solution the problems may be treated as an optimization problem: Find a valid solution (movement) from the solution space (all possible movements)

that is optimal regarding given objectives. To solve the optimization problem, we need a formal definition and an algorithm for solving it.

4.1 Definition

A formal optimization problem is a triple (U, S, ϕ). The input U parameterizes the set of valid solutions $S(U)$. The weighting function ϕ defines some value by mapping every elements to a real number:

$$\phi : S(U) \mapsto \mathbb{R} \tag{1}$$

This triple has to be defined for our given problem. The input consists of the source and destination position and the environment information generated from the sensor data. The positions are defined by two-dimensional world coordinates. The environment information is given by a height map which defines the height of a point every 100 mm.

The solution space represents all valid movements. It should include *every* possible solution. Otherwise a good one could already be excluded by the definition. The whole movement is composed by the motion of the individual legs and the robot body. Each of them is described by a reference point. For the legs this is the foot, for the body the center of gravity. Their movement is described by a list of *events*. An event defines a linear movement between two points in a given time. By concatenation it can be used for linear approximation with arbitrary precision. So we have seven lists of such events for describing the movement of the robot.

The weighting function assigns a value to reflect the quality of a movement. We considered several criteria for this. They are combined to define the final value. The most important criterion is the speed. We use the direct distance to the destination divided by the time. The faster, the better the movement is. An additional point we considered is the stability of the robot. This is affected by the ground and the arrangement of the legs. For the latter case the stability margin (as described in [15]) gives a good measurement. We calculate the minimal value for the whole movement. The bigger, the lower is the risk to tip over. For the stability of the ground we considered steer areas. The robot can loose it's grip on this. We use the maximal gradient of the ground under all steps. The lower it is, the better the movement. All these conditions can be incorporated into a weighting function $\psi_i : S(U) \mapsto \mathbb{R}$. These terms are linearly combined to the final value: $\phi(s) = \sum_i \lambda_i \psi_i(s)$. The λ_i allow to adjust the influence of the criteria.

4.2 Possible Heuristics

Due to the complexity of the problem no exact solving algorithm is known. So we concentrate on finding a good heuristic. Some of the commonly used methods to solve optimization problems were investigated regarding their

Table 1. Selection of possible heuristics

Heuristic	Anytime Algorithm	Parallel Computation	Required Memory	General Applicability
greedy	no	no	low	bad
branch and bound	yes	yes	low	bad
local search	yes	multi start	low	good
tabu search	yes	multi start	high	good
random sampling	yes	yes	low	very good
simulated annealing	yes	multi start	low	good
genetic algorithms	yes	yes	high	good

practicability for our problem. At first this is only done theoretically by checking common criteria. The results are listed in Table 1. Their description can be found in [1, 10, 19].

We want some anytime algorithm [3] that allows us to stop the calculation at any time giving a valid (maybe suboptimal) result. Furthermore, to enable the possible use of all processors in the robot, parallel computation should be possible and the memory requirements should not be too high. The most important requirement is the general applicability to the problem. This ensures that we can use it for our problem.

As a result, three possible heuristics are chosen: *local search*, *random sampling* and *simulated annealing.*

4.3 Implementation

For the three heuristics we need some way of generating random valid movements. This appears to be a nontrivial task. No efficient algorithm is known for only determining, if a valid movement exists [5]. So we lower our requirement to an algorithm which generates random movements that are valid in most cases. The result is a multistage process as described in [9]. For proper handling of invalid solutions we use a penalty value. They are treated as valid but really bad. So they should never occur as optimum.

For local search and simulated annealing a neighborhood definition is needed. Here a neighbor is defined by a small difference in one of the movement parameters, e.g. a foot position.

The three heuristics are applied to our optimization problem. The resulting algorithms follow the usual description. For testing we use a virtual environment. The world model is created manually at the moment. It is represented by a raster image where the green channel of a pixel corresponds to the height. This allows us to create various scenarios. The movement of the robot is planned regarding the given terrain. To verify the results we developed a visualization (see Fig. 1).

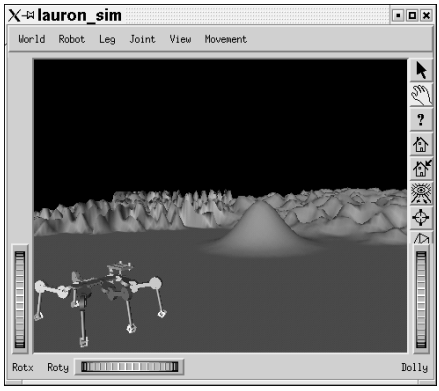


Fig. 1. Robot visiualization

5 Experimental Results

5.1 Results from Stereo Vision

We found out that our anytime stereo vision algorithm provides results much faster than the unmodified stereo vision algorithm we used as basis, if more imprecise results are acceptable. In Fig. 3 one can see that the results after every pyramid step becomes better. This is, the more time is available for the computation, the more detailed results are produced. For motion planning, an earlier result could be used as big structures and objects are already visible. This is sufficient for avoiding large obstacles.

5.2 Results from Motion Planning

Our tests with the different motion planning algorithms showed that only the random sampling approach is suitable for practical use. Instead of the other ones its computation time meets our requirements.

As random sampling is a probabilistic algorithm we had to run it several times for evaluation. Figure 2 shows the valid results for walking on even terrain. We got 2 259 valid movements for 10 000 iterations. So the probability for a valid solution per iteration is $p = \frac{2259}{10000} = 0.2259$. The chance of finding at least one valid solution depends on the number of iterations n. it can be computed by $P(\text{n times}) = 1 - (1 - p)^n$. For $n = 100$ it is already 0.999 999 999 992 419. So you can assume that after a certain number of iterations the chance of finding a valid one is nearly one. Figure 2 shows the distribution of the solutions. To get a solution of the best 1% more iterations are required. In this example there is a probability of 0.89 for 1 000 iterations. This is typical for a calculation time of 30 seconds.

The same way other scenarios are tested: *trench, step, unsurmountable obstacle,* and *irregular terrain.* All of them are solved sufficiently. The benchmarks *trench* and *step* are constructed in a way that normal regular gait

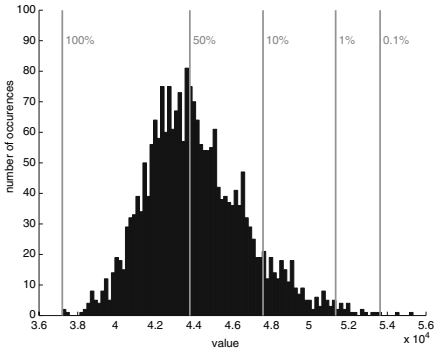


Fig. 2. Results: flat ground

Fig. 3. Original image on the left and the results after every pyramid step

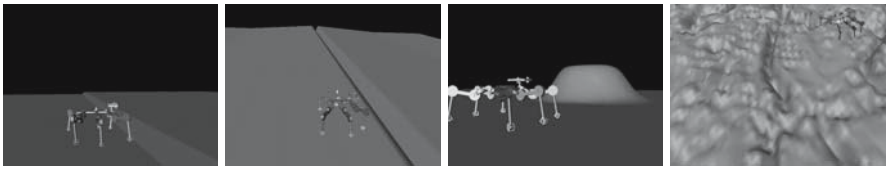


Fig. 4. Scenarios: step, trench, unsurmountable obstacle, irregular terrain

pattern could not handle them. The robot has to choose the right distance to the obstacle and a special order of moving the feet. Our motion planner is able to handle this. The scenario *unsurmountable obstacles* demonstrates that the motion planner is even able to correct wrong guidelines from the application. The robot should walk to point straight ahead. But in direct line there is an obstacle it can not overcome. The planner handles this situation by walking around. The scenario of *irregular terrain* represents a typical case. It contains many steer areas the robot should not step on. The planner is also able to handle this.

6 Conclusion

Concluding, we want to point out that we dealt with the problem of motion planning based on information provided by optical sensors. This problem includes two main subproblems. First, obstacles had to be detected under real-time conditions. Second, this data were used for motion planning. Obstacle

detection was solved with an anytime version of a stereo vision algorithm. This allows to comply with the real-time constraints. The motion planning has been treated as an optimization problem, which allows solving it with a good heuristic.

References

1. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer Verlag, 2000.
2. C.-H. Chen, V. Kumar, and Y.-C. Luo. Motion planning of walking robots in environments with uncertainty. In *Journal for Robotic Systems*, Vol. 16, pp. 527–545. 1999.
3. T. L. Dean. Intrancibility and time-dependent planning. In *Workshop on Reasoning about Actions and Plans*, 1987.
4. E. D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1, 1988. Springer International.
5. C. Eldershaw. *Heuristic algorithms for motion planing*. D.Phil. Thesis, University of Oxford, 2001.
6. O. Faugeras et al. Real time correlation-based stereo: algorithm, implementations and applications. Technical Report 2013, Institut national de recherche en informatique et en automatique, 1993.
7. Cynthia Ferrell. A comparison of three insect-inspired locomotion controllers. *Robotics and Autonomous Systems*, 16:135–159, 1995.
8. B. Gaßmann, K.-U. Scholl, and K. Berns. Locomotion of LAURON III in rough terrain. In *International Conference on Advanced Intelligent Mechatronics*, volume 2, pp. 959–964, Como, Italy, July 2001.
9. A. Herms. Entwicklung eines verteilten Laufplaners basierend auf heuristischen Optimierungsverfahren. Diploma thesis, University of Magdeburg, 2004.
10. J. Hromkovič. *Algorithmics for hard problems: introduction to combinatorial optimization, ranomization, approximation and heuristics*. Springer Verlag, 2001.
11. T. Ihme and R. Bade. Method for hierarchical stereo vision for spatial environment modelling supported by motion information. In *Proceedings of Robotik 2004*. VDI-Verlag, June 17–18, 2004. Munich, Germany.
12. M. A. Jiménez and P. González de Santo. Terrain-adaptive gait for walking machines. *The International Journal of Robotics Research*, 16(3), June 1997.
13. K. Konolige. Small vision systems: Hardware and implementation. In *Eighth International Symposium on Robotics Research, Hayama, Japan*, pp. 203–212. Springer Verlag, oct 1997.
14. A. Kelly L. Matthies and T. Litwin. Obstacle detection for unmanned ground vehicles: A progress report. In *International Symposium of Robotics Research, Munich, Germany*, oct 1995.
15. E. Papadopoulos and D. Rey. A new measure of tipover stability margin for mobile manipulators. In *IEEE International Conference On Robotics and Automation*, 1996.
16. T. Ohm R. Volpe, J. Balaram and R. Ivlev. The rocky 7 mars rover prototype. In *International Conference on Intelligent Robots and Systems, Osaka, Japan*, volume 3, pp. 1558–1564, nov 1996.

17. Shin-Min Song and Kenneth J. Waldron. An anaytical approach for gait study and its applications on wave gaits. *The International Journal Of Robotics Research*, 6(2):60–71, Summer 1987.
18. S. Tanimoto and T. Pavlidis. A hierarchical data structure for picture processing. In *Computer Graphics and Image Processing*, volume 4, 1975.
19. P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, 1987.
20. D. Wettergreen. *Robot walking in natural terrain*. PhD thesis, Carnegie Mellon University, 1995.
21. C. Zhang. A survey on stereo vision for mobile robots. Technical report, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh.

Developing Climbing Robots for Education

K. Berns, T. Braun, C. Hillenbrand, and T. Luksch

Robotics Research Lab, University of Kaiserslautern, Germany
{berns, braun, cahillen, luksch}@informatik.uni-kl.de

Summary. This paper presents a practical course that introduces students to the development and construction of a mobile robotic system. Using a wheeled climbing robot for metallic walls as an example, the course aims at teaching practical skills in engineering as well as developing soft skills like project management and teamwork. To increase motivation and allow the construction of a working system in the limited time available during one university term, the students are provided with several pre-made components and software tools. In the paper, the general structure of the course is presented and the different deliverables are detailed along with the supplied components. The paper ends with some example designs students came up with during the first time that the course was held in winter 2003/04 and an evaluation of the learning effect that was achieved.

1 Introduction

A lab offered for students in the area of robotics should not only provide them with technical knowledge about robots, but also give insights into the general field of robotics and some problems commonly encountered there. Such a course should be challenging but fun to take and encourage students to continue working in that area. The construction of a mobile robot is a topic that is well suited for this task. Building a real robot requires engineering skills like system design, noise tolerant signal processing and the development of an adequate control structure. These fundamentals are rarely taught in the usually theory-oriented courses in computer science.

Motivated by this, a practical course was developed that introduces students to the development and construction of climbing robots. The goals of the course are to familiarize the students with the design methodology needed for the development of a complex robotic system, the practical problems arising during construction and team work within a medium sized project group. The interdisciplinarity of the subject requires a course that can accomodate students with various backgrounds and different skill levels. In Sect. 3.1, the structure of the course is introduced.

To present an interesting and active research area, the course focuses on climbing machines. Because of the limited amount of time available during one university term, the robot that should be built was chosen to be *wheel-driven* and restricted to metallic walls, so that permanent magnets can be used for adhesion. This type of robot can be constructed and controlled faster than for example a legged climbing machine. A typical application for such a robot could be cleaning or painting wall sections in an office building. Even with this type of robot, the design of a complete system from scratch is not feasible. Therefore, the students are provided with several pre-made components and use an existing software framework, which will be described in more detail in Sect. 3.2. Some results of the first time the course was given are summarized in Sect. 4. The paper will close with a discussion of the learning effect that was achieved and an evaluation of the benefits and difficulties that are connected with such a course.

2 Related Work

Robots have been an interesting topic for education for some time now. A lot of different approaches exist, many of them based on commercial robotic toolkits like the LEGO Mindstorms [2] set. One advantage of these kits is that they reduce the amount of low-level 'hardware' work that needs to be done so that even students without a background in mechanical or electrical engineering can successfully design a functional robot. In [7], the outreach program BotBall is presented which uses such a kit and focuses on an audience of mid to high-school pupils and undergraduate students. A different approach for university education is presented in [4]. This course is designed for the education of students in electrical engineering and therefore sets a priority on the construction of robotic hardware instead of higher level control software. The aspect of telematics and teleoperation, missing in many other approaches, is considered in [5] and aiming at planetary exploration applications. An overview of different types of robots that can be used for educational purposes is given in [3], while [1] presents a non-commercial toolkit for one specific autonomous indoor wheel-driven robot.

3 Course Description

3.1 Concept

The time available for the course is approx. 19 weeks, with students working one half to one full day per week on the term project. At the beginning of the course, groups of 6 people each are formed. One project manager is elected, who acts as spokesperson for the group and is in charge of organizing group meetings, defining milestones and meeting deadlines. He is also responsible

for enforcing a coding standard and the use of a version control system. Obligatory group meetings take place on a weekly schedule, helping to detect both technical and personal problems within the team [4]. For the actual implementation work, the formation of 2-person-subgroups is encouraged. This forces students to specify interfaces between different aspects of the robot which leads to a clearer and more explicit design.

The course is split into two parts. The first part is an introduction to the software framework and some of the supplied hardware components such as sensors and motor electronics. It consists of five small assignments with the following topics:

1. introduction to the software framework *Modular Control Architechture 2* (MCA2) [6]
2. implementation of a simple control program for a differential drive
3. programming and simulation of a path planning algorithm
4. motor control with a CPLD
5. yaw angle calculation based on an inertial sensor

Each assignment has to be finished within one week and presented to the teaching assistant along with a code inspection. This way, the students learn how to use the software well and some of the produced code can also be reused as a basic frame for the real robot, which saves precious time. In addition, the assignments give group members the opportunity to estimate their competence in the various aspects adressed, giving them a chance to choose a suitable area of work for the second part of the course.

This second part takes about three quarters of the available time and deals with the construction and programming of the climbing robot. The students are given three application scenarios that the robot should be able to cope with (see Fig 1). In the first scenario, the robots task is to completely cover a rectangular area with known dimensions and no obstacles. Additionally, the used algorithm should guarantee that no area is passed twice. This navigation strategy is easy to motivate and could for example used by a wall painting robot. The second scenario adds a *forbidden area*, a region

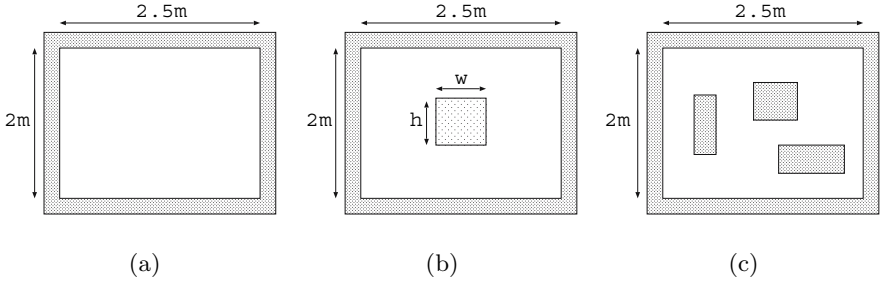


Fig. 1. Three different application scenarios for the climbing robot. The dark shaded areas represent obstacles, the *light shaded* area in 1b a 'forbidden region'

whose position and size is known a priori and which must not be crossed by the robot. This situation could occur for example when the robot needs to avoid door or window openings. In the last scenario, the area to be covered is *not* known beforehand. This implies that the robots needs to explore the environment using its sensors. Some sort of path planning algorithm needs to be implemented in a way such that the generated path covers as many free regions as possible while avoiding obstacles and double crossings.

In addition to the application scenarios, each team is supplied with several components (motor units, sensors, wheels etc.) that will be described in more detail in Sect. 3.2. There are no restrictions placed on the used algorithms or construction details of the robots, as long as it is able to work in the scenarios. This approach not only trains implementation skills, but also forces the students to go through a specification phase and encourages them to use their own creativity. During the second part of the course, there are no deliverables or deadlines imposed on the students – it's their own responsibility to organize their time and working force. Of course, the teaching assistants are available during lab hours to answer questions and give technical advice. To allow the students to work on the control software and the hardware setup of the robot simultaneously, a basic simulation of a two-wheeled robot is provided in MCA. This simulation does include realistic motor behaviour and odometry data, but no other sensors, maps of the environment or higher level functions. Still, using and enhancing this simulation, software subgroups are able to implement behavioural algorithms for the climbing robot while the hardware itself is being constructed by the hardware subgroups. This approach speeds up the whole process significantly.

During the last two weeks of the course, each student group has to prepare a final presentation, write a report and give a practical demonstration of the robot. Grading is based on the quality of these deliverables and the five assignments.

3.2 Supplied Components

Each group of students is provided with a toolkit of hardware components that contains (see Fig. 3):

- 1 wooden board
- 1 board with a DSP and power + I/O electronics
- 2 motor-and-gear units with wheels
- 3 infrared sensors
- 2 ultrasonic sensors
- 4 permanent magnets with mountings
- 1 inertial system
- 2 bumpers

The controller board, which is under constant development in the robotics lab at Kaiserslautern, consists of a Motorola DSP 564803 connected with

a CPLD, two DC Motor Control circuits and several A/D ports for the sensory equipment. The motor units together with the chosen gears are able to generate a force of 40N at the contact point with the wall, while the encoders generate 512 impulses per motor axis turn, which results in a very accurate measurement of the wheel's rotation. The infrared sensors work in a range of 3-30 cm, while the ultrasonic sensors have a longer range. The magnets generate a force of 450N on contact with a metallic wall; this force drops to 20% with an air gap of 1 mm and below 10% over 2 mm. This indicates that the distances of the magnets need to be carefully adjusted. Table 1 lists possible sources for the components. The price for one set totals up to about 1000 €, but all components can be reused, so that the expense is acceptable.

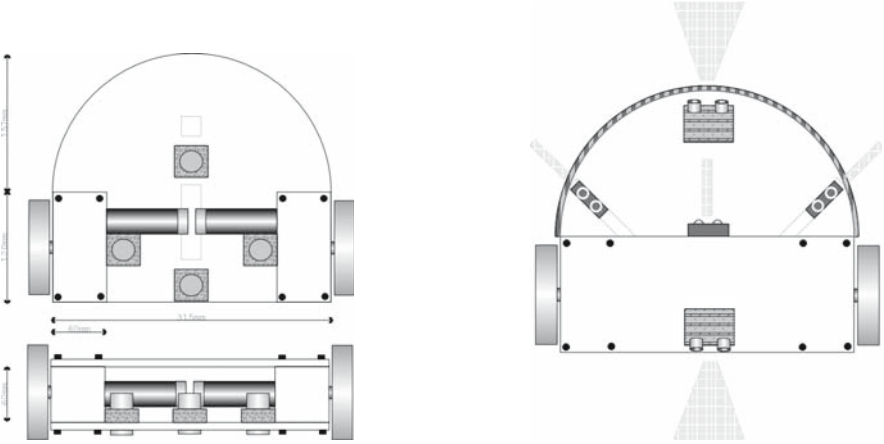
Table 1. Sources for the used components

Component	Source	Part No.
Motor	Faulhaber	2657
Gear	Faulhaber	26/1
Encoders	Faulhaber	IE2-512
Magnets	Schramberg	01-9200038
IR-Sensors	Sharp	GP2D12/GP2D15
US-Sensors	Murata	MA40B8S/R
Inertial-Sys.	Analog Devices	ADXRS150/ADXL103

4 Results

The practical course has been offered during the winter semester 2003/2004 at the University of Kaiserslautern and was taken by three groups of students (20 students in total). All of them have succeeded in designing and building a robot that moves along vertical walls in the robotics lab. Figure 2 shows the mechanical setup of one robot developed by the students. In Fig. 2(a), the position of the 4 permanent magnets is indicated by the grey squares. In Fig. 2(b), the layout of the sensors is displayed; the two ultrasonic sensors are facing in frontal and backwards direction, the three infrared sensors are placed left, right and at the top of the robot and directed in a 45 degree angle towards the wall. Figure 4 shows the assembled robot driving along a wall in the lab.

Unfortunately, the construction of a new controller board took place during the practical course and was delayed because of unforeseen capacity problems with the CPLD. Therefore, the controller was not available for the students and the sensor data delivered by the infrared and ultrasonic sensors could not be processed by the robot as planned. This forced the groups to work mainly with the MCA-simulation (Fig. 5 shows a test run) and led to the removal of



(a) Mechanical layout with differential drive and permanent magnets

(b) Sensoric system with infrared and ultrasonic sensors

Fig. 2. CAD Layout of a robot developed by one student group

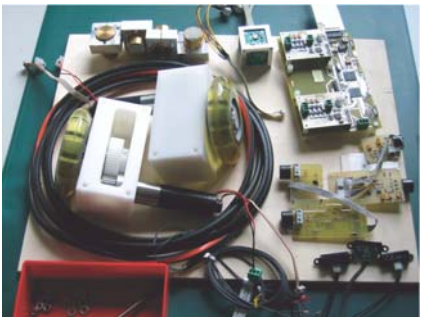


Fig. 3. The components given to the students

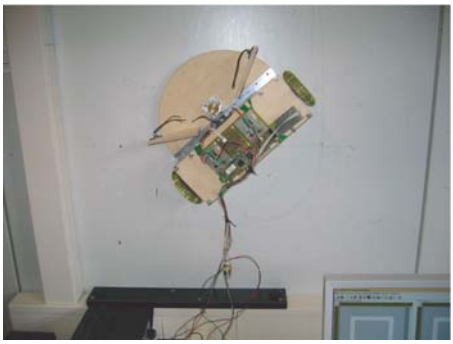
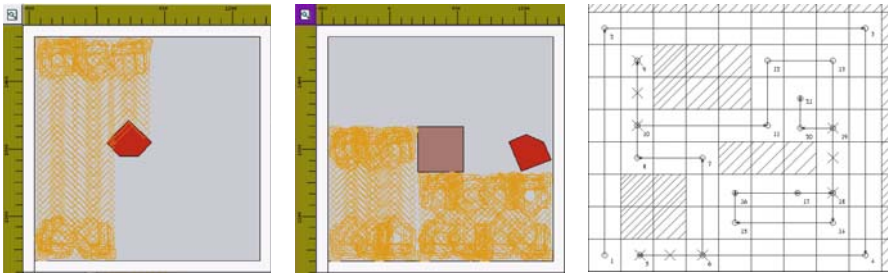


Fig. 4. An assembled climbing robot. The sensors are only partially mounted and not connected to the controller



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 5. Evaluation of a path planning algorithm in the MCA simulation. The covered area is marked in orange. For scenario 3, only a paper based simulation was performed.

the third application scenario from the deliverables. However, motor control and odometric measurements were possible with an older version of the controller, so that at the end of the course, at least the first two scenarios could be demonstrated.

After the initial specification of the robot, the prototypical implementation of the basic algorithms in the MCA-simulation took place together with the setup of the real hardware. As soon as the first test runs were performed, several unexpected effects occured, such as wall bending caused by the magnets, unequal drift of the wheels for different driving directions etc. The groups had to react to these effects by redesigning the robots (for example devising new positions for the magnets and reinforcing the robot frame) and the navigational strategies. This way, the complex interaction between software, hardware and the application scenario became obvious. The learning effect achieved by the lab was substantial.

In spite of the problems encoutered, the students were highly motivated throughout the course, spending a lot of time on developing and optimizing their part and helping out other students when needed.

5 Conclusion

In this paper, a lab was presented that gives an introduction to the development of service robots. The students taking this course did not only get a good insight into robotics and learned technical skills, but also acquired soft-skills like time management and teamwork. This is a key qualification for the industry and certainly increases their chances of getting a good position after university. Although the previous knowledge of many students was limited, the designed prototypes all were of high quality. This was made possible through the use of pre-made components and a co-design approach for soft- and hardware simultaneously, which left enough time for several redesigns.

This course also shows that the construction of a climbing robot is possible given the limited time frame of one term.

Possible future work includes the integration of a new controller board so that the sensors can be used by the robot, adding a power supply and a small pc-on-a-chip systen to make the robot autonomous and the inclusion of an effector (for example, a liftable sponge) for painting or cleaning.

References

1. T. Gockel, O. Taminé, P. Azad, and R. Dillmann. Edukabot – aufbau eines edukativen roboter-baukastensystems. Institut für Rechnerentwurf und Fehlertoleranz (IRF), Universität Karlsruhe, 2003.
2. The LEGO Group. *www.legomindstorms.com/*.
3. H. Loose. Wheeled and legged robots in collegiate education. In *International Colloquium on Autonomous and Mobile Systems*, pp. 179–182, Magdeburg, June 2002.
4. D. J. Mehrl, M. E. Parten, and D. L. Vines. Robots enhance engineering education. *Frontiers in Education Conference*, pp. 613–618, 1997.
5. K. Schilling, H. Roth, and O. J. Rösch. Mobile mini-robots for engineering education. In *3rd UICEE Annual Conference on Engineering Education*, 2002.
6. K.-U. Scholl, B. G. J. Albiez, and J. Zöllner. MCA – modular controller architecture. In *Robotik 2002, VDI-Bericht 1679*, 2002.
7. C. Stein. Botball: Autonomous students engineering autonomous robots. *ASEE Annual Conference and Exposition*, June 2002. Montréal, Quebec, Canada.

Robust Localisation System for a Climbing Robot

André Martins, Lino Marques, and A.T. de Almeida

Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra 3030-290 Coimbra, Portugal
{amartins,lino,adealmeida}@isr.uc.pt

Abstract. This paper describes a method for localisation of one or more climbing robots moving on a planar surface. The proposed method uses low-cost technology and can be implemented using a common PC-based computer with a minimum of one webcam to acquire pictures of the mobile robots workspace. The vision system tracks coloured markers placed on the top of mobile platforms and can be easily calibrated through known target displacements. The main advantages of the proposed method are its non-intrusive nature and its ability to be easily adapted to compensate the platforms dead-reckoning errors.

1 Introduction

A relevant problem in mobile robotics is the absolute localisation of moving robots or objects. Most of the methods commonly employed to solve this problem require some kind of change the robot's hardware, like adding sensors such as ultrasounds, IR and electromagnetic sensors or cameras. While this can be effective, sometimes it is not possible or desirable, so some *external* method is required.

This paper presents an easy to set up, vision based method, for providing position values of a climbing robot moving on a wall. It is aimed for indoor or outdoor experimentation, and requires no additional hardware on the robots. Using only low-cost hardware (a minimum of one "web-camera"), it can track several colour coded robots. The system tracks a colour target placed on top of the robot, and different colours or colour combinations can be used in order to track multiple robots. The use of more cameras can provide a better coverage of the workarea, with a consequent reduction of the probability of occlusion. World coordinates are acquired through an homography transformation, requiring no knowledge of the cameras intrinsic parameters. Using an additional marker in the robot, heading information can also be extracted. To improve this technique, sensor fusion with any other available localisation method can be used, providing a better estimate

of the robot's state, and reducing the long term drift introduced by the dead reckoning system. In this paper, we use an Extended Kalman Filter (EKF) to fuse dead reckoning information from the robot with the global localisation.

This method follows and adapts previous localisation works with communities of wheeled mobile robots [5] for the model and self-localisation system of the climbing robot described in the following section.

2 Visual Localisation System

The absolute position of the robot is determined by vision. The vision system must be able to track multiple colour markers in images acquired from one or more cameras and map their positions in world coordinates.

Tracking colours under varying lighting conditions, or distinguishing colours in a cluttered environment can be a problem, but here we make several assumptions on the environment: since the system was thought for laboratory experimentation and requires static cameras, the choice of the colour markers can be made a priori, and based on the existing background to facilitate distinction. In order to deal with changes of lighting conditions the (R, G, B) values are first normalised:

$$\begin{cases} (r, g, b) = \frac{255*(R,G,B)}{R+G+B} & (R+G+B) > \gamma \\ (r, g, b) = (R, G, B) & (R+G+B) \leq \gamma \end{cases} \tag{1}$$

The threshold γ is used to filter dark pixels from the normalisation process, since these pixels usually have one dominant value that would be greatly amplified. The choice of this threshold is based on.

The normalised image is then submitted to a colour blob detection algorithm similar to the one used in the CMUCAM [1] that extracts the centroid coordinates of each blob. The algorithm checks if each set of (r, g, b) values lay between predefined bounds, and accumulates the u and v positions for the detected pixels. Then the sums of all u and v coordinates are divided by the total number of pixels detected to determine the centroid of the blob. Once a marker is being tracked, a Region of Interest (ROI) is created around its centroid, and used as a search area for the next blob search. The size requirements for this ROI are quite simple:

$$radius = \frac{max(size_u, size_v)}{2} + \delta \tag{2}$$

where *radius* is the radius of the ROI, *size* is the current marker size in pixels and δ is a constant related to the maximum possible displacement between readings at the current frame rate, to assure the ROI will include the robot in the next run. The ROI can also be used to detect if the robot is partially out of the camera field of vision.

The real world coordinates (x, y) are calculated from the image coordinates (u, v) using the method proposed by [2]. This method relates the real and image points by the simple equations that follow:

$$\begin{cases} x = \dfrac{a'u + b'v + \lambda_1}{pu + qv + 1} \\ y = \dfrac{c'u + d'v + \lambda_2}{pu + qv + 1} \end{cases} \tag{3}$$

where (u, v) coordinates are the centre of the colour blob detected in the camera image, and the (x, y) coordinates must all belong to the same real world plane. The constants a', b', λ_1, c', d', λ_2, p and q are determined by some previous calibration, as is described in the next subsection.

The heading of each robot can be determined from the angle of the displacement vector $\mathbf{d} = \mathbf{X_i} - \mathbf{X_{i-j}}$ where $\mathbf{X_i} = (x_i, y_i)$. However, this method will not be able to track pure rotations. Since climbing robots often use rotations, a different approach is required. Using a secondary marker on the robot, the heading of the robot can be determined by calculating the orientation of the line passing the two centroids. It should be noted that this approach doesn't take into consideration any possible partial occlusion while determining the robot's position, so some error may occur due to incorrect localisation of the marker's centroid.

2.1 Calibration

Since all real world coordinates must be in the same plane it is important that all colour targets in the robots be positioned at the same height or that each robot uses its own calibration values.

The values of a', b', λ_1, c', d', λ_2, p and q are camera dependant and must be determined after positioning or replacing a camera. They can be determined by matching at least 4 points from the image plane to points from the real world plane and calculated as

$$\begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 \\ \cdots & & & & & & & \end{bmatrix} \mathbf{Z} = \begin{bmatrix} x_1 \\ y_1 \\ \cdots \end{bmatrix} \tag{4}$$

where $\mathbf{Z} = [a'\ b'\ \lambda_1\ c'\ d'\ \lambda_2\ p\ q]^T$. Using more points well spread across the workspace for the calibration will result in a better accuracy, and the system above can still be easily solved through a least squares minimisation approach.

3 Sensor Fusion

Combining a robot's internal state with the measurements acquired from the vision system will provide a better estimate of that robot's state, reducing the

long term drift introduced by any dead reckoning system. Since the model for the robot's position is nonlinear (see Sect. 3.1), this combination is achieved through the use of a discrete EKF [3, 4].

Since the actual values of the noises present are unknown, the state and measurement vectors are approximated by dropping the explicit noises from the general equations for nonlinear systems.

$$\hat{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k; 0) \tag{5}$$

$$\hat{\mathbf{z}}_k = \mathbf{h}(\mathbf{x}_k; 0) \tag{6}$$

where $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{z}}_k$ are a posteriori estimates of the state. The system and measurement noises are assumed to be zero mean, Gaussian noise, and represented by their covariance matrices $\mathbf{Q}$ and $\mathbf{R}$.

The robot keeps a prediction of it's system state, $\hat{\mathbf{x}}_k^-$, and of the state error covariance matrix, $\mathbf{P}_k^-$, using the EKF time update equations:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k; 0) \tag{7}$$

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1}^- \mathbf{A}_k^T + \mathbf{Q}_{k-1} \tag{8}$$

where the matrix $\mathbf{A}_k$ is the Jacobian of the system function $\mathbf{f}$.

At the times when the vision system produces measurements ($\mathbf{z}_k$), the measurement update equations are used to update the state estimate $\hat{\mathbf{x}}_k$ by combining the a priori estimate $\hat{\mathbf{x}}_k^-$ (from the dead reckoning system) with the measurement residual ($\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, 0)$):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{9}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, 0)) \tag{10}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \tag{11}$$

and $\mathbf{H}_k$ is the measurement Jacobian, with:

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}} (\hat{\mathbf{x}}_k^-, 0) \tag{12}$$

3.1 System Model

The climbing robot presented in Fig. 1(a) consists of a translation part (1) and a rotation part (2). These parts are connected by means of a junction bracket. The translation part has two pneumatic cylinders with pedipulators fixed at the ends of cylinder's piston rods. They have concentric grippers (4) and sealing grippers (5) that are actuated by means of lifting cylinders (6). The rotation part has a pneumatic rotary actuator with the same combination of the described grippers and lifting cylinders (7). Robot motion is carried out by alternate vacuum fixation of the translation part grippers and rotation part

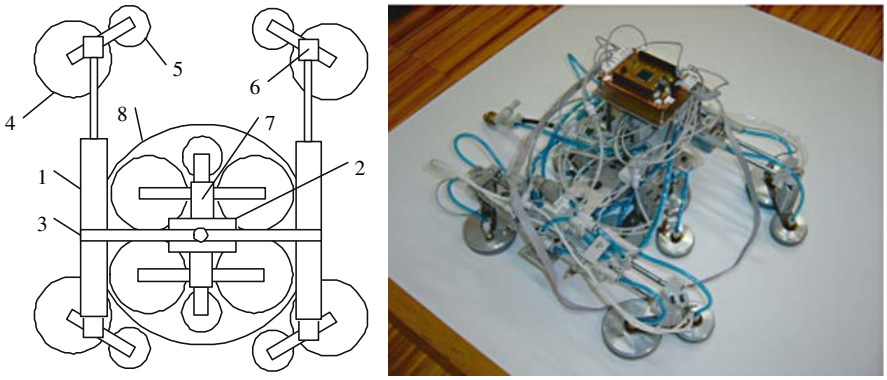


Fig. 1. Climbing robot used – mechanical architecture (*left*) and photo (*right*)

grippers to motion surface and step-by-step motion of the cylinder piston rods and cylinder bodies [6]. Full step detection is achieved using end-of-stroke detectors. The robot has a potentiometer for measuring the angle between the translation part (1) and the rotation part (2), and a 2D inertial sensor that gives inclination (gravity while static) and allows an estimation of the movement by the double integration of the platform acceleration. The localisation of the system can be estimated by dead reckoning. This method can be described as determining the current position based on a known position, and calculating the movement of the robot over time based on speed and heading, without any external references. Information about current position (x, y) and heading θ can be derived from the kinematics equations:

$$x_{k+1} = x_k + \Delta D_k \cos(\theta_k + \Delta\theta_k) \tag{13}$$

$$y_{k+1} = y_k + \Delta D_k \sin(\theta_k + \Delta\theta_k) \tag{14}$$

$$\theta_{k+1} = \theta_k + \Delta\theta_k \tag{15}$$

where ΔD_k is the translation movement and $\Delta\theta_k$ the change in orientation, in the last time period. As such, the state vector will be $\mathbf{x} = [x\ y\ \theta]^T$ and equations 13, 14 and 15 will represent the system functions for the EKF, using ΔD_k and $\Delta\theta_k$ as the system inputs.

The Jacobian of the system function ($\mathbf{A}$) will be:

$$\mathbf{A}_k = \begin{bmatrix} \frac{\partial f_x}{\partial x_k} & \frac{\partial f_x}{\partial y_k} & \frac{\partial f_x}{\partial \theta_k} \\ \frac{\partial f_y}{\partial x_k} & \frac{\partial f_y}{\partial y_k} & \frac{\partial f_y}{\partial \theta_k} \\ \frac{\partial f_\theta}{\partial x_k} & \frac{\partial f_\theta}{\partial y_k} & \frac{\partial f_\theta}{\partial \theta_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta D_k sin(\theta + \Delta\theta_k) \\ 0 & 1 & \Delta D_k cos(\theta + \Delta\theta_k) \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

Since the measurements acquired from the vision system are also position and orientation, the measurement Jacobian $\mathbf{H}_k$ will be the Identity matrix.

3.2 Filter Initialisation

Matrixes $\mathbf{R}$ and $\mathbf{Q}$ will determine the belief in the position and heading information acquired from the vision and dead reckoning systems. The measurement noise (associated with the vision system) is related to the error in the colour blob centroid coordinates, and is related to white noise from the imaging sensor (in webcams it is always present). The measurement noise covariance $\mathbf{R}$ can be calculated a priori by finding the error variance in a series of values acquired from the vision system. We should note that this error depends on the pose of the camera and increases with the distance to the target. Due to the method used to determine the heading, its error is larger than the positional error. However, if the workspace is known in advance, position and orientation noise standard deviations σ_{vx}, σ_{vy} and $\sigma_{v\theta}$ can be experimentally estimated in order to construct the measurement noise covariance matrix:

$$\mathbf{R}_k = \begin{bmatrix} \sigma_{vx}^2 & 0 & 0 \\ 0 & \sigma_{vy}^2 & 0 \\ 0 & 0 & \sigma_{v\theta}^2 \end{bmatrix} \tag{17}$$

The process noise covariance matrix $\mathbf{Q}$ is harder to determine since odometry errors depends from the actuators stroke. Odometry errors occur mainly on fractional steps and in fractional rotations, when end-of-stroke detectors are not used. However, even during planned full stroke motions, fractional steps or rotations might be necessary in situations where there is a flaw in the wall and the robot can not get enough grip, having to search for a suitable position with grip. This process generates unexpected errors that can become a problem in localisation.

4 Experiments

Testing of the system was performed both indoor and outdoor. Outdoor experimentation was idealised to prove the effectiveness of the visual localisation method, while most error analysis was performed in a safer indoor environment.

4.1 Results

The visual localisation method had already been tested in [5]. This time, experimentation was mostly focused on outdoor performance and heading extraction. The climbing robot uses end-of-stroke detectors, and shows a clear

(a) (b)

Fig. 2. Screenshots from the robot tracking. *Left* show outdoor tracking with centroid and ROI. *Right* shows tracking of both robot and heading marker

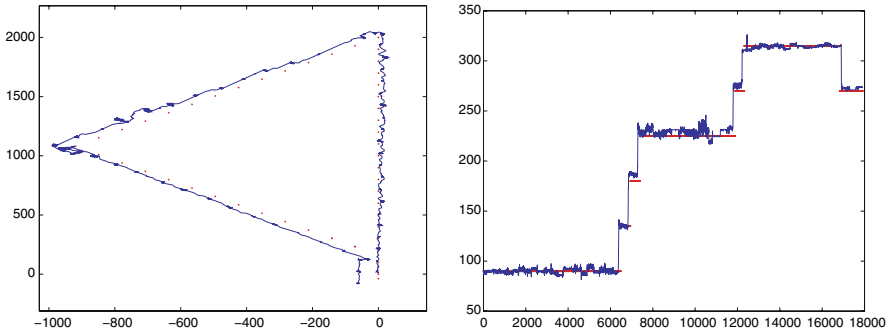


Fig. 3. Experimental Results. *Left* shows path perceived by robot (*dotted line*) and vision. *Right* shows heading from robot(*straight lines*) and vision

delay compared to the faster vision system. Also, using only these detectors, it is impossible for the robot to track it's position between full steps. However, position and angle values are also being tracked by the vision system, and at a fast enough rate to allow integration into the robot's state during its motion.

When the potentiometer is used to track partial rotations, the heading error can be reduced to $< \pm 1°$. Partial translation movement, however, was not tracked yet using the robot's acceleration sensors, so a maximum error equal to the full stroke length (10 cm) could be achieved during translations. This error can be effectively compensated using position data from the vision system.

4.2 Future Work

To reduce the error in the detection of colour blobs, a model fitting algorithm can be used to effectively determine the robot's position. In our situation, an ellipse could be used to effectively match the robot's circular marker, providing

a more reliable way of dealing with partial occlusion or with the robot being partially out of the camera's field of view.

A method of generating accurate partial steps can be implemented on the robot, by using vision based sensor fusion to estimate valve actuation times. The integration of acceleration information in order to estimate fractional-step motions is still being pursuit.

5 Conclusions

The visual localisation system (both for position and heading extraction), was demonstrated with the climber robot. Its easy-to-setup and low-cost properties make it a useful application for robot testing, and using several cameras a very large workspace can covered.

Due to its non-intrusive characteristics, this type of system is particularly useful as a sensorial method for following paths and avoiding obstacles, providing zero interference with other sensors used by the robot (like IR and ultra-sound).

External vision and odometry sensor fusion reduced the long term dead-reckoning error of a climbing robot. The system provided enough performance (namely in terms of precision and response-time) for the envisaged applications.

References

1. Rowe A, Rosenberg C and Nourbakhsh I (2002) A Low Cost Embedded Color Vision System. In Proc. of Intl. Conf. on Intelligent Robots and Systems (IROS) 2002
2. Bénallal M, Meunier J (2003) A simple algorithm for object location from a single image without camera calibration (Michel Chasles Theorem). In Proc. of Intl. Conf. on Computer Science and its Applications (ICCSA) 2003
3. Bar-Shalom Y, Xiao-Rong Li (1993) Estimation and Tracking – Principles, Techniques and Software. Artech House
4. Kelly A (1994) A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles. CMU-RI-TR-94-19, The Robotics Institute, Carnegie Mellon University
5. Martins A, Marques L, de Almeida A T (2004) Robust Localisation System for Robot Communities, in Proc. of CONTROLO, 2004.
6. Rachkov M, Marques L, de Almeida A T (2002) Climbing robot for porous and rough surfaces, in Proc. Intl. Conf. on Climbing and Walking Robots (CLAWAR), 2002.

Roboclimber: Proposal for Online Gait Planning

M. Moronti, M. Sanguineti, M. Zoppi, and R. Molfino

Department of Mechanics and Machine Design, University of Genova, via Opera Pia 15A, 16145, Genova, Italia
{zoppi,molfino}@dimec.unige.it

1 Introduction

Roboclimber [1] is an heavy duty, four-legged, two-roped walking robot conceived and designed for consolidation of almost vertical and irregular rocky walls (Fig. 1).

Landslides and mass movements represent a serious danger both in terms of fatalities and economic losses. Today, the standard technique of intervention consists of setting up reinforced piles, anchored to deeper stable rock layers. These piles compress and block the outer instable layers so making safe the site. The set up of each pile consists of 3 main operations: drilling of an about 90 mm diameter deep hole, long up to 30 m; set up of a steel reinforcing inside the hole, usually a system of steel ropes or bars; filling up of the hole with grouting mortar. A geological map of the intervention provides the location of each hole on the rocky wall together with its direction and deepness. Finally, steel nets are eventually disposed to protect from smallest falling rocks.

The set up of the piles can be performed with the help of vehicles with telescopic arms, in the case of smaller walls. Today, larger surfaces are equipped with scaffolds and the drilling and setting up operations are performed by human workers, with a high cost and the risks for the safety of the workers. Roboclimber represents an effective alternative to these traditional technologies.

The robot can walk on irregular surfaces up to 30 degrees sloping and can climb almost vertical walls by making use of two ropes, suitably anchored to the top of the wall [2]. The legs have a quasi-cartesian architecture: two orthogonal prismatic mobilities and one hip rotation. The ropes are pulled by two winches of the type *tirfor*, with a particular mechanical architecture allowing "hand-to-hand" recovery of the rope. The tirfors have to be operated in co-operation with the legs. The drilling operations are performed by a

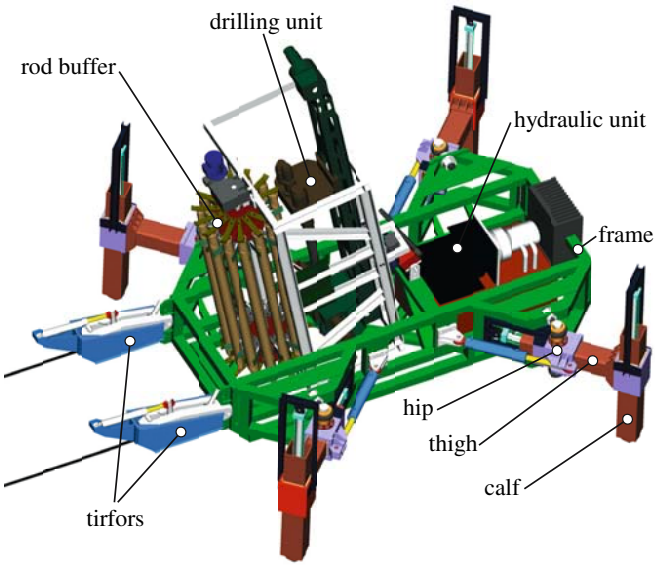


Fig. 1. Digital mock-up of Roboclimber

middle size drilling unit placed in the middle of the robot. The drilling rods are stored in a rotating buffer in front of the drilling unit. A robotic arm brings the rods from the buffer and delivers them to the drilling unit, which screws up automatically each rod on the drilling line. The hammer is down the hole (immediately before the cutting head), so the drilling unit has to push the drilling line with a constant force of about 10000 N and applies a torque of about 3000 Nm. Electrical power is provided through an umbilical together with the compressed air required by the drilling hammer. The legs and the tirfors have hydraulic actuators powered by an on-board electrical hydraulic unit.

The robot is remotely operated. Four moving options are available: *backward, forward, left, right*, while the drilling operations are performed autonomously under remote surveillance. The overall mass of the machine is about 3000 kg. This high mass and the task of the robot suggest the choice of a quasi static gait. The control system is asked for planning online the gait. The present paper describes an effective planning technique and presents the simulation results obtained so far. Offline information about the wall region reachable by the robot, the region where the robot can perform the drilling operations, and the best location of the anchorage points is also required to plan the intervention. This information can be obtained off-line by means of the planning algorithm. Explanatory results are shown.

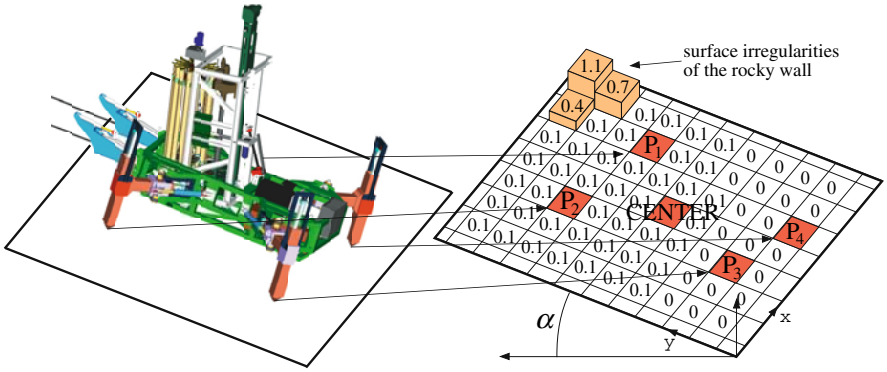


Fig. 2. Schematic of Roboclimber on the wall surface

2 Gait Planning

Relevant parameters for the gait planning are [3]:

- the strokes of the prismatic joints of the legs and the maximum actuation forces available;
- the span of the hip rotation of the legs and the maximum actuation moment available;
- the robot geometry and the actual robot configuration (coordinates of four points representing the hips of the legs and coordinate of two points representing the points of connection of the ropes to the robot).

The average rocky wall surface is divided in identical square cells. Bumps and sockets are modelled by associating an amount of extrusion to each cell (Fig. 2). The size of the cells is chosen accordingly to the distribution of the irregularities on the rocky wall, in order to get a meaningful discretization. The controllable parts of the robot are the four feet P_1, P_2, P_3, P_4 and the geometric centre of the frame. Each controllable part is supposed to correspond to one cell of the grid. The state of the robot is described by a set of five vectors $P_i(x_i, y_i, z_i), i = 1, 2, 3, 4$, and $CENTER(x_c, y_c, z_c)$, one for each controllable part, where x_i, y_i and $z_i, i = 1, 2, 3, 4, c$, are, respectively, the two coordinates and the extrusion of the cell containing the i-th controllable part.

Due to the presence of four feet and two ropes, the equilibrium of the robot cannot be assessed with a rigid body modelling [4]. It is necessary to consider the compliance of the ground and the distributed and concentrated compliances of the robot (joints, links, actuators, frame). We refer to the model and the algorithm proposed in [5] to assess the static equilibrium of robots with legs and ropes.

The working environment and the operative conditions are described by:

- the average slant wall surface angle α;
- the height and the width of the considered wall region;

- the coordinates of the anchorage points;
- the side of the cells;
- the matrix of the extrusions associated to the grid cells;
- the parameters expressing the compliance of the ground and of the robot;
- the maximal tangential force resisted by the ground at the interface ground-feet (scratching).

The goal of the gait planning [6] is to select a sequence of static equilibrium configurations of the robot (steps) in order to translate it of a number of cells corresponding to its side length in one of the considered moving directions (*backward, forward, left, right*). To iterate the gait, the last robot configuration of the sequence is the same as the start one, translated of the side length in the selected direction. The gait is planned using state exploration. To fast the process, the sequences of actions that are explored involve cells in a square subset surrounding the start cell. At each step, four actions can modify the state of the robot: *backward, forward, left, right*. Each action can act on any controllable part and induces a displacement of one cell in the corresponding direction (down, up, left, right), e.g., $act(P_1, up)$.

Several AI (*artificial intelligence*) techniques was considered for state exploration. After an accurate comparison of the results, a *search algorithm* appeared the best option, due to the effective plans computed and to the low running time, that would allow online use of the algorithm in the control system of Roboclimber. The following section presents the planning technique and part of the preliminary results obtained.

3 The Search Algorithm

The search algorithm used is A^* [7], an informed search algorithm based on best-first logic. The function principle of the search algorithm is to progressively generate and explore a tree of feasible states obtained step by step from the start state. At each step, any possible feasible action is tested and, depending on the value of a state cost, one of the tested actions is selected and the corresponding new state is appended to the path.

The state cost used is the sum of two subcosts:

- *backward*: the sum of the costs associated to the actions executed to reach the current state from the start state. For Roboclimber, the considered cost for all actions is 1, so the backward subcost is equal to the sum of all the actions that have been necessary to reach the current state from the start state;
- *forward*: an estimate of the number of actions required to reach the goal state from the current state.

So defined, the whole cost of the current state is the sum of the estimates of the two distances of the current state from the starting and the goal states.

The distances are computed by subtracting the coordinates of the cells corresponding to the current state from the coordinates of the cells of the starting and goal states (Manhattan distance).

The heuristics that steers the algorithm refers to the forward cost of the current state. The action that the heuristics suggests is the one leading to the state with minimum forward cost between the reachable ones, no matter if static equilibrium holds. It is an admissible heuristics [8] since every path, composed of static equilibrium states only, is longer or equal to the minimum path that would be obtained following strictly the heuristics.

At every step, first the current state is expanded by generating all its children states. The children states are explored starting from the one with minimum cost and following the suggestions of the heuristics. Every suggestion is accepted by the search algorithm only if it leads to a state of static equilibrium. Else, if the static equilibrium test returns false, the suggestion is judged unfeasible and the following child state is examined, and so on. If all children states are unfeasible, the algorithm performs backtracking. The solution found by A* is the one with the lowest number of actions [7].

The search algorithm was implemented in *Prolog* language and accurately tested. The running time varies in a range between 0.1 s for the simplest test cases to a maximum of 1.7 s for the more complex cases tested. This proves the effectiveness of a search algorithm in the case of gait planning of walking robots with a cumbersome static equilibrium problem.

4 Preliminary Results

The described search algorithm can be used:

- to plan online the gait of Roboclimber following the moving options selected by the remote operator;
- to generate offline maps of the wall regions that can be reached by Roboclimber with a quasi static gate (reachable regions);
- to plan the repositioning operations of the anchorage points by means of maps of the reachable regions for any given set of anchorage points;
- to map the wall regions where drilling can be performed without introducing additional ropes to fix the robot to the wall (free drilling regions).

To obtain the map of the reachable wall regions on a wall with a given slope, the locations of the anchorage points are selected. Then, any starting location on the wall is chosen, provided that it is of static equilibrium. The gait planner is launched using the chosen starting cell. Each wall cell, one by one, is taken as the goal cell for the gait planner. If a path is found connecting the goal cell to the starting cell, that goal cell is included in the map of the reachable wall regions. The same mapping process is repeated, if necessary, for different starting cells.

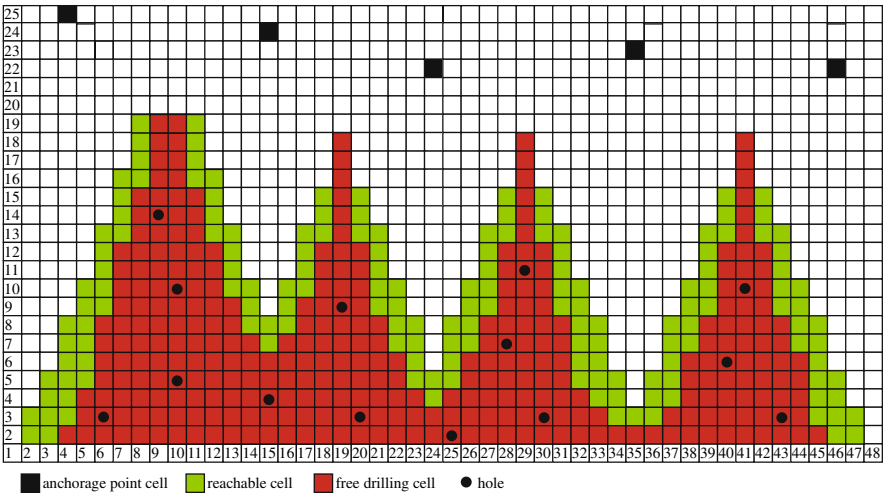


Fig. 3. Explanatory map of reachable and drilling region on a flat 60° sloping wall

Figure 3 shows an explanatory map of the reachable wall region on a flat 60° sloping wall, for four relocations of the two anchorage points. The grid is composed of macrocells gathering a number of cells equal to the onplant area of the robot. The darker grey macrocells represent the free drilling regions. In such regions, Roboclimber keeps in static equilibrium also under the relevant drilling forces and torques. To perform drilling operations out of the free drilling regions it is necessary to introduce additional constraints, e.g., additional ropes fixed to rock nails positioned close to the robot. This is time consuming and dangerous, since the nailing operations should be executed manually by human workers. For this reasons, it is preferable to plan the intervention in order that all the holes are inside free drilling regions. Also the order of execution of the holes is planned off-line in order to minimize the overall cost of the intervention.

5 Conclusions

The paper presents a strategy for the gait planning of Roboclimber, a heavy duty robot designed for consolidation of rocky walls. The gait of the robot is complex since the winches of the two ropes have to be operated in coordination with the four legs. The robot is remotely operated from ground and the remote controller have to provide simple moving options, such as backward, forward, left, right, in order to make its use more easy. Moreover, the robot must always keep in static equilibrium. For these reasons, it is necessary to introduce in the Roboclimber: proposal for online gait planning 7 control system a gait planning algorithm able to generate the quasi-static gait in response to the

selected moving options. An additional strict constraint is that such planning algorithm has to be very fast in order to be used online. The dynamics of the robot is slow, due to the high overall mass and to the task, anyway planning times of few seconds have to be considered. The paper presents part of the preliminary results obtained using a search algorithm. This choice appears satisfactory for many reasons: the running time, also in the case of complex plans, is less than 2 s and the same algorithm can be used for several offine operations regarding the organisation of the consolidation intervention, such as generation of maps of the reachable regions of the wall and planning of the set up operations of the rocky wall.

Due to the generality of the approach, the results obtained can be of general interest for the online gait planning of walking robots with ropes.

Acknowledgments

The Roboclimber project is funded by the European Commission under the V-FP (CRAFT Contract N G1ST-CT-2002-5016). The European Community and the Roboclimber Partnership are hereby kindly acknowledged.

References

1. The roboclimber partnership. Roboclimber. In *1st Int. Workshop on Advances in service Robotics ASER03*, Bardolino, Italy, March 13–15 2003.
2. G. Acaccia, L.E. Bruzzone, R.C. Michelini, R.M. Mol.no, and R.P. Razzoli. A tethered climbing robot for .rming up high-steepness rocky walls. In *6th IAS Int. Conf. on Intelligent Autonomous Systems*, pp. 307–312, Venice, Italy, 2000.
3. M. Moronti and M. Sanguineti. *ROBOCLIMBER: analisi dell'equilibrio e pianificazione del movimento*. Master thesis, Univesity of Genova, Genova, Italy, March 2004.
4. S. Hirose, K. Yoneda, and H. Tsukagoshi. Titan vii: quadruped walking and manipulating robot on a steep slope. In *Int. Conf. on Robotics and Automation*, Albuquerque, NM, April.
5. M. Zoppi, S. Sgarbi, R.M. Molfi no, and L. Bruzzone. Equilibrium analysis of quasi-static, multi-roped walking robots. In *Int. Conf. CLAWAR03*, pp. 259–266, Catania, Italy, September, 17–19 2003.
6. J. Estremera, E. Garcia, and I. Gonzales de Santos. A continous free crab gait for quadruped robots on irregular terrain. In I*nt. Conf. CLAWAR03*, pp. 585–592, Catania, Italy, September, 17–19 2003.
7. S. Russel and P. Norvig. *Artificial Intelligence: a modern approach*. Prentice Hall Series in Artificial Intelligence.
8. Richard E. Korf. Recent progress in the design and analysis of admissible heuristic functions. In *7th Nat. Conf. on Artificial Intelligence and 12th Conf. on Innovative Applications of Artificial Intelligence*, pp. 1165–1170. AAAI Press/The MIT Press, 2000.

Adhesion Control for the Alicia^3 Climbing Robot

D. Longo and G. Muscato

Dipartimento di Ingegneria Elettrica Elettronica e dei Sistemi, Università degli Studi di Catania, viale A. Doria 6, 95125 Catania Italy
dlongo@diees.unict.it, gmuscato@diees.unict.it

Abstract. Climbing robots are useful devices that can be adopted in a variety of applications like maintenance, building, inspection and safety in the process and construction industries.

The main target of the Alicia^3 robot is to inspect non porous vertical wall with any regard for the material of the wall. To meet this target, a pneumatic-like adhesion for the system has been selected. Also the system can move over the surface with a suitable velocity by means of two DC motors and overcome some obstacle thanks to a special cup sealing.

This adhesion technology requires a suitable controller to improve system reliability. This is because small obstacles passing under the cup and wall irregularity can vary the value of the internal pressure of the cup putting the robot in some anomalous working conditions. The methodologies used for deriving an accurate system model and controller will be explained and some result will be presented in this work.

1 Introduction

Climbing robots can be used to inspect vertical walls to search for potential damage or problems on external or internal surface of aboveground/ underground petrochemical storage tanks, concrete walls and metallic structures [1–4]. By using this system as carrier, it will be possible to conduct a number of NDI over the wall by carrying suitable instrumentation [5, 6].

The main application of the proposed system is the automatic inspection of the external surface of aboveground petrochemical storage tanks where it is very important to perform periodic inspections (rate of corrosion, risk of air or water pollution) at different rates, as standardized by the American Petroleum Institute [7]. The system can be also adopted to inspect concrete dams.

While these kinds of inspections are important to prevent ecological disasters and risks for the people working around the plant, these are very expensive because scaffolding is often required and can be very dangerous

(a) (b) (c)

Fig. 1. Typical operating environment and the Alicia^3 robot

for technicians that have to perform these inspections. Moreover, for safety reasons, plant operations must be stopped and the tank must be emptied, cleaned and ventilated when human operators are conducting inspections. In Fig. 1(a) and 1(b) typical environments for climbing robots are shown. Figure 1c shows the Alicia^3 robot prototype while attached to a concrete wall during a system test.

2 System Description

The Alicia II system (the basic module for the Alicia^3 system) is mainly composed by a cup, an aspirator, two actuated wheels that use two DC motors with encoders and gearboxes and four passive steel balls with clearance to guarantee plain contact of the cup to the wall. The cup can slide over a wall by means of a special sealing that allows maintaining a suitable vacuum inside the cup and at the same time creating the right amount of friction with respect system weight and a range of a target wall kind.

The structure of the Alicia II module, shown in Fig. 2, currently comprises three concentric PVC rings held together by an aluminums disc. The bigger ring and the aluminums disc have a diameter of 30 cm. The sealing system is allocated in the first two external rings. Both the two rings and the sealing are

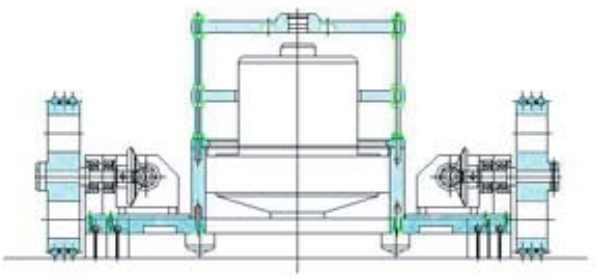


Fig. 2. Structure of the Alicia II module

designed to be easily replaceable, as they wear off while the robot is running. Moreover the sealing allows the robot passing over small obstacles (about 1 cm height) like screws or welding traces. The third ring (the smallest one) is used as a base for a cylinder in which a centrifugal air aspirator and its electrical motor are mounted. The aspirator is used to depressurize the cup formed by the rings and the sealing, so the whole robot can adhere to the wall like a standard suction cup.

The motor/aspirator set is very robust and is capable of working in harsh environments. The total weight of the module is 4 Kg.

The Alicia[3] robot is made with the three modules linked together by means of two rods and a special rotational joint. By using two pneumatic pistons it is possible to rise and to lower each module to overcome obstacles. Each module can be raised 15 cm with respect to the wall, so obstacles that are 10–12 cm height, can be easily overcame. The system is designed to be able to stay attached using only two cups while the third, any of the three, is raised up. The total weight of the system is about 20 Kg.

3 Electro-Pneumatic System Model

By using this kind of movement and sealing method, it is possible, due to unexpected small obstacles on the surface, to have some air leakage in the cup. This leakage can cause the internal negative pressure to rise up and in this situation the robot could fall down. On the other side if the internal pressure is too low (high Δp), a very big normal force is applied to the system. As a consequence, the friction can increase in such a way to not allow robot movements. This problem can be solved by introducing a control loop to regulate the pressure inside the chamber to a suitable value to sustain the system. The considered open loop system and the most easily accessible system variables has been schematized in Fig. 3; in this scheme the first block includes the electrical and the mechanical subsystem and the second block includes the pneumatic subsystem. The used variables are the *Motor voltage reference* (the input signal that fixes the motor power) and the *Vacuum level* (the negative pressure inside the chamber).

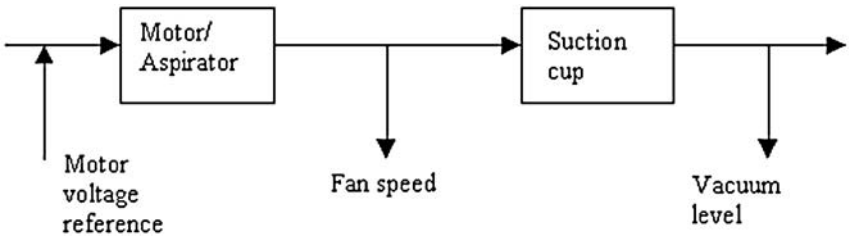


Fig. 3. The open loop system considered

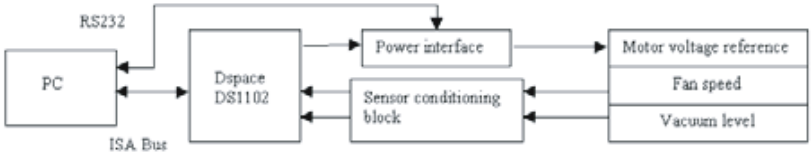


Fig. 4. I/O variable acquisition scheme

Since it is very difficult to have a reliable analytical model of that system, because of the big number of parameters involved, it has been decided to identify a black box dynamic model of the system by using input/output measurements. This model was designed with two purposes: to compute a suitable control strategy and to implement a simulator for tuning the control parameters.

An experimental setup was realized, as represented in Fig. 4, by using the DS1102 DSP board from Dspace in order to generate and acquire the input/output variables. Since the aspirator is actuated by an AC motor, a power interface has been realized in order to translate in power the reference signal for the motor coming from a DAC channel of the DS1102 board. The output system variable has been measured with a piezoresistive pressure sensor with a suitable electronic conditioning block and acquired with one analog input of the DS1102. The software running on the DSpace DSP board, in this first phase simply generates an exciting motor voltage reference signal (pseudo random, ramp or step signals) and acquires the two analog inputs with a sampling time of 0.1 s, storing the data in its internal SRAM.

Typical Input/Output measurements are represented in Fig. 5 and Fig. 6. In order to obtain better results in system modeling, the relationship between Input and Output needs to be considered as non-linear. A NARX model has

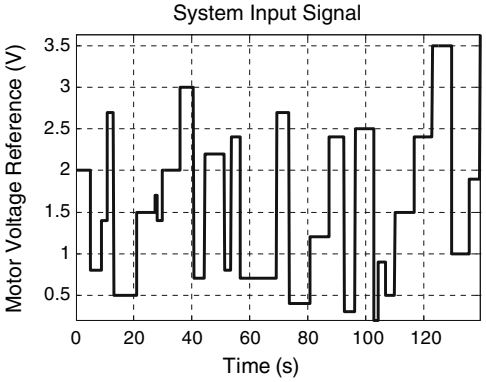


Fig. 5. Typical Input signals

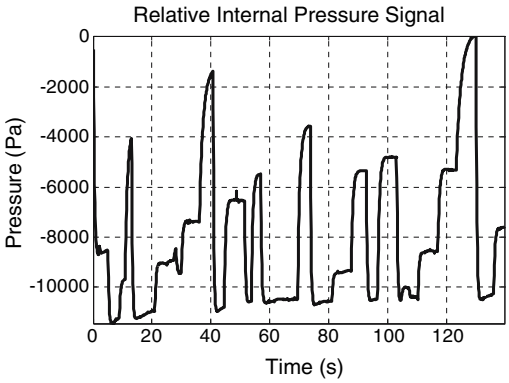


Fig. 6. Typical Output signals

been used is in the form of (1), where f is a non linear function [8, 9].

$$y(k) = f(u(k), u(k-1), \ldots; y(k-1), y(k-2), \ldots) \tag{1}$$

To implement this kind of non-linearity, some trials have been done using Neuro-Fuzzy and Artificial Neural Network (ANN) methodologies. Once that model has been trained to a suitable mean square error, it has been simulated giving it as input the real input measurement only (infinite step predictor) [8]. So (1) can be modified in order to obtain (2).

$$\tilde{y}(k) = f(u(k), u(k-1), \ldots; \tilde{y}(k-1), \tilde{y}(k-2), \ldots) \tag{2}$$

In (2), $\widetilde{y}$ is the estimated system output. In order to compare the simulation results, a number of descriptor has been defined and used. Among these are mean error, quadratic mean error and some correlation indexes. A first set of simulation for both methodologies has been done to find out the best I/O regression terms choice.

3.1 Neuro-Fuzzy Identification

Using this kind of methodology, the best model structure was found to be in the form of (3).

$$y(t) = f(u(t), y(t-1)) \tag{3}$$

Once the best model structure has been found, some trials have been performed modifying the number of membership functions. The best results, comparing the indexes described above, have been obtained with 3 functions and in Fig. 7 the simulation results has been reported. The structure of the Neuro-Fuzzy model is the ANFIS-Sugeno [10].

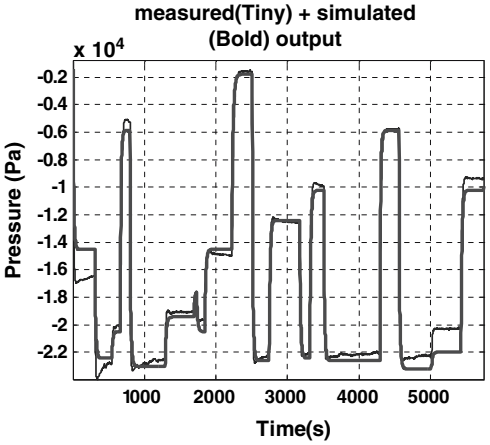


Fig. 7. System identification: Neuro-Fuzzy model with 3 membership function

3.2 ANN Identification

Using this kind of methodology, the best model structure was found to be in the form of (4).

$$y(t) = f(u(t), u(t-1), u(t-2), y(t-1), y(t-2)) \tag{4}$$

A single layer perceptron network has been used. The training algorithm is the standard Levenberg–Marquardt.

Once the best model structure has been found, some trials have been performed modifying the number of hidden neurons. The best results, comparing the indexes described above, have been obtained with 7 hidden neurons and in Fig. 8 the simulation results has been reported.

From a comparison between the two models and their related indexes, it can be seen that the Neuro-Fuzzy model has best approximation performance and use less input information. In the next section, this model will be used as system emulator to tune and test the required regulator.

4 Pressure Control Algorithm

Once a system model has been obtained, a closed loop configuration like that in Fig. 9, has been considered.

The target of the control algorithm is to regulate the internal vacuum level to a suitable value (from some trials, it was fixed to about 10 kPa) to sustain the whole system and its payload; the maximum steady state error allowed was fixed to less than 200 Pa. Moreover the time constant of the real system (about 10 s) has to be considered. A first simulation trial has been done with a fuzzy controller while during a second trial a PID controller has been tuned

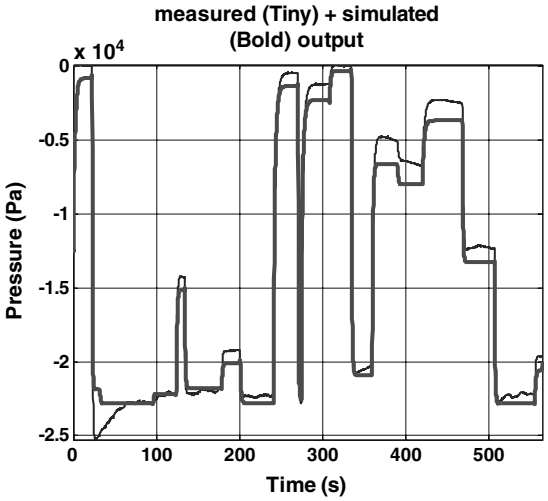


Fig. 8. System identification: ANN model with 7 hidden neurons

over the system emulator to meet the controller target. All these simulations have been performed by using Simulink from Mathworks.

4.1 Fuzzy Controller

During this simulation, a fuzzy controller that uses as input only the system error has been used. This controller has three membership function (triangular and trapezoidal) and three output crisp membership functions.

The reference was set to 10 kPa and the noise signal on the pressure level is a series of steps. In Fig. 10 a plot of the noise, reference and closed loop pressure signal is represented [11].

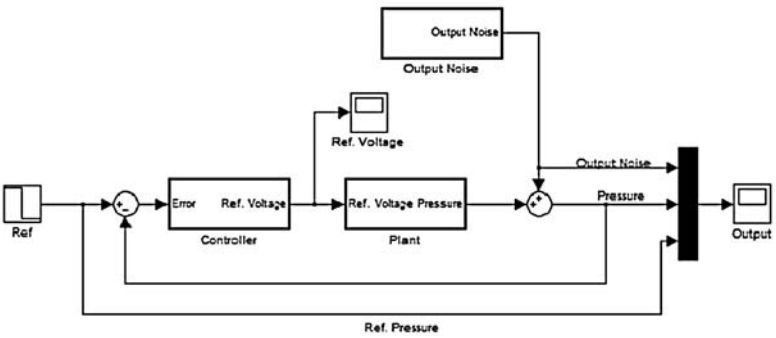


Fig. 9. Closed loop scheme

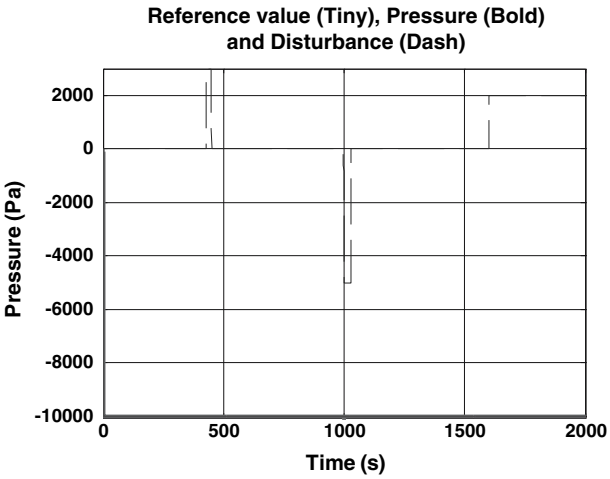


Fig. 10. Closed loop simulation (Fuzzy controller)

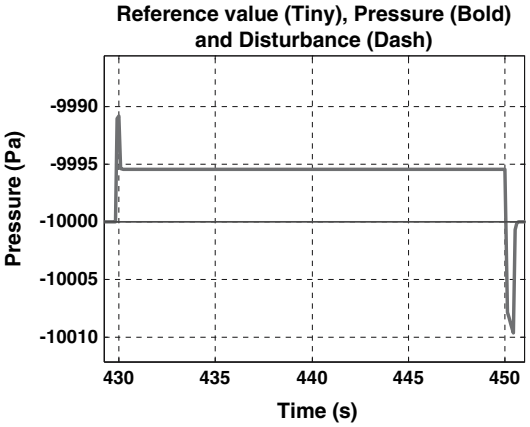


Fig. 11. Closed loop simulation (Fuzzy controller): a detail

In Fig. 11 a detail of the simulation is represented. It is possible to see that a disturbance of about 3 kPa on the internal pressure is compensated by the controller with only a very little error. Also the transient is very short with respect the system dynamics.

4.2 PID Controller

A second simulation has been done tuning a PID controller over the Neuro-Fuzzy system emulator. As the system model is non-linear, trial and error technique has been used. The controller has been tested in the same condition

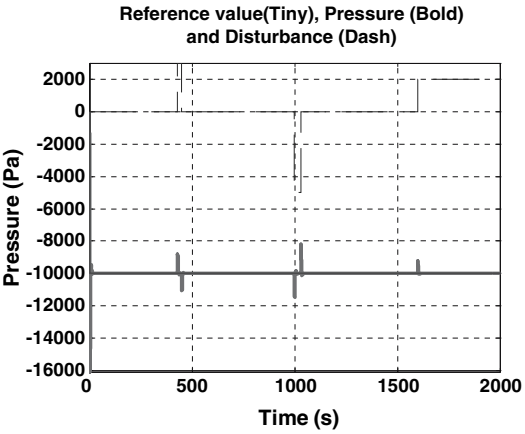


Fig. 12. Closed loop simulation (PID controller)

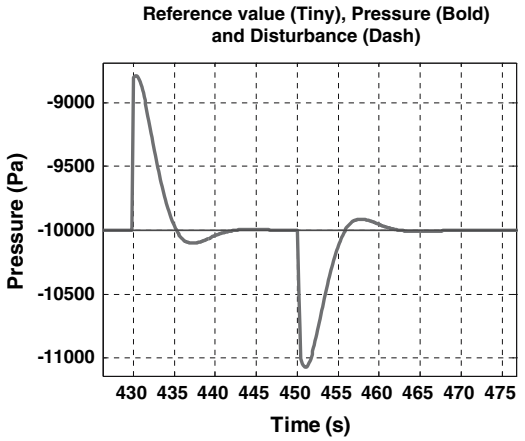


Fig. 13. Closed loop simulation (PID controller): a detail

of the fuzzy controller. From the Fig. 12 it is possible to see that now the closed loop system has little more overshooting (see Fig. 13 for detail) but the same steady state error. It has to be noted that overshooting is higher that the maximum error allowed but is faster with respect the system time constant.

5 Conclusion

In this work the Alicia^3 climbing robot was presented. Due to its special adhesion mechanism, a controller for the vacuum level inside the cup is required. First of all, a system emulator has been designed by using black box identification methodologies. Among all the performed trial, Artificial Neural

Networks and Neuro-Fuzzy are the two best models found and the Neuro-Fuzzy one has been selected as system emulator. A set of indexes has been introduced in order to make a comparison and to select the best system model. Once a system emulator has been become available, some Simulink simulations have been performed in order to tune a controller. In that case a Fuzzy and a PID controller have been compared. Between the two, the Fuzzy controller works better than the PID but this is much simpler in its implementation and its performances are not so worst; in any case, it is compatibles with system dynamics.

References

1. Wang Y, Zhao X, Xu D (1999) Design of a wall cleaning robot with a single suction cup. Robot Research Institute of Harbin Institute of Technology, China. Proceedings of 2nd International Conference on Climbing and Walking Robots, p. 405
2. White TS, Cooke DS (2000) Robosense – Robotic delivery of sensors for seismic risk assessment. Portsmouth Technology Consultant Limited, UK Proceedings of 3rd International Conference on Climbing and Walking Robots, p. 847
3. Wang Y, Zhao X, Xu D (2000) The study and application of wall-climbing robot for cleaning. Robot Research Institute of Harbin Institute of Technology, China. Proceedings of 3rd International Conference on Climbing and Walking Robot, p. 789
4. La Rosa G, Messina M, Muscato G, Sinatra R (2002) A low cost lightweight climbing robot for the inspection of vertical surfaces. Mechatronics 12, 71–96, Pergamon
5. Weise F, Kohnen J, Wiggenhauser H, Hillenbrand C, Berns K (2001) Non-destructive sensors for inspection of concrete structures with a climbing robot. In: Proceedings of the 4th International Conference on Climbing and Walking Robots CLAWAR 2001, Karlsruhe, Germany, 24-26 September 2001, Professional Engineering Publishing, pp. 945–952
6. Schraft RD, Simons F, Schafer T, Keil W, Anderson S (2003) Concept of a low-cost, window-cleaning robot. In: Proceedings of the 6th International Conference on Climbing and Walking Robots CLAWAR 2003, Catania (Italy), 17-19 September 2003, Professional Engineering Publishing, pp. 785–792
7. American Petroleum Institute (2005) Tank Inspection, Repair, Alteration and Reconstruction. Standard 653, January 1992, API, 1220, L. St. NW, Washington, DC
8. Narendra KS, Parthasarathy K (1990) Identification and control of dynamical system using neural networks. IEEE Transaction on Neural Network, Vol. 1 no. 1, pp. 4–27
9. Chen S, Billings SA, Cowant CFN, Grant PM (1990) Practical identification of NARMAX models using radial basis functions. Int. J. Control, Vol. 52, no. 6, pp. 1327–1350

10. Jang, J-SR (1993) ANFIS: Adaptive-Network-based Fuzzy Inference Systems. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 665–685
11. Jang J-SR, Sun CT (1995) Neuro-fuzzy modeling and control. In: Proceedings of the IEEE, March 1995

Part IX

Applications

In Service Inspection Robotized Tool for Tanks Filled with Hazardous Liquids – Robtank Inspec

A. Correia Cruz and M. Silva Ribeiro

Instituto de Soldadura e Qualidade. Taguspark, Portugal
accruz@isq.pt

1 Introduction

Large storage tanks are prone to degradation by corrosion mechanisms, mainly at the bottom floor plates where as a consequence, leaks tend to occur. Leakage from corroded floor plates is a major environmental, safety and economic hazard in the petrochemical and oil industry. Leakages are prevented by industry recognized inspection procedures, that include emptying the tanks for internal inspections, what implies large periods of equipment unavailability, with expensive, time consuming and hazardous cleaning processes, which involve high risk exposure of workers to chemicals. The purpose of the Robtank Inspec system is to make available a vehicle able to perform in-service inspection of storage tanks by using ultrasonic systems and avoiding the inconvenience of emptying the tanks.

2 The Problem

Storage tank owners and users want to prevent any sort of leakage due to the high risks related to environment, safety and economic impacts. The inspection procedures in practice consider periodically external and internal inspections in order to assess the tank condition and establish the adequate inspection periods. However, available internal inspection techniques involve emptying and cleaning the tanks, which operations present several disadvantages:

- are expensive;
- time consuming, as between the beginning of the operations and the commissioning of the tank it could take large periods depending on the dimension of the tank, stored product, location and work conditions;

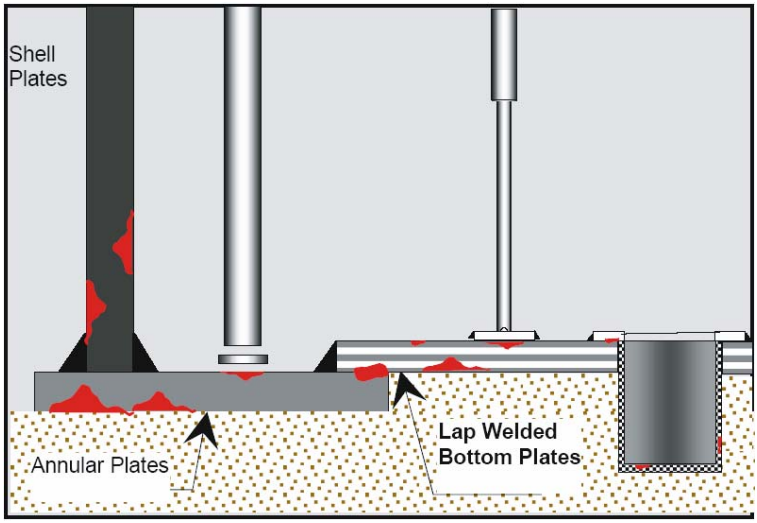


Fig. 1. Typical problems at tank bottoms

- high risk exposure of workers to chemicals during the cleaning operation, even when proper individual protection equipment\is used;
- large unavailability periods for the equipment;
- it is an hazardous process frequently involving transport of dangerous products to alternative sites.
- inspections tasks involve hard working conditions requiring several safety precautions.

The typical corrosion areas in storage tanks to be looked during an internal inspection are shown in Fig. 1. Robtank Inspec system aims to perform such inspection, covering the shown critical areas, including e.g. lap points and the area below the level meter tube. However the system is not able to inspect the well pit area.

Figure 2 shows typical examples of corrosion damage possible to found in storage tanks bottom plates.

3 Conventional Inspection Procedures

Conventional periodical inspection procedures on storage tanks follows international standards, e.g. API (American Petroleum Institute). Depending on the owner or/and on the country, the applicable standards could be more severe than the international ones. Conventionally, internal bottom plates inspection is performed based on a standardized periodicity, or based on corrosion rate date. In the present, the international standards applicable to the

Fig. 2. Example of bottom plate pitting and plate corrosion

construction, maintenance and inspection of this type of tanks are the following:

- API Recommended Practice 575
- API STANDARD 653
- API STANDARD 650
- API STANDARD 620

The conventional inspection processes could be divided in intrusive and non-intrusive processes. Intrusive processes means that to perform the inspection, the inspector and operators need to enter in the interior of the tank. Non-intrusive processes means that the inspection could be made from the exterior, without emptying the tank. Examples of non-intrusive processes are the external visual inspection, acoustic emission and chemical analysis of the surrounding and under the tank soil, that only can give a qualitative assessment of the internal tank condition. Examples of intrusive processes are the MFL (Magnetic Flux Leakage) [2], X-rays, thickness measurements using ultrasounds. Current external inspection techniques allow the detection of leaking tanks only, but do not give information about the extent of corrosion pitting. That's why the internal inspection is necessary on the majority of the cases, to have an evaluation of the condition of the bottom plates. Figure 3 shows part of a bottom tank diagram plate distribution, indicating the locations to apply the different inspection methods.

As mentioned before, the inspection tasks inside the tank usually involve hard working conditions, especially because of the high temperatures as the operators must use the proper individual protection equipment. Following figures show two pictures of inspection works performed inside a storage tank.

Some other innovative techniques indicated in the references are use in special cases [1, 3].

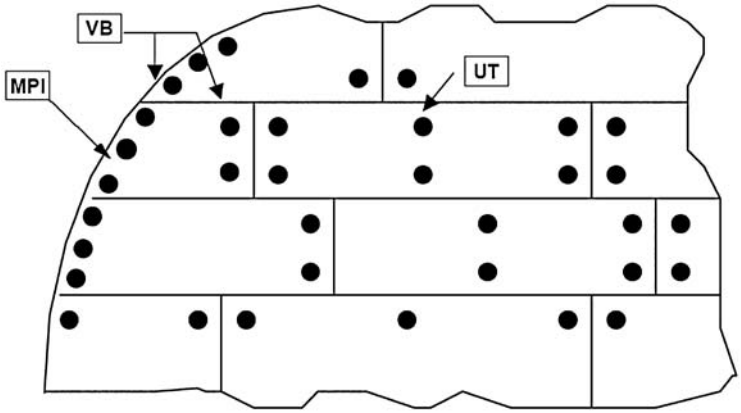


Fig. 3. Bottom plates diagram

Fig. 4. Performing internal conventional MFL and UT measurements

4 Inspection Environment: Storage Tank Types

Storage tanks on petrochemical and oil industry are mainly vertical cylindrical tanks. Figure 5 shows some typical storage tanks. They are classified on the basis of their roofs with the majority either having floating roofs or fixed roofs and also with both fixed and floating roofs. Floating roofs of large tanks can be 15–20 m above ground and the diameter can be up to 100 m. These tanks are mainly used to store crude oil.

The bottom of these tanks is usually located at ground level and is constructed of lap-welded carbon steel plates, normally 6-9 mm in thickness. The floor plates can be distorted. In large floating roof tanks, access to the tank is through a single man-hole of minimum diameter 300 mm diameter riser pipes. Normally roof legs are inserted through the risers and support the roof when the tank is empty. Smaller fixed roof tanks have the same type of floor construction as the large floating roof tanks. They have only one or two accesses hole 300–600 mm in diameter on the roof for the deployment of any

Fig. 5. Typical hazardous products storage tanks

Robotic device. Petroleum products in these types of tanks are usually clean containing light liquid products e.g. kerosene, gasoline, jet-fuel, other light crude derivatives. Access to loaded storage tanks is usually possible through manholes in the tank roofs. Process industries store different products, some of them in fiber glass tanks, others in concrete tanks and some others in stainless steel tanks.

5 Robotized Solution for in Service Tank Inspection

The main objective of RobTank Inspec is to provide an in service in pection system to access the condition of tank floor and wall plates that are subjected to degradation by corrosion, in order to prevent leaks. The new inspection system will avoid out of service inspection practices that require emptying and cleaning the tanks before inspection and involve environmental risks. However, the use of such an inspection system is not an easy task as the interior of tanks is not a free space and depending of the tank type and stored product, several auxiliary equipment exist and impose constraints to the vehicle movement for the inspection tasks. The main difficulties are the following:

- sludge deposits at the bottom of crude oil tanks can be in the order of a few meters, this problem can be solved by product circulation and filtering systems that remove sludge and can reduce the bottom deposit down to less than 50 mm;
- the remaining sludge can cause sliperage and existing different types of debris can cause difficulties to the vehicle movement;
- internal furniture like drain wells and pipes, roof supporting legs or columns and gauging equipments will be obstacles to the free movement of the vehicle and its umbilical;
- tank floor plates distortion such as rippling, bulges, depressions due to soil settlement, etc. and may be corroded from both underside and topside.

Robotic methods of non-destructive testing and automated and robotic solutions for inspection in hazardous environments are all of strategic importance to the European Union. So far, the use of robotics in industrial applications is a field dominated by Japan and USA. The future growth prospects for Europe in this field are important.

A system that allows an extensive evaluation of the condition of the storage tank without emptying it, will bring significant savings as referred before. Inspecting the bottom plates with a robotized system that could operate immersed on the stored product, in complement to the conventional non intrusive methodologies, is the aim of the RobTank Inspec project and of the robotized system presented. Its first objective is to provide an in-service inspection system to access the condition of tank floor and wall plates subjected to corrosion and to prevent leaks. The project aims at different and a wider range of applications besides the oil and petrochemical industries. Other industrial sectors use fuel storage tanks and more stringent environment requirements regarding hazardous liquids leakage, extends the industrial application sectors for the inspection system under development, to smaller light petroleum tanks, chemical tanks, and others.

The capability to enter through relatively small diameter openings (300 mm diameter) on the roof of the tanks, together with the ability to climb tank walls are advantages regarding RobTank Inspec competitors that are mainly tank floor inspection vehicles. The vehicle will operate under hazardous environments. The ability for the vehicle to climb by generating an adhesion against the walls independently of any usual material, allows the possibility to inspect nonmagnetic materials used in some products that can be found in storage tanks in the chemical industry.

The concept of the system is represented on Fig. 6. The vehicle is operated from an exterior operation post, located in a safe location. It carry sensors

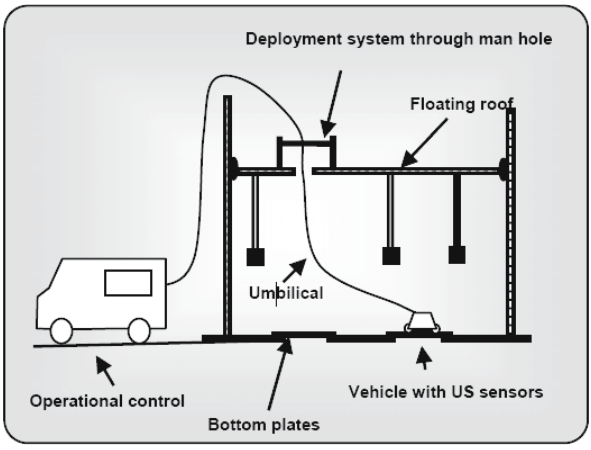


Fig. 6. Robtank Inspec system concept

to perform tank floor and wall inspections, using ultrasonic UT techniques, covering American Petroleum Institute (API) standards requirements for tank inspections. The inspection vehicle will be resistant to hostile environments and will comply with the necessary safety requirements. An umbilical system is used to deploy and retrieve the vehicle, to provide the necessary power and for the communication between the robot inside the tank and the control and inspection consoles outside the tank.

The main advantages of this system are:

- to prevent the unavailability and cleaning operations for an internal inspection;
- the system is able to detect corrosion in critical areas not possible to be inspected by conventional NDT methods;
- will allow to assess corrosion rates and based on that Operational control Deployment system through man hole Bottom plates Vehicle with US sensors Floating roof Umbilical and API recommendations, define more accurate inspection intervals and prevent leaks;
- besides tank bottoms the system can inspect tank walls from inside;
- the system aims at the inspection of crude oil tanks.

The final system will comply with the following applicable standards:

- API 653 – Tank inspection, Repair, Alteration and Reconstruction,
- API 575 – Inspection of Atmospheric and Low-Pressure Storage Tanks,
- Machine Directive 98/37/EC,
- ATEX Directive 94/9/EC,
- Low Voltage Directive 73/23/EEC,HSR of workers Directive 1999/92/EC, CEN EN 1127-1: 1997CENELEC EN 50014: 1997CENELEC EN 50284:1999.

6 The Vehicle

The initial concept model shown in Fig. 7 below consisted of a four-wheeled steering vehicle. The vehicle can be steered by differentially driving the two sets of wheels at different speeds. The vehicle is able to climb the walls of a tank by generating an adhesion force against the wall, with one or more propellers at its top. The motors and their drivers are sealed in a pressurised box with inert gas to make the vehicle complying with the safety requirements.

Requirements ask for a vehicle the light and compact enough to be manually transported on a tank roof and inserted through a man-hole opening larger than 300 mm diameter and retrievable via an umbilical with tether function. The vehicle deploys a payload of NDT sensors for the inspection of plate corrosion on the tank plates with thickness between 5 and 25 mm while operating in liquids such as crude oil, light petroleum products, water or other chemical products.

Fig. 7. Prototype #1 – Concept model

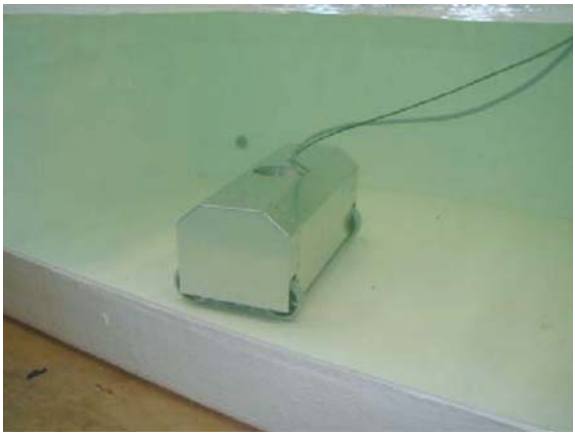


Fig. 8. Prototype #1 operating in water

The vehicle is able to travel on an uneven tank floor that may have plate distortion such as rippling, bulges, depressions, etc. It should be able to travel on or through sediment layers on the floor due to sludge, sand, wax, etc. with maximum height of 50 mm and change surfaces from the floor to a wall to inspect the lower part of tank walls that may be below ground or not accessible from the exterior. The vehicle should be able to carry a payload of navigation sensors and NDT sensors that provide information to an operator to determine vehicle location (position and orientation in the tank), avoid obstacles, select an area of floor that is to be inspected, and avoid tank furniture while moving to the selected area. Control of vehicle to be via both tele-operation and autonomous scanning routines.

The vehicle construction should prevent ingress of these liquids into the central box and to umbilical circuits. Other specifications related to the operational environment are:

Fig. 9. Prototype #1 immersed rotating in wall

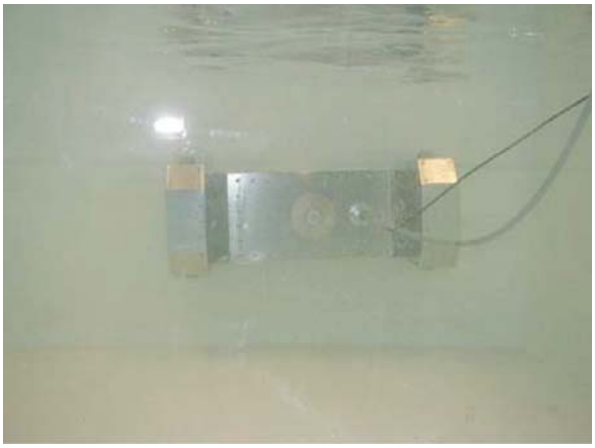


Fig. 10. Prototype #1 immersed climbing in wall

- pH metween 5 and 12;
- temperature of product between 0°C and 70°C,
- maximum pressure 3 bar,
- zone 1 and Explosion proof approval;
- maximum speed of 150 mm/s,

The design of the vehicle for the Robtank Inspec system is modular being made up of three main modules, the central box with propeller, the shell, and the motion mechanism module. The benefit of this way of design is that it is easy to rebuild the vehicle to meet the requirements of different environments. A first prototype surface changing vehicle was developed and was demonstrated operating while submerged in a water tank, as on Fig. 8.

The vehicle is able to enter through manhole openings of 300 mm diameter, travel on the floor, rotate through any angle within the full 360 degree, and change surfaces from the floor to the wall and back to the floor. It can climb the wall vertically to the full height of the liquid column and also travel horizontally on the wall around the tank.

The specification for the umbilical size and its deployment has introduced new requirements on the vehicle. The accommodation of all the components necessary to gather the UT data and for the navigation systems, introduced constraints, due to the change on its mass-which increased- and buoyancy and hence its dynamic characteristics.

The necessary improvements resulted in a second prototype vehicle with optimised shape, size and number of propellers, providing the right surface changing capabilities and providing sufficient thrust for proper adequate wall adhesion. The engineering drawing of prototype #2 is represented on Fig. 11. To monitor and control the vehicle and for data gathering, an integrated console was developed. This console has also features of post processing.

From the database where all the data is stored, it allows post calculations with the measured results and automatically generates reports. It also provides the capability to draw the distribution of plates of the tank bottom and wall when this drawing is not available.

The vehicle is complemented with a vision system, based in an infrared miniaturized camera. This system serves to gather images from the interior of the tank. These images are recorded and are visualised in a dedicated monitor. The prototype #2 system shown on Fig. 11 was field tested in real storage water tanks and in a training tank. The achieved results allow to progress to field hardened system that will fulfill safety requirements and allows industrial application on the referred areas.

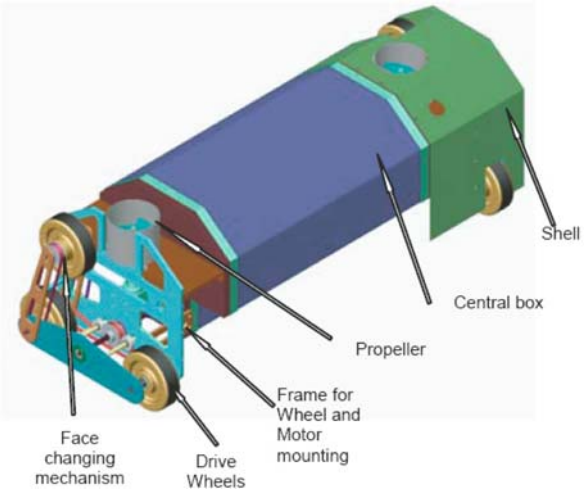


Fig. 11. Prototype #2 concept

Fig. 12. Prototype #2 of Robtank Inspec system

In Fig. 12 it is shown an example of an output reading from the ultrasonic probes of the vehicle. It is possible to identify the recorded A-scan and B-scan echoes measured on the bottom steel plate of the testing tank. The Fig. 13 below, it's a real picture of the vehicle entering a storage tank from the manhole on the roof.

The vehicle must pass through the manhole in vertical position assuming after passage the normal position.

From the inspection made to a raw water storage tank using Robtank Inspec system, a picture captured from the recorded images showing a weld, is below.

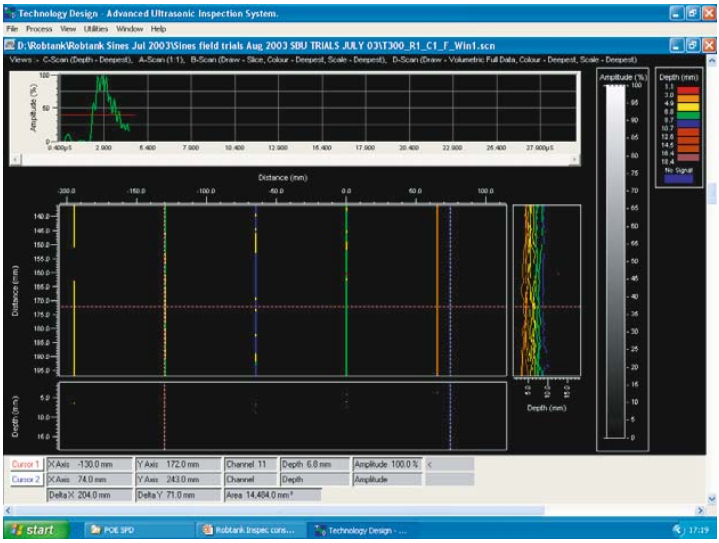


Fig. 13. Example page of console – Results from UT measurements

Fig. 14. Vehicle entering from the roof man-hole

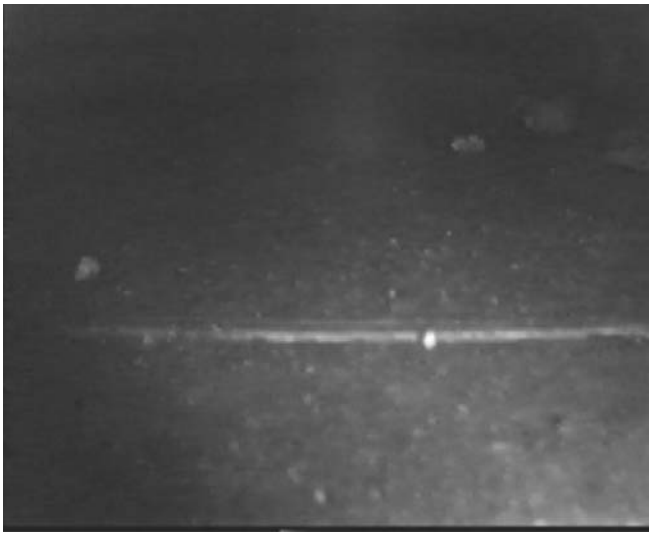


Fig. 15. Part of a weld captured image

7 Conclusions

With the system developed by this project, the emphasis will be placed on oil and chemical leak prevention and condition evaluation of the storage tanks rather than leak detection, which is prevalent actually in some tank users industry. The inspection vehicle will be resistant to hostile environments and will comply with the necessary safety requirements. The inspection tool will advance the field of Robotic NDT application on in-tank and in service inspection.

8 Partnership

Tecnatom, SP, South Bank University, UK, Phoenix ISL, UK, OIS-Oceannering, UK, MT Integridade, PT, Galpenergia, PT., ISQ, PT (coordinator).

Acknowledgements

The authors wish to acknowledge the financial support of the EU, through the Competitive and Sustainable Growth Program and to PRIME – Programa de Incentivos à Modernização da Economia; Economie Ministry of Portugal. Acknowledgements are also due to all partners in the research project.

References

1. Z. You, D. Bauer (1994) Combining Eddy Current and Magnetic Flux Leakage for Tank Floor Inspection, Materials Evaluation. July 1994.
2. K. Newton, D.H. Saunderson (1992) NDT Research for the Oil and Gas Industry, British Journal of NDT, Vol. 34, No. 3. March 1992.
3. U.B. Halabe, K.R. Maser (1993) Leak Detection from Large Storage Tanks Using Seismic Boundary Waves, Journal of Geotechnical Engineering, Vol. 119, No. 3, March 1993.

SIRIUSc – Facade Cleaning Robot for a High-Rise Building in Munich, Germany

N. Elkmann, D. Kunst, T. Krueger, M. Lucke, T. Böhme, T. Felsch, and T. Stürze

Fraunhofer Institute for Factory Operation and Automation, Sandtorstrasse 22, D-39106 Magdeburg, Germany

1 Introduction

The Fraunhofer Institute of Factory Operation and Automation (IFF) has developed and realized an automatic facade cleaning robot SIRIUSc for use on a high-rise building in Munich, Germany. The robot employs the kinematic principle of the "advanced sliding module" to move quickly and efficiently along the facade. It does not need guide rails mounted on the building's facade. This installation constitutes the first commercial implementation in the world of a climbing robot for facade cleaning. SIRIUSc is scheduled to be commissioned in fall of 2004.

2 SIRIUSc

SIRIUSC, a robot for high-rise and skyscraper facade cleaning was developed on the basis of the SIRIUS prototype. The robot cleans one vertical panel at a time. It starts supported by the cables of the gantry at the top of a facade and travels down its side. In the next step, it walks up the facade cleaning as it goes so as to not leave any tracks. A gantry then moves SIRIUSC laterally and it begins the process anew on the next panel of the facade. The robot can clean up to 80 m2 per hour, which includes the time spent moving downward and laterally on the facade. During cleaning no water drips or runs on the panel. The water is sectioned off and filtered in a closed cycle.

The main components of SIRIUSc are:

- Mechanics, kinematics,
- Rooftop gantry and crane,
- Integrated cleaning unit,
- Control technology, sensor systems and navigation and
- Power supply.

Fig. 1. SIRIUSc at the building in Munich

3 SIRIUS Kinematics

Central topic of modular kinematics include ensuring constant contact between robot and facade, being able to overcome a multitude of typical obstacles and being able to move quickly along the facade. Vacuum suckers since they are best suited for providing secure contact to the building's surface. No vacuum suckers are positioned over an obstacle being overcome.

The kinematics is based on a structure of two pairs of linear modules, called the "advanced sliding frame mechanism". A certain number of vacuum suckers are mounted on the modules. The dimensions of the modules and the number of suckers are system parameters which depend on the surface structure of the respective object. An additional degree of freedom (pneumatic cylinder) enables the suckers to move perpendicular to the building surface. Each cylinder can be actuated separately to react when walking over an obstacle. Two linear modules constitute one pair to perform the same linear movement. This guarantees secure and stable contact to the facade. Each pair of linear modules is driven to move the system continuously or intermittently upward and downward. The inner pair of linear modules can rotate around

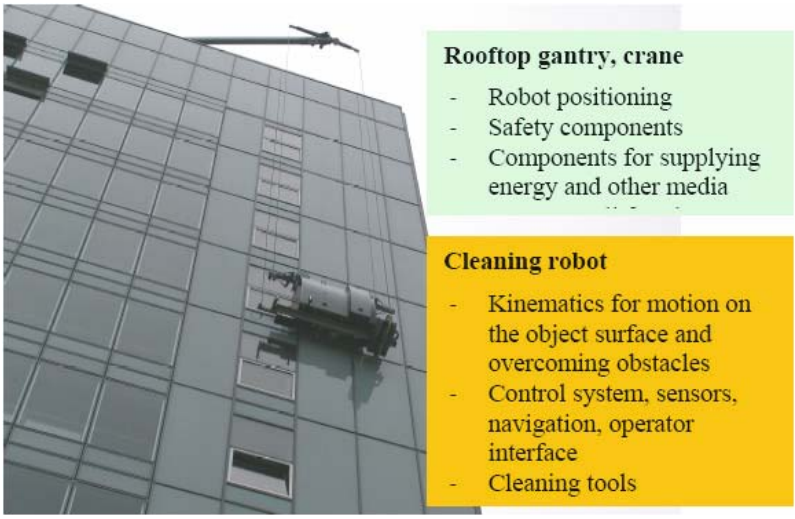


Fig. 2. Components of facade cleaning robot SIRIUSc

a fixed point. This is necessary to compensate for possible robot drift, e.g. caused by wind, while moving along the façade surface.

4 Rooftop Gantry

The complete system consists of more than just the robot kinematics. Four cables connected to the gantry on top of the building secure the robot against falling. Since the cables must be taut to ensure this protection, a logical decision was to also use the cables to bear the load of the modular robot. Data is transferred and power supplied over cables too. Thus, the load of the robot on the respective façade is small. A winch on the rooftop gantry actuates the winding and unwinding of the cable to produce robot movement along the surface. The gantry and cantilever are integrated in the robot control system. Hardware changes were necessary to allow for automatic positioning control.

5 Control Technology

The entire control system is located on the robot and performs the tasks of navigating, controlling the mounted tool and implementing the human-machine interface.

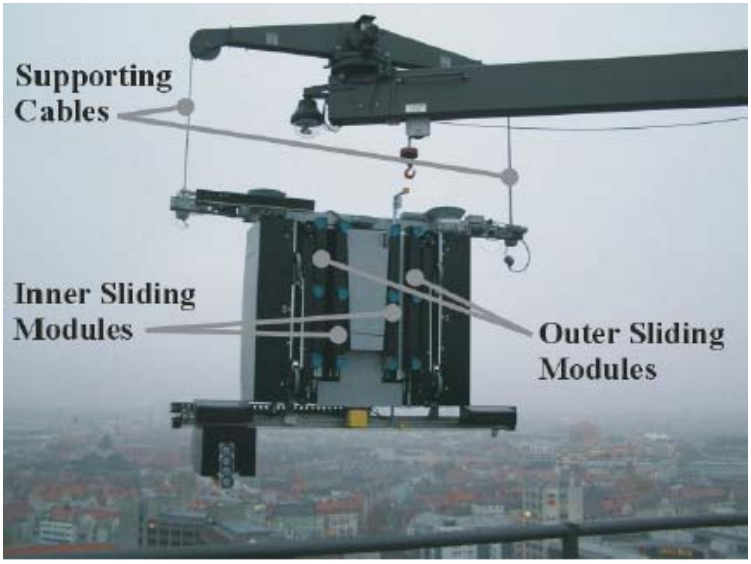


Fig. 3. Kinematics Concept – Advanced sliding frame mechanism

6 Movement Control (Navigation)

Knowledge about the general structure of the building surface was needed before robot movement could be generated. The inputted data contains end-positions, moving distances and path characteristics. This a priori data is supplemented by online sensors which detect the facade surface and search for possible obstacles.

In addition to identifying obstacles, the external sensor technology corrects the direction of motion. Sensors detect the robot's deviation from the ideal path, e.g. sensing girders or window and panel seals.

The possibility of the robot colliding with open windows posed a special risk. All windows are automatically controlled and integrated in the building management control system. The robot control system communicates with the building control system to ensure windows are closed in areas being cleaned. What is more, laser scanners perform a necessary double check in case the main system has malfunctioned and a window has been left open.

7 Rooftop Gantry Control

The Fraunhofer IFF also was also responsible for software implementation and hardware specifications for changes to the gantry. The winch, the lateral movement of the gantry on rails and the positioning of the cantilever were adapted and now have special positioning software for automatic robot

Fig. 4. Rooftop gantry and cantilever with SIRIUSc

operation. Their movement and control is integrated in the robot's control system. Since rooftop gantry systems are usually only manually operated, this development was an important step in making the cleaning system fully automatic.

8 Cleaning Unit Control

The cleaning unit works with its own dedicated plc. An object-based approach was used to create the communication between the cleaning unit plc and the main plc. This approach allowed for easier debugging during the test phase and safe and simple operation of communications during normal use.

9 Human-Machine Interface

For interaction with the operator, all relevant data is run through a control device. The control device only has the tasks of visualizing and starting and stopping robot action. All control functions such as error control are implemented on-board and are executed by robotic control. A WLAN network connects the operator panels to the robot. Important for the human-machine interface were the different layers of control granted to the operator and

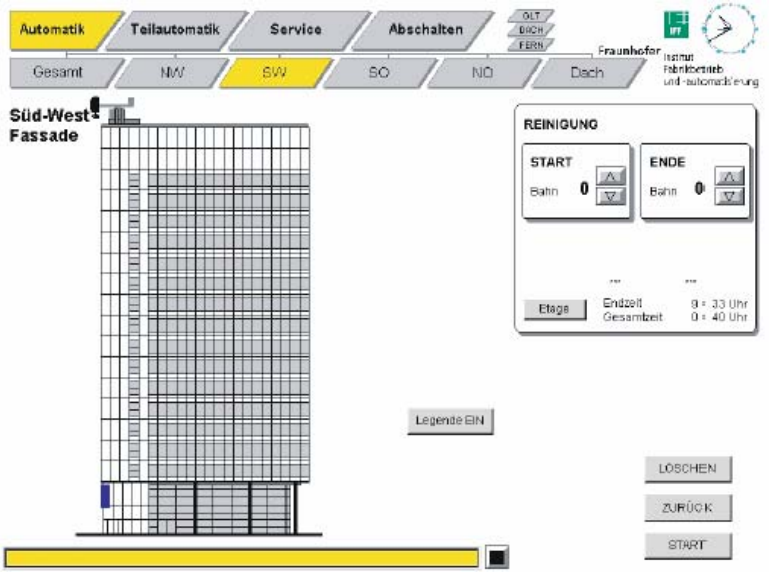


Fig. 5. Operator interface for automatic cleaning

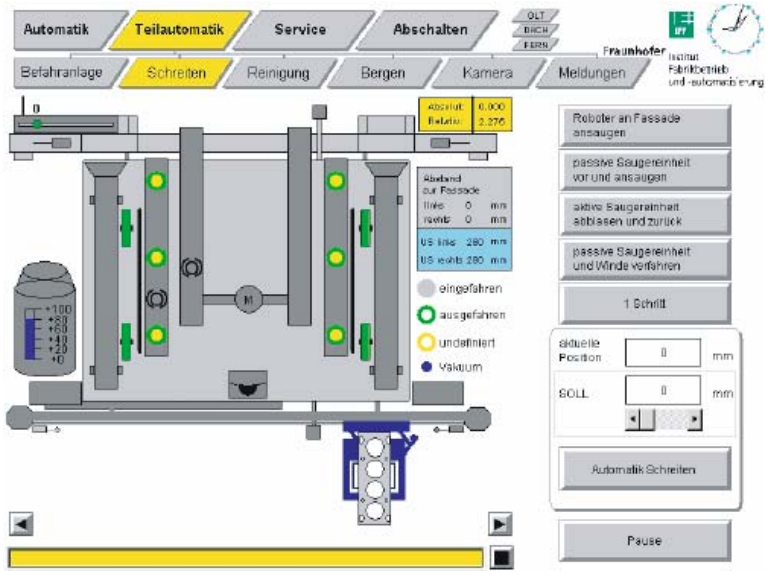


Fig. 6. Maintenance technician interface for semi-automatic functions

Fig. 7. SIRIUSc

maintenance technicians. The system also allows web-based monitoring of the robot's status to better maintain the system.

10 Conclusion

SIRIUSc facade cleaning robot was developed for use on a high-rise building in Munich, Germany and will be commissioned in fall of 2004. This will be a major step for robots outside of factories and will help foster public acceptance of robots in the service sector.

References

1. Böhme T., Elkmann N., Felsch. T., Sack M.,: Service Robots for Façade Ceaning. IECON'98, 24th Annual Conference of the IEEE Industrial Electronics Society, Aachen 1998, pp. 1204–1207
2. Elkmann, N., Felsch, T.: Climbing Robot for Operation at Vertical Facades. Proceedings of CLAWAR 1998, First International Symposium on Mobile, Climbing and Walking Robots, Brussels 26–28. October, 1998, pp. 373–377
3. Elkmann, N, Schmucker, U., Böhme, T., Sack, M.: Service Robots für Facade Cleaning. Advanced Robotics: Beyond 2000. The 29th International Symposium on Robotics, 27th of April – 1st of May Birmingham, 1998, pp. 373–377
4. Elkmann N.: New concepts of service robots for motion at facades. Dissertation. TU Vienna, 1999

5. Elkmann N., Felsch. T., Sack M., Böhme T.: Modular climbing robot for outdoor operations. Proceedings of CLAWAR 1999, Second International Conference on Climbing and Walking Robots, Portsmouth 13–15. September, 1999, pp. 413–419
6. Felsch, Torsten, Elkmann, Norbert; Sack, Mario; Saenz, José: Concepts of Service Robots for Facade Cleaning. International Symposium on Robotics ISR 2001, Seoul, 19.04.01–21.04.01, p. 66
7. Elkmann, Norbert; Felsch, Torsten; Sack, Mario; Saenz, José: SIRIUS, modular climbing robot for facade cleaning and other service jobs. International Conference on Field and Service Robotics FSR 2001, Helsinki, 11.06.01–13.06.01, pp. 403–408

In-Pipe Microrobot with Inertial Mood of Motion

G. G. Rizzoto, M. Velkenko, P. Amato, V. Gradetsky, S. Babkirov, M. Knyazkov, and V. Solovtov

The institute for Problems in Mechanics, RAS, Russia.
gradet@ipmnet.ru
St Micirelectronics Srl, Italy
paolo.amato@at.com

Abstract. Mechatronic in-pipe microrobots are intended to produce inspection or technological repair inside of the small diameter's tubes. In this paper the results of design and study of in-pipe electromagnetic mechatronic microrobot with inertial mood of motion are discussed. The prototypes of the in-pipe electromagnetic robots can move inside of horizontal, vertical and curvature surfaces of the tubes. The inertial mood of operations satisfy the robot's reverse motion in two directions without turning on 180°.

The design of this in-pipe robot is based on the application of controlling arresting devices and on the distinction of friction coefficients between working internal surfaces of robot and pipe in different phases of the motion to satisfy the braking and reliable alternate coupling of moving robot's part with internal surface.

1 Introduction

The problem was reflected early of design and application of miniature or micro in-pipe robots [1, 2]. Various design methods were suggested in the articles [2–13]. One of important application up to day is inspection of wall cracks of the small diameter pipes [13]. Such a robot can carry small CCD camera inside of the tubes with 10–40 mm diameters over vertical, horizontal or curved pipes to detect the quality of holes or cracks over internal surfaces of the tubes. Such a robots can to repair defects in order to prevent accidents.

Methods of the motion can be different and have some advantages and disadvantages. Analytical solutions of such a models are rather complex and not delivered up to day even.

In this paper new type of microrobot with inertial mood of motion is delivered. Analytical description of this system can be obtained only under some admissions on mechanical phenomena of the processes. The main of the admissions are the following: the mechanical bodies are solid, no deformations

of the interaction bodies, mechanical model is linear, the displacements are small.

The analytical description of the robot motion model is suggested. In the designed model the core is moving inside of robot body and translate it some momentum of motion. Having got momentum from the core, the body begins forward motion. After stop the core fluently moves backward and gathers momentum, the body during this process stays on the same place because of friction forces. Then the core having got momentum again and translate it to robot body.

The dynamical equation solutions of this system were obtained under conditions that sinusoidal force between body and core and sinusoidal magnetic field is applied. The numerical calculations were produced for working designing parameters of the system to receive dynamic, responces, vibrations, resonance frequency, forces characteristics and phenomenon of "sliding back" of the system.

The experiments were produced **and the experimental characteristics** of the system were obtained and compared with analytical and numerical data.

The experiments show the effectiveness of the suggested solutions and expansion of the possible application areas of in-pipe electromagnetic microrobots.

The generalized data were presented that was necessary to illustrate parametrical analyzes of the typical characteristics of in-pipe electromagnetic robot to show how the interference and mutual effect of working parameters were produced, what needed for design and development of prototypes.

The calculation of magnetic field near solenoid permitted to receive the characteristics in 3D for the design the dependence of magnetic field intensity in different zones of system, the dependence of magnetic pressure on geometrical design parameters and the magnetic pressure on distance from the edge of solenoid.

2 Analytical Description of the Robot Motion

Design of in-pipe robot (Fig. 1) is based on inequality of friction coefficients between working surfaces of robot and pipe. This model has such disadvantage as possibility of coming to a standstill because small rise of friction coefficient in one of bases.

If friction coefficient of forward motion in back base is more then friction coefficient of backward motion in front base, then robot will not move forward and come to a standstill.

Another model (Fig. 2) does not have any external moving parts and that is why it does not come to a standstill under the same circumstances. Friction coefficient of forward and backward motion may be equal and construction is

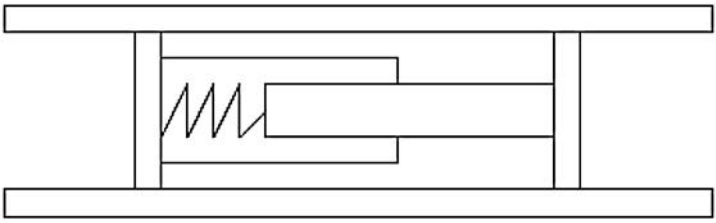


Fig. 1. In-pipe robot's design

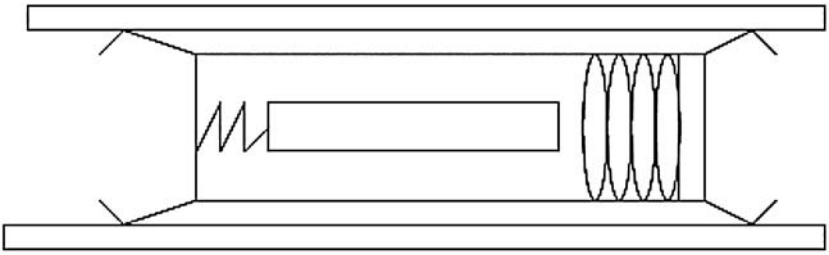


Fig. 2. In-pipe robot with inertial type of motion

much easier, the diameter of the robot may not be equal to the diameter of the pipe.

Design scheme of the in-pipe robot with inertial type of motion consists of body and core. Core can move inside the body and collide front side of the body giving it some momentum. Having got momentum from the core, body begins forward motion. After stop the core fluently moves backward and gathers momentum, the body during this process stays on the same place because of friction forces. Then the core having got momentum again collides the body and again gives it some momentum. It was approximate description of process of motion of this robot.

Let us consider the problem of motion of this robot.

During the collision between the core and the body amount of energy and momentum remain the same:

$$\begin{cases} m\nu + MV = m\nu_0 \\ \dfrac{m\nu^2}{2} + \dfrac{MV^2}{2} + Q = \dfrac{m{\nu_0}^2}{2} \end{cases}$$

where m and M–masses of the core and the body respectively,

v and V – their velocities after collision,

v_0 – velocity of the core before the collision,

Q – thermal energy gained during the collision, that was kinetic before it, this parameter characterizes viscosity of the collision.

The solution of this system will be the following:

$$\begin{cases} v = \dfrac{m}{m+M}\left(v_0 - \dfrac{m+M}{m}\sqrt{{v_0}^2 - (1+\dfrac{m}{M})\dfrac{2}{m}Q}\right) \\ V = \dfrac{m}{m+M}\left(v_0 + \sqrt{{v_0}^2 - \left(1+\dfrac{m}{M}\right)\dfrac{2}{m}Q}\right) \end{cases}$$

It is seen, that velocity v of the body rises when $Q \to 0$, $\frac{m}{M} \to \infty$. Besides this, it is obvious, that velocity of the core after collision can not be smaller than $(-v_0)$.

Let us consider motion of this system after collision. It is supposed that the core has the coordinate along the direction of motion x, and the body y; stiffness coefficient of the spring is equal to k, the force of friction equals F_{mp}.

3 The Dynamical Equation Solutions

Then, dynamical equations of this system will be as follows:

$$\begin{cases} m\ddot{x} = -k(x-y) \\ M\ddot{y} = -k(y-x) + F_{mp} \end{cases}.$$

k – is the coefficient between displacement and force in the case of linear force dependence on displacement.

The solution of this system can be represented in the form of harmonic function combination in the common case:

$$\begin{cases} x(t) = C_1 + C_2 t + F_{mp}\dfrac{1}{2(m+M)}t^2 + C_3\cos(wt) + C_4\sin(wt) \\ y(t) = \left(C_1 + \dfrac{m}{k(m+M)}F_{mp}\right) + C_2 t + F_{mp}\dfrac{1}{2(m+M)}t^2 \\ -C_3\dfrac{m}{M}\cos(wt) - C_4\dfrac{m}{M}\sin(wt) \end{cases},$$

where $w = \sqrt{k\frac{m+M}{mM}}$.

To find constants C_1, C_2, C_3, C_4, it is necessary to introduce initial conditions. In the case of our analytical description the following initial conditions can be introduced:

$$\begin{cases} x(0) = x_0 \\ \dot{x}(0) = v \\ y(0) = y_0 \\ \dot{y}(0) = V \end{cases}.$$

After solution of the dynamical equations and substitution of the initial conditions, in the receiving solution, the constants C_1, C_2, C_3, C_4 may be expressed as follows:

$$C_1 = \frac{M}{m+M}\left(\frac{m}{M}x_0 + y_0 - \frac{m}{k(m+M)}F_{mp}\right),$$
$$C_2 = \frac{M}{m+M}\left(\frac{m}{M}V - \nu\right),$$
$$C_3 = \frac{M}{m+M}\left(\frac{m}{M}x_0 - y_0 + \frac{m}{k(m+M)}F_{mp}\right),$$
$$C_4 = \frac{M}{m+M}\frac{1}{w}(v - V)\ .$$

Naturally, this solution is valid until velocity's $\dot{y}$ sign remains the same. Frankly speaking, it is more correctly to consider semicontinuous right part of the second differential equation (equation of the body motion):

$$\begin{cases} m\ddot{x} = -k(x-y) + F(t) \\ M\ddot{y} = -k(y-x) - |F_{mp}|\, sign(\dot{y}) - F(t) \end{cases}$$

Let us apply a sinusoidal force between the body and the core. As far as more probably, that is will be caused by sinusoidal magnetic field, let us set it equals:

$$F = F_0 \sin^2(wt)\ .$$

Further the numerical calculations will be produced with parameters

$$M = 1,\ m = 0.3,\ k = 1,\ F_{mp} = 0.5,\ F_0 = 0.1,\ Q = 0.5 * \left[\frac{m\nu_0^2}{2}\frac{1}{1+\frac{m}{M}}\right]$$

(half of kinetic energy of the core transfers to heat).

4 The Results of Experiments and Numerical Calculations

For the convenience of numerical calculation and to increase the accuracy it is useful to change the force of friction to continuous function of speed:

$$F_{mp} = F \cdot sign(\dot{x}) \approx F \cdot \left(\frac{2}{1 + Exp(-a\dot{x})} - 1\right)\ .$$

If a is great, such function precise enough approximate the function $F \cdot sign(\dot{x})$.

The dependence of coordinate of the body of this robot of time is represented on Fig. 3 and the coordinate of the core on Fig. 4.

Besides this, if robot's design is as described in this paper, it is possible to move in both directions – forward and backward. To do it, it is only needed to change the direction of the force causing core vibrations for it to collide not front side but back one.

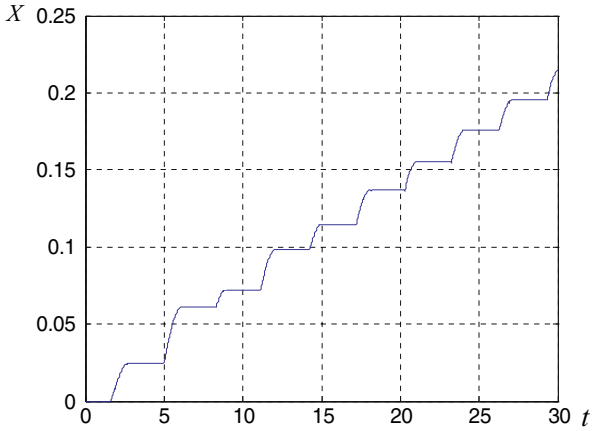


Fig. 3. The dependence "robot's body coordinate – time"

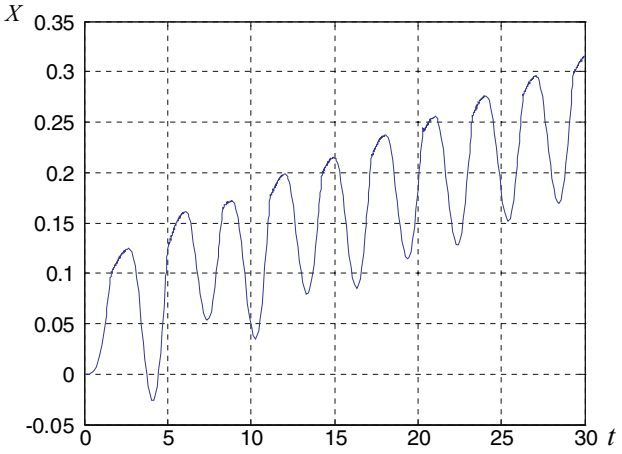


Fig. 4. The dependence "coordinate of the core – time"

Let us consider the same problem for small dimensions of robot:

$$M = 0.002,\ m = 0.0005,\ k = 0.2, F_{mp} = 0.002,\ F_0 = F_{mp}/2, Q = 0.5^* \times \left[\frac{m{v_0}^2}{2}\frac{1}{1+\frac{m}{M}}\right].$$

When the frequency of the force causing the core vibrations is equal to resonance frequency, the dependence of coordinate of the body of robot of time is represented on Fig. 5.

Let us find out the dependence of mean velocity of mass of the core in the case of resonance frequency of vibrations.

If Q is small, the formula for initial speed of the body V will be:

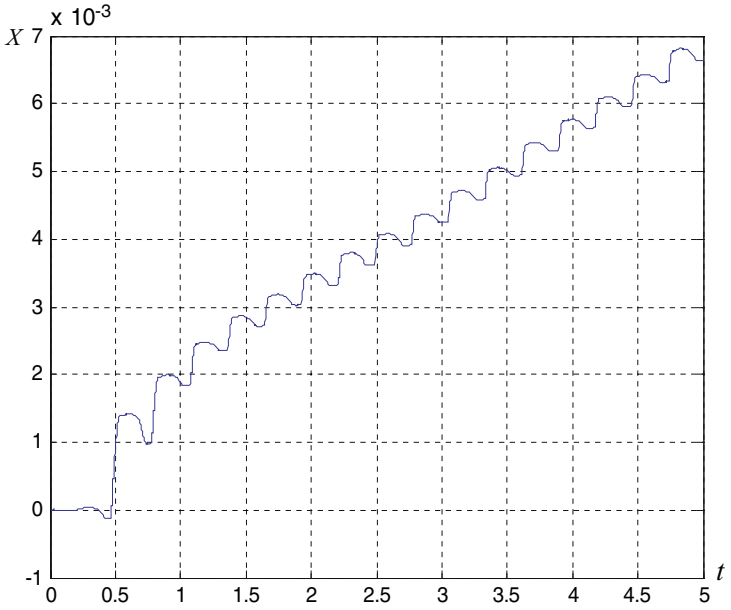


Fig. 5. The dependence "coordinate of the core – time"

$$V = C \cdot \frac{m}{m+M} \approx C \cdot \frac{m}{M}\left(1 - \frac{m}{M} + O\left(\left(\frac{m}{M}\right)^2\right)\right) \approx C_1 \cdot m \quad \left(\frac{m}{M} \to 0\right) ,$$

the dependence tends to linear one.

It is obvious, that frequency of resonance vibrations under the same approximations is the function of the core mass such as:

$$w = \sqrt{k\frac{m+M}{mM}} = C \cdot \sqrt{\frac{1}{m}} \cdot (1 + \frac{1}{2} \cdot \frac{m}{M} + O\left(\left(\frac{m}{M}\right)^2\right)) \approx C_1 \cdot m^{-\frac{1}{2}} .$$

The dependence of mean velocity with frequency to be equal the resonance one of $\frac{m}{M}$ is represented on Fig. 6.

Also it is interesting to determine the displacement of the robot per one period, this dependence is shown on Fig. 7.

The rise of displacement then $\frac{m}{M}$ is small is caused by that during a collision the body of robot is given more momentum. And the fall then $\frac{m}{M}$ is great is caused by appearing the phenomenon of sliding back – Fig. 8.

It is possible to avoid it by increasing the force of friction, but it will decrease forward displacement.

On the base above mentioned calculation and experimental study the modular design of in-pipe microrobots was suggested and illustrated by Fig. 9.

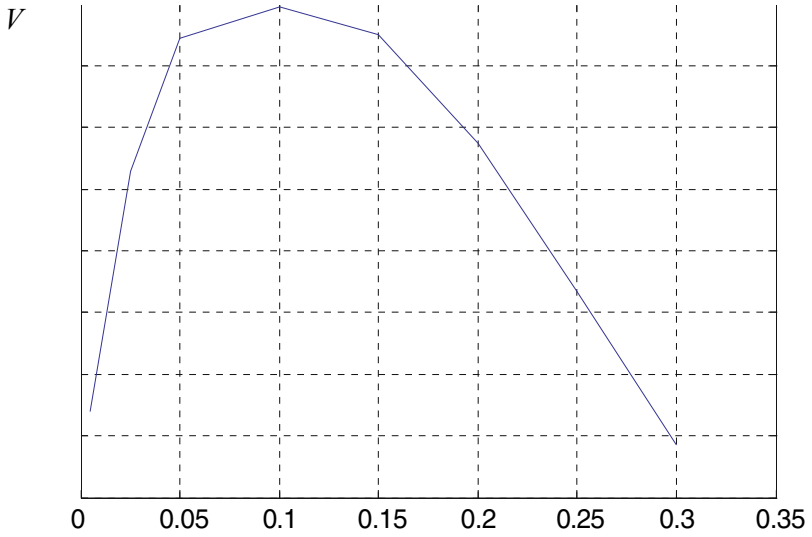


Fig. 6. The dependence of mean velocity of m/M in the case of m/M resonance frequency

Table 1.

Following Parameters Diameter of the Robots	B	Φ	W	F_1	ΣF_2	U_{CD}	I_{CD}	t_π	$t_П$	T	f
MM	TJI.	–	H.	H.	B	A	C	C	C	C	$Г_П$
10	0,5	4.10^{-6}	200	0,4	0,31	6,8	0,8	$3{,}2\text{-}10^{-2}$	$1{,}8.10^{-2}$	$5{-}10^{-2}$	20
5	0,5	1.10^{-6}	180	$0{,}25.10^{-1}$	$0{,}2.10^{-1}$	4,5	0,5	$2{,}8{-}10^{-2}$	$2{,}7.10^{-2}$	$5{,}5{-}10^{-2}$	18,1
1	0,5	4.10^{-8}	120	$0{,}4.10^{-2}$	$0{,}32.10^{-2}$	1,5	0,15	$2{,}9{-}10^{-2}$	$2{,}9.10^{-2}$	$5{,}8{-}10^{-2}$	17,2
0,1	0,5	4.10^{-10}	80	$0{,}4.10^{-4}$	$0{,}33.10^{-4}$	0,8	$0{,}6{-}10^{-2}$	$3{,}2\text{-}10^{-2}$	$2{,}8.10^{-2}$	$6{,}0{-}10^{-2}$	16,6
0,01	0,5	4.10^{-12}	50	$0{,}4.10^{-6}$	$0{,}35.10^{-6}$	0,4	$0{,}8{-}10^{-4}$	$3{,}9\text{-}10^{-2}$	$2{,}9.10^{-2}$	$6{,}5{-}10^{-2}$	15,0

The main technical working parameters of one of the prototypes are delivered in Table 1

Here are the following parameters:

B – magnet induction or induction of magnet field; d – robot body diameter; Φ – magnet flow; W – coil's number of stator; F_1 – electromagnetic force or acting force; ΣF_2 – resistance forces; U_{cp} – supply voltage, average value; I_{cp} – current, average value; t_u – time of supply pulse; t_n – time delay between pulse supply pulses; T – period of pulse series; f – frequency.

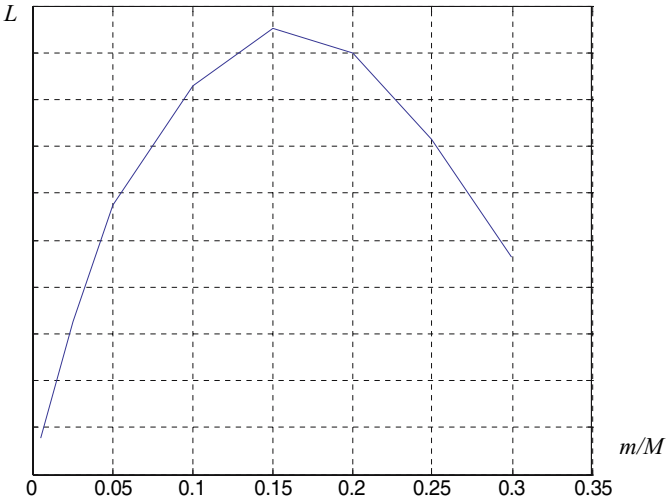


Fig. 7. The dependence of displacement of robot per one period of m/M

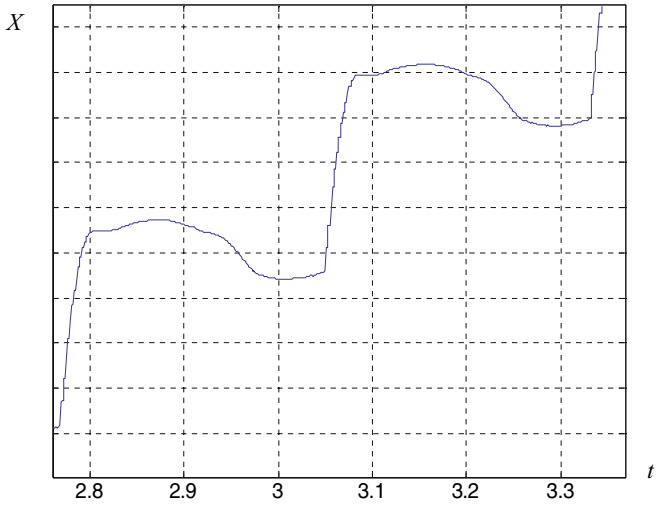


Fig. 8. The phenomenon of sliding back

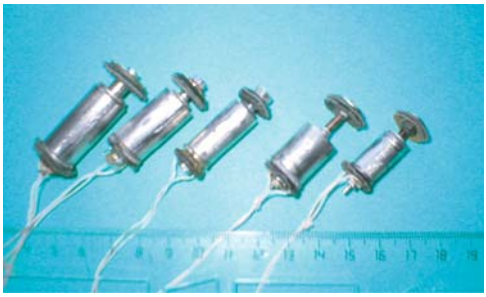


Fig. 9. Modular design of in-pipe microrobots

5 Conclusion

The analytical description and experiment study of the in-pipe microrobot with inertial mood of motion is suggested. In the designed model the core is moving inside of robot body and translate it momentum of motion. The results of experiments and numerical calculation are discussed.

References

1. G.G. Rizzotto, P. Amato, V. Gradetsky, V. Solovtsov, M. Knyazkov "Simulation and Modeling of Electro-Magnetic Mechatronic Microsystems", Proceedings of the IARP Workshop on Microrobots, Micromachines and Systems, Moscow, Russia, April 24–25, 2003.
2. V. Gradetsky, V. Solovtsov, M. Knyazkov, G.G. Rizzotto, P. Amato "Modular Design of Electro-Magnetic Mechatronic Microrobots", Proceedings of the 6th CLAWAR 2003 International Conference, Italy, 2003, pp. 651–658.
3. V. Gradetsky, V. Veshnikov, S. Kalinichenko, L. Kravchuk "Control motion of the mobile robots over orbitralary oriented surfaces into the space", Nauka publisher, Moscow, 2001, p. 359 (in Russian).
4. V. Rakhovsky "Power actuators for micro and nanotechnology", Proceedings of IARP WS on Micro Robots, Micro Machines and Systems, November 1999, Moscow, pp. 219–229.
5. E. Kallenbach, E. Saffert, O. Birii, E. Raumschiissel "Function Oriented Configuration of Integrated Mechatronic Systems", Technical University of Ilmenau, Faculty of Mechanical Engineering Institute of Microsystems Technology, Mechatronics and Mechanics Ilmenau, Germany, 2000.
6. V. Gradetsky, V. Veshnikov, L. Kravchuk, S. Kalinichenko, M. Knyazkov, V. Solovtsov "Miniature Robot Control Motions Inside of Tubes". Proceedings of the 5th CLAWAR 2002 International Conference, pp. 643–650.
7. Takahashi M., Hayashi I. et al., "The Development of An In-Pipe Micro Robot Applying The Motion of An Earthworm", Proceedings of MHS'94, pp. 35–40, Nagoya, Oct. 4–6, 1994.
8. Hayashi I. et al., "The Running Characteristics of A Screw Principle Micro Robot in A Small Bent Pipe", Proceedings of MHS'95, pp. 225–228, Nagoya, Oct. 4–6, 1995.

9. Sun Linzhi, Sun Ping, Qin Xinjie "Study On Micro Robot In Small Pipe", Proceedings of Inter. Conference on Control'98, pp. 1212–1217, Swansea, 1998.
10. Idogaki T. et. al, "Characteristics of Piezoelectric Locomotive Mechanism for an In-Pipe Micro Inspection Machine", Proceedings of MHS'95, pp. 193–198, Nagoya, 1995.
11. Kawakita S. et al., "Multi-Layered Piezoelectric Bimorph Actuator", Proceedings of MHS'97, pp. 73–78, Nagoya, Oct. 5–8, 1997.
12. Sun L., Sun P., Luo Y., Zhang Y., Gong Z. "Micro in-pipe robot with PZT actuator", Proceedings of 4th Inter. Climbing and Walking Robots, pp. 539–546, Karlsruhe Sept. 24–26, 2001.
13. Sun L., Lu L., Qin X., Gong Z. "Micro Robot for Detecting Wall Cracks of Pipe", Proceedings of 6th CLAWAR 2003 International Conference, Italy, 2003, pp. 643–650.

The Layer-Crossing Strategy of Curved Wall Cleaning Robot

R. Liu, J. Heng, and G. Zong

Robotics Institute, Beihang University, Beijing, People's Republic of China

Abstract. In this paper, a layer-crossing control strategy applied to curved wall cleaning robot, which is a kind of self-climbing robot, is introduced. With this strategy, the robot can smoothly climb form one layer to another, which have different sloping angles with respect to horizon. Some problems, which may occur during the process of layer crossing, are analyzed. Some solutions are put forward and they are proved effective in practice.

1 Introduction

Aimed at cleaning the curved wall of the National Grand Theater of China, a dedicated cleaning robot has been developed recently [1]. The curved wall is vertically divided into different layers, with each layer being a plane having different height and sloping angle. To clean the whole curved wall, the robot must autonomously climb layer by layer on the wall. However, the process of layer crossing is very complicated, during which the robot has to adjust its orientation to comply with the sloping angle of the layer. Without proper technique, the robot may impact the wall and its motion is unstable.

In this paper, an adaptable control strategy for the layer crossing of the cleaning robot is presented. With the help of supporting mechanism on the robot, the robot can smoothly climb form one layer to the next. This paper is organized as follow: in Sect. 2, the mechanism of the robot and its control system are introduced. The wall climbing process of the robot is presented in Sect. 3. In Sect. 4, some problems, which are found during the layer crossing, are discussed detailedly. In Sect. 5, the strategy for the layer crossing aimed at solving above problems is discussed. Section 6 gives the conclusion of the paper.

2 The Mechanism and Control System

An auto-climbing robot on the curved wall of the National Grand Theater of China is designed as shown in Fig. 1. There are three mechanical modules which correspond to different actions on the robot, namely, climbing module, round-moving module and washing module. The motion of robot across the layers is carried out by the climbing module. On the climbing module, there are two frames – main frame and auxiliary frame, which can relatively move when one frame is fixed on the wall. A pair of clutches on each of frames is designed to grasp the track between the layers on the wall. On the main frame, there are two extensible wheels, which can be used as supports for the weight of robot when the robot climbs across the layers.

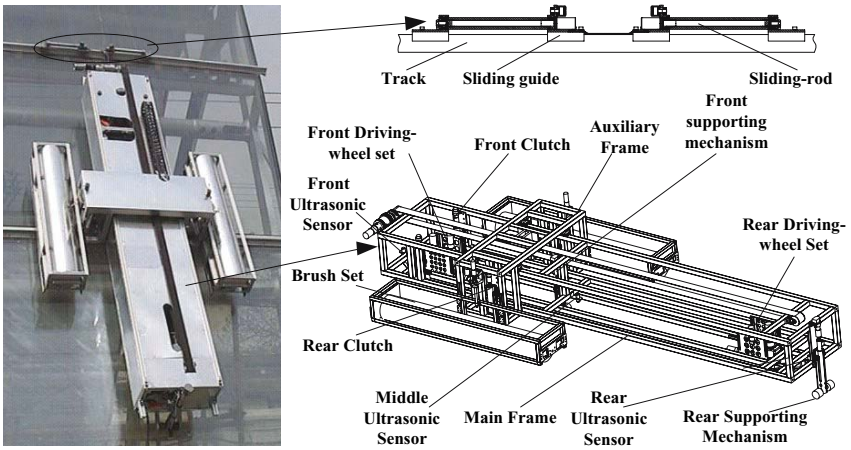


Fig. 1. Mechanical Structure of Cleaning Robot

The robot is controlled by a programmable logic controller (PLC), which has the features of high reliability and easy programming. Sensors used for robot climbing include two kinds: three ultrasonic range sensors mounted on the front, middle and rear of the main frame are used to detect the position of track between layers, and several electromagnetic switches are used to detect some special relative positions between the main frame and the auxiliary frame.

The detailed information about the mechanism of the robot and the use of the sensors can be found in [1].

3 Climbing Process Analysis

The climbing-up process of robot is shown in Fig. 2. Each step is illuminated as follows:

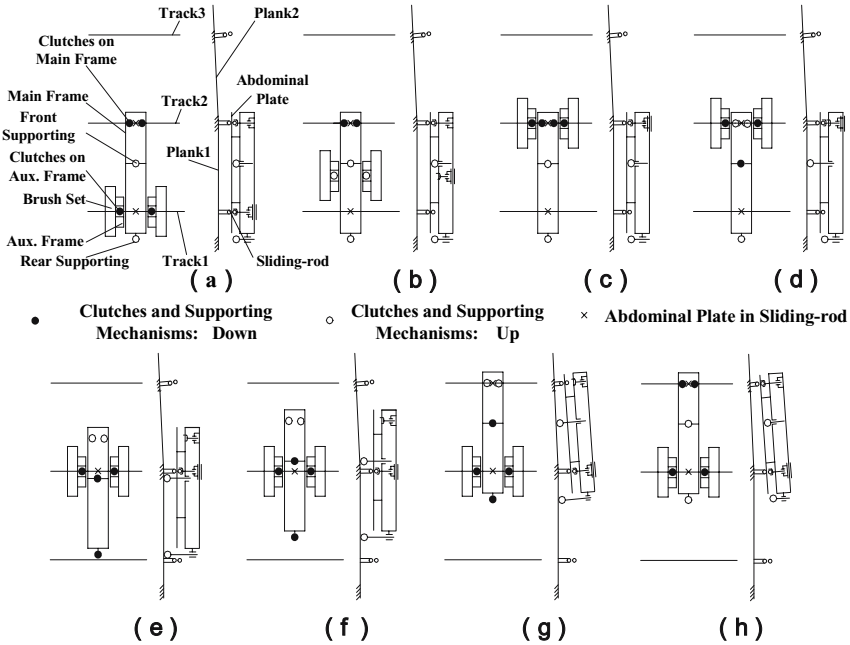


Fig. 2. The process of climbing up

(a) The robot is in Home State.
(b) The auxiliary frame is pulled up with respect to the main frame.
(c) The clutches on the auxiliary frame protrude down to hold the sliding-rod.
(d) The Clutches on the main frame withdraw up.
(e) The main frame moves up.
(f) The front and rear supporting mechanism adjust the parallel of robot to the plank.
(g) The front ultrasonic sensor detects the track and stops the main frame.
(h) The robot is back to Home State again.

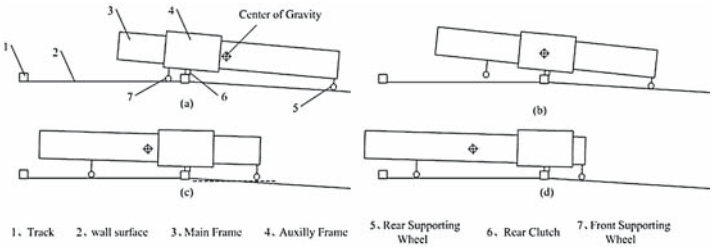
It can be seen that during this process the clutches on the main frame and the auxiliary frame alternately grasp the sliding rod on the track, making corresponding frame be a fixed object with respect to the wall, then the relative motions between two frames can achieve the functions of climbing-up or climbing-down. To accomplish all climbing-up steps, the collaboration within the main climbing motor, two pairs of clutches and two supporting wheels are needed.

The climbing-down process has similar steps. In following analyses, the climbing-up process will be discussed for the most part.

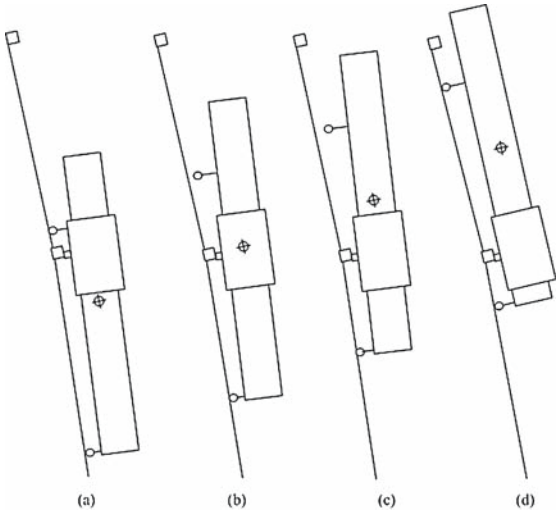
4 Problems During the Layer Crossing

Among all steps of climbing-up in Fig. 2, step-(f) and (g) demonstrate the process of robot climbing from one layer to the next. During this layer crossing process, only the clutches on the auxiliary frame grasp sliding rod on the track. In this case, the sliding rod acts as a pivot around which the robot may rotate under the effect of gravity.

Figure 3 shows the detailed layer crossing processes for two typical situations. In the start position (the state-(a) for both processes), two supporting wheels touch the upper and lower layers respectively. If the adaptable control strategy presented in the next section is not applied, when the main frame moves up the front wheel will lift apart from the upper layer gradually until the center of gravity of robot is just vertically above the pivot (state-(b) for



(1) The basic sloping angle of layer is 15°, the angle difference of between the upper layer and lower layer is 3°.



(2) The basic sloping angle of layer is 75°, the angle difference between the upper layer and lower layer is 5°.

Fig. 3. Climbing up on different sloping-angle layers

process-(1), and state-(d) for process-(2)). Near these critical states the robot may rotate either clockwise or anti-clockwise, depending on which wheel being on the wall and some other factors, such as inertia, wind force, etc. It can be seen from Fig. 3 that if the sloping angle of layer is different the critical state in which the center of gravity is right above the pivot is also changed, and with the increase of the sloping angle of layer the time when the robot arrives at the critical state is prolonged.

Because the moment at which the critical state arrives is practically unexpectable, it makes the smooth layer crossing of robot difficult. The rotation of robot around the pivot with one supporting wheel in the air may result in impact between the wheel and the wall, which may bring in unstable swaying of robot, and is harmful for both the robot and the wall. For example, the ultrasonic range sensors, which are used to detect the position of track between layers, are likely to be disabled because of the swaying. Without the help of the ultrasonic sensors, the robot will quite likely bump into the track ahead.

5 Control Strategy for Layer Crossing

To achieve smooth layer crossing near the critical state, the best solution is to let both supporting wheels touch respective layers at the same time. However, this solution may result in over-constraint to the robot, because there are three contacting points(two wheels and one pivot) between the robot and the wall. Therefore, some other measures have to be taken to achieve smooth layer crossing.

Firstly, both of the extensible supporting wheels are equipped with touch sensors, which can signal the PLC whether or not the wheels touch the wall under it (Fig. 4). Furthermore, a mechanical spring is mounted in the wheel structure, which provides a passive freedom for the wheel when it is on the wall and prevents the robot from being locked because of over-constraint.

Secondly, a special control method is designed. Two supporting wheels are differently treated for different climbing direction, one being main wheel and the other being auxiliary wheel. When the robot climbs up the front wheel is treated as main wheel, otherwise the rear wheel is treated as main wheel. The adaptable control strategy is to keep the main wheel touching the wall at any time during the layer-crossing process.

Using above adaptable control strategy, the layer-crossing process shown in Fig. 3 can be modified as following:

While the main frame is moving up from the start position, the front wheel (main wheel) will extend out to touch the upper layer wall until it reaches its outmost stroke, which means that the front wheel can not keep touching the wall any longer by adjusting itself. If the main frame continues to move up, only the rear wheel (auxiliary wheel) can be used to extend out, which will make the robot rotate around the pivot and keep the fully extended front

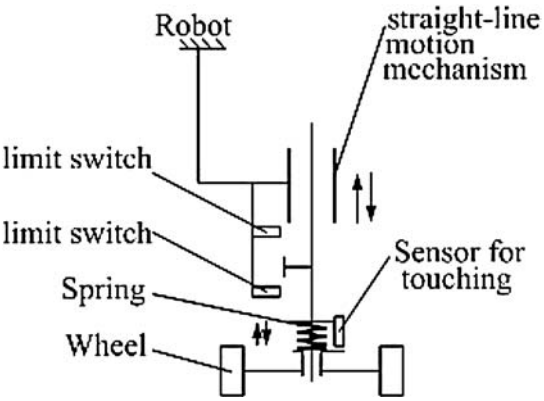


Fig. 4. Mechanical Structure of Supporting Wheel

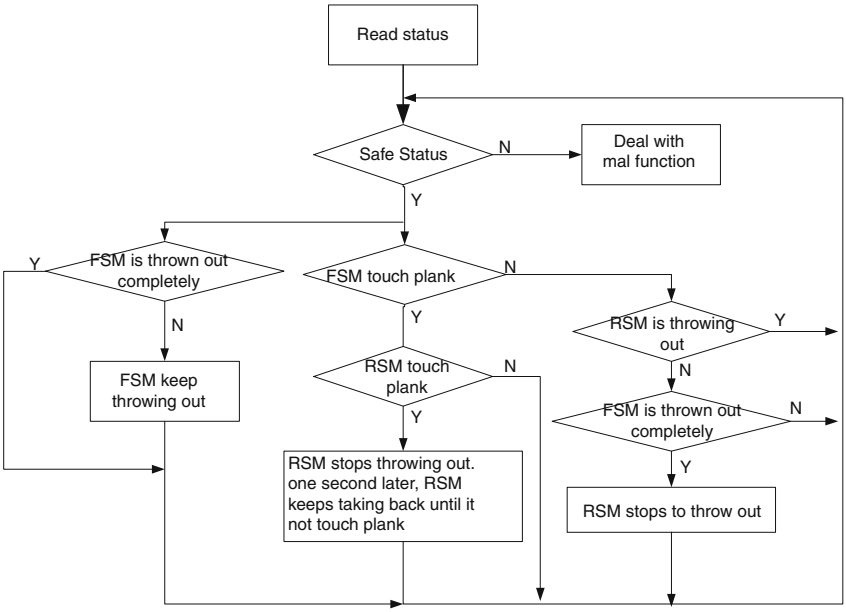


Fig. 5. Flow diagram for climbing up

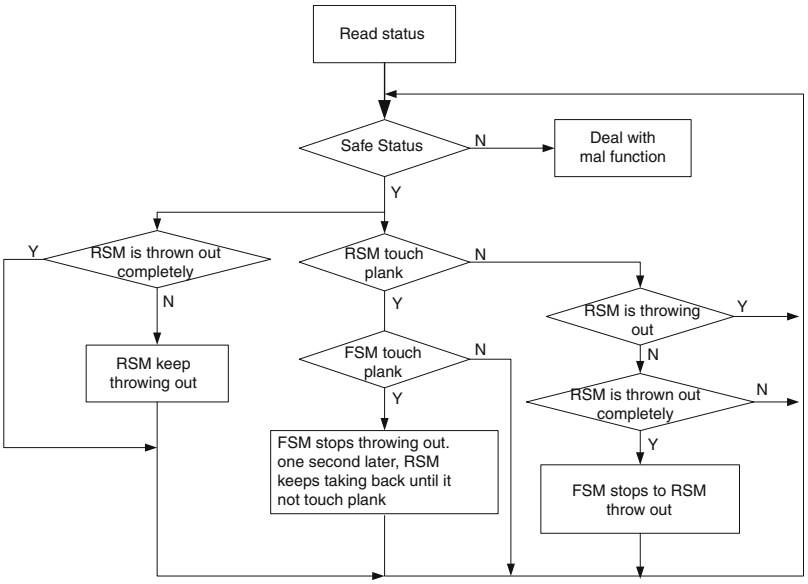


Fig. 6. Flow diagram for climbing down

wheel touching the wall. To avoid over-constraint, the controller will record the time in which both wheels simultaneously touch the wall. If the time exceeds a given limit, the rear wheel (auxiliary wheel) will be withdrawn back until one of wheels lift apart from the wall (The front wheel will lift apart from the wall if the center of gravity is still behind the pivot, otherwise the rear wheel will lift if the center of gravity has exceeded the pivot.). In the case of the front wheel leaving the wall, the rear wheel will be controlled to extend out to let the front wheel touch the wall again. This control loop (rear wheel extending out and withdrawing back) will go on until the rear wheel lift apart from the wall or the robot reaches the track ahead. Figure 5 shows the flow diagrams for climbing-up control. The similar control strategy is used in Moving-down process. Figure 6 demonstrates its flow diagram.

Experiments of robot climbing have been done with above adaptable control strategy, on the 15° layer and 75° layer respectively. In the process of layer crossing, the repeating adjustment of the auxiliary wheel may result in little swaying on the robot, here is no impact between the wheel and the wall. Furthermore, during the short-time simultaneous touching of both wheels, the mechanical springs on them provide a good tolerance to avoid over-constraint.

6 Conclusion

Tests of robot on an experimental curved wall have proved the validity of the control strategy presented in this paper. The time it takes for the robot to climb from one layer to the next is about 3 minutes, and during the layer crossing there is no impact between the supporting wheel and the wall.

Reference

1. Liu R, Zong G, Zhang H, Li X (2003) A Cleaning Robot for Construction Out-wall with Complicated Curve Surface. In: Proceedings of CLAWAR03, pp. 825–832, Catania

Pneumatic Climbing Robots for Glass Wall Cleaning

Houxiang Zhang, Jianwei Zhang, Rong Liu, Wei Wang, and Guanghua Zong

TAMS, Department of Informatics, University of Hamburg, Germany; Robotics Institute, BeiHang University, China

Abstract. Recently various robots have been designed for wall cleaning and maintenance. There are three kinds of kinematics for the motion on smooth vertical surfaces: multiple legs, a sliding frame and a wheeled and chain-track vehicle. Four different principles of adhesion which are vacuum suckers, negative pressure, propellers and grasping are also used by climbing robots. Since 1996 the group at the Robotics Institute of BeiHang University has developed a series of autonomous climbing robots with sliding frames for glass-wall cleaning. In this paper, the individual robots' mechanical constructions and unique aspects are introduced in detail.

1 Introduction

There are a large number of high-rise buildings with glass-curtain walls in modern cities. External cladding of buildings not only provides an attractive exterior appearance, but also increases their durability. These walls require constant cleaning which is always dangerous and laborious work in mid-air. An alternative solution is to use climbing robots which overcome the abovementioned problems [1, 2]. Application of such cleaning systems can free workers from the hazardous work and realize the automatic cleaning of high-rise buildings, thus improving the technological level and productivity of the service industry in the area of building maintenance.

Since 1996 the group at the Robotics Institute of BeiHang University has developed a family of autonomous climbing sky cleaner robots with sliding frames for glass-wall cleaning. This paper is organized as follows: in Sect. 2, the basic functions of the glass wall cleaning robot are discussed. Section 3 presents an overview of three robotic systems. The individual robots' mechanical constructions and unique aspects are introduced in detail. A summary of the main special features of the three cleaning robots is given in Sect. 4.

2 Basic Functions of Cleaning Robotic Systems

The basic functions of glass wall cleaning robots include 8 aspects.

(1) The safe and reliable attachment to the glass surface: The climbing robot should be sucked to the glass wall safely and overcome its gravity.
(2) Movement spreading over all working areas: The robots should have a function to move in both the up-down direction as well as the right-left direction to get to every point on the glass, so that the cleaning movement would cover the working areas.
(3) The ability of crossing window obstacles: In order to finish the cleaning work, the robots will have to face all obstacles and cross them safely and quickly.
(4) Enough intelligence for the discrimination: Multiple sensing and control systems are incorporated to handle uncertainties in the complex environment and realize an intelligent cleaning motion.
(5) Working autonomously: Once the global task commands are entered by the user, the robot will keep itself attached to and move on the glass while accomplishing the cleaning tasks.
(6) Motion control function: To meet the requirements of all kinds of movement functions especially in crossing window obstacles, the motion control function is needed.
(7) Friendly Graphical User Interface (GUI): The robot can be controlled remotely by the operator to provide global task commands or emergency-type interaction through the GUI. All information while working will be sent back and displayed on the GUI at the same time during the phase of the feedback.
(8) Efficient cleaning: Efficient cleaning is the ultimate objective of cleaning robots. All functions above should serve this key point.

3 Overview of Robotic Systems

The family of the sky cleaner autonomous climbing robots for glass wall cleaning is totally actuated by pneumatic cylinders [3]. There are two reasons for designing full pneumatic cleaning robots besides advantages such as low cost and cleanliness. Firstly, the climbing robot can be made lightweight and dexterous using the pneumatic actuators. The lightweight construction is one of the most important specifications for devices working on high-rise buildings. Secondly, the movement driven by pneumatic actuators has the characteristic of passive compliance due to the compressibility of the air, thus making the robot safer than being driven by motors under the situation of interacting with the brittle glass.

The robots can walk and clean on a plane glass wall or on some special curved glass with a small angle between the glasses. A hose for water, a trachea

for pressured air, cables for power and control signals are provided from the supporting vehicle on the ground. Even if the robots are very complex and intelligent, the suitable height for working is below 50 m because the weight of the hoses has to be taken into account when robots are working in mid-air.

3.1 Sky Cleaner 1

The group in BeiHang University designed the first kind of cleaning robot named sky cleaner 1 for cleaning the glass top of the Beijing West Railway Station in 1997 [4] (shown in Fig. 1). The system consists of a robot, which is remotely operated and autonomously moves on glass walls to accomplish the cleaning work, and a support vehicle stationed on the ground providing electricity, air source and cleaning liquid. The lightweight robot can move on the surface of a slope up to 45 degrees in two perpendicular directions. Here the following unit for protection is not needed because the robot is safe on the target surface while cleaning.

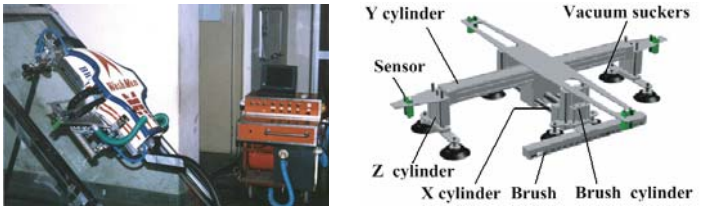


Fig. 1. A real photograph and an artistic impression of sky cleaner 1

It can automatically detect obstacles and cross the glass panes. The main body consists of two cross-connected rodless cylinders named X and Y cylinders. Connected at the ends of the X and Y cylinders are four shortstroke foot cylinders named Z cylinders, whose function is to lift or lower the vacuum suckers in the Z direction and support the body on the wall. Each group of vacuum suckers is arrayed in line, so the robot needs precise position control when it moves on the surface in order not to touch any obstacles. The practicable suction method on window glass is to use vacuum suckers, which are generally controlled by a vacuum ejector. The vacuum ejector (shown in Fig. 2) needs to share the compressed air source with other pneumatic cylinders, which simplifies the whole system. Outside the Z cylinders there are also two brush cylinders connected to the ends of the X cylinder, which lift and lower the brushes. The passive cleaning head (with brushes) is designed to be capable of supplying the detergent and collecting the drainage.

A PC is used as a console on the ground, and the on-board controller includes PC104 and PLC (shown in Fig. 3). The PC104 computer is the core controller and in charge of the global intelligent control such as planning and

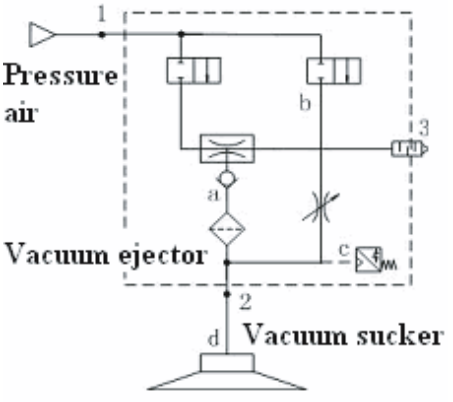


Fig. 2. Vacuum sucker

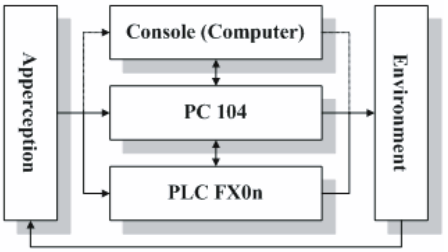


Fig. 3. The control system of sky cleaner 1

identifying the sensor inputs. PLC is the assistant controller that collects the internal switch sensor signals and actuates all the solenoid valves. The main specification features of sky cleaner1 are shown in Table 1.

However, the system has only limited dexterity and cannot work on a vertical wall. Because it has no waist joint, the robot is unable to correct the direction of motion. And the frequency for dealing with and crossing the obstacles is very high so that the cleaning efficiency is only about 300 $m^2/8$ hours. Due to the above-mentioned technical problems the robot will not really become a commercial product.

3.2 Sky Cleaner 2

The 1999 follow-up project to Sky cleaner 1 was aimed at developing a cost-effective, mobile and hassle-free robotic system for moving on vertical glass walls and high quality cleaning on the surfaces of high-rise curtain walls [5]. This project was based on the collaboration with the Centre for Intelligent Design, Automation & Manufacturing at the City University of Hong Kong.

Sky cleaner 2 is designed to be compact and easy to transport from place to place. The robot is featured with 16 suction pads which can carry a payload of

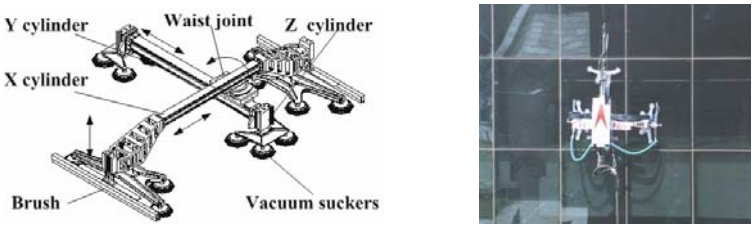


Fig. 4. An artistic impression and a photographic view of sky cleaner 2

approximately 45 kg including its body weight. Because of the special layout of the vacuum suckers, the robot can walk in all directions freely without attention to the seals. A pair of pneumatic cylinders provides both vertical and horizontal motion. A specially designed waist joint, located at the centre of the robot (as shown in Fig. 4), gives a turning motion to the robot. For a turning action, the position pin cylinder is aired to release the locking pin so that turning motions can be actuated by the waist-turning cylinder. The waist joint is used for the correction of inclination during the robot's movement. A relatively small degree of rotation (1.6°) per step is turned in the present stage.

Only an on-board PLC executes a sequence of solenoid valves on/off actions to perform commands that are sent by the operator through the PC console. A PWM method associates with a mechanical braking mechanism to achieve better position accuracy for placing the suction pads as close as possible to the window frames when a feedback signal has been detected by the ultrasonic sensors. The main specification features of the sky cleaner 2 are also shown in Table 1.

The robot is portable and cleaning efficiency is about 600 m^2/8 hours. But considerable stress was laid on weight reduction, the construction stiffness is somewhat low so that there is a little distortion while cleaning and climbing.

Table 1. Specification features of the robots

Type Capability	Sky Cleaner 1	Sky Cleaner 2	Sky Cleaner 3
Target character	Glass wall0° − 45°	Glass wall 0° − 90°	Glass wall 0° − 90° (with < 2° angle)
Efficiency(m^2/8 hours)	300	600	800–1000
Cross obstacles (mm^2): Height × Width	Window frame: 30 × 60	Window frame: 30 × 60 Seal: 1 × 20	Window frame: 10 × 60; Seal: 1 × 20
Weight (kg)	25	35	45
Body Mass (mm^3): Length × Width × Height	935 × 900 × 320	1220 × 1340 × 370	1136 × 736 × 377
Supply water(L/hour)	50 (reused)	50 (reused)	50 (reused)
Operators	1	1–2	1–2

On the other hand, the function of the following unit on the top of the building is simple and primary.

3.3 Sky Cleaner 3

Sky cleaner 3 is a real product designed for cleaning the complicated curve of the Shanghai Science and Technology Museum (shown in Fig. 5) in 2001 [6]. The building top is 40 m from the ground. From the left to the right its height gradually decreases. The total surface area of the outer wall is about 5000 m^2. Due to the special arc shape, each glass is connected to its surrounding glasses at an angle of 1.5 degrees. The robotic system consists of three parts: 1) a following unit; 2) a supporting vehicle; 3) the cleaning robot. The robot is supported above by cables from the following unit mounted on the top of the building. All following movements of the unit which protects against falling due to any type of failure are synchronized by the robot itself.

Fig. 5. Sky cleaner 3 cleaning the target glass wall

A turning waist joint actuated by a pendulum cylinder connects the X and Y cylinders. On opposite ends in the Y direction there are also four brush cylinders, which actuate the brushes up and down. An adaptive cleaning head is designed especially for effective, efficient and safe cleaning, equipped with a drainage collecting device. When the glass is being cleaned, the water is not allowed to drip down; it is firstly drawn off the glass wall through a vacuum pump on the robot. Then the water flows down because of the gravity and is collected on the supporting vehicle on the ground. At last the drainage is filtered, and then reused for cleaning. The ultrasonic sensors that can detect window obstacles are mounted on each end of the X and Y cylinders. The robot can both clean and walk on the glass walls automatically in the up-down direction as well as the right-left direction.

Because the glass walls of the Shanghai Science and Technology Museum have no window frames, there are supporting wheels near the vacuum suckers in the X and Y directions, which have been added to the mechanical construction to increase the stiffness. In order to move from one column of glass to another in the right-left direction, a specially designed ankle joint gives

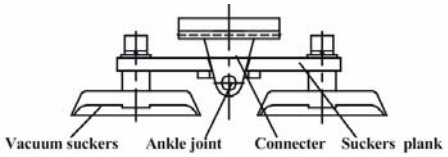


Fig. 6. The joint of the vacuum suckers

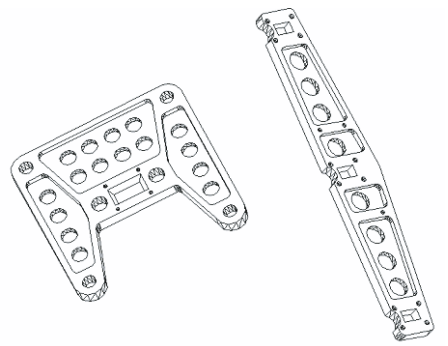


Fig. 7. Examples of mechanical planks

a passive turning motion to the suckers. This joint is located between the connecting piece which joins the vacuum suckers with the Y cylinder and the plank beneath it to which 4 vacuum-suckers are attached (shown in Fig. 6). In order to meet the requirements of the lightweight and dexterous movement mechanism, considerable stress is laid on weight reduction. All mechanical parts are designed specifically and mainly manufactured in aluminum. Figure 7 shows some examples of the mechanical planks.

A PLC is used for the robot control system (shown in Fig. 8), which can directly count the pulse signals from the encoder and directly drive the solenoid valves, relays and vacuum ejectors. FX2N-4AD which is added to the system can identify the ultrasonic sensor signals and other analog sensors. The control and monitoring of the robot is achieved through the GUI to allow an effective and user friendly operation of the robot. The communication interface between the PLC and the controller of the following unit is designed to synchronize the following movement of the cables. There are two kinds of external sensors on the robot: touchable sensors and ultrasonic analog sensors which are responsible for collecting information about the operational environment. The internal sensors are to reflect the self-status of the robot. For each active joint, there are limit switches to give the controller the position of the joint. On the joint where the accurate position is needed, the optical encoder is used. The vacuum sensors are used to monitor the vacuum condition of the suckers and provide information to determine whether the suction on the glass surface is stable.

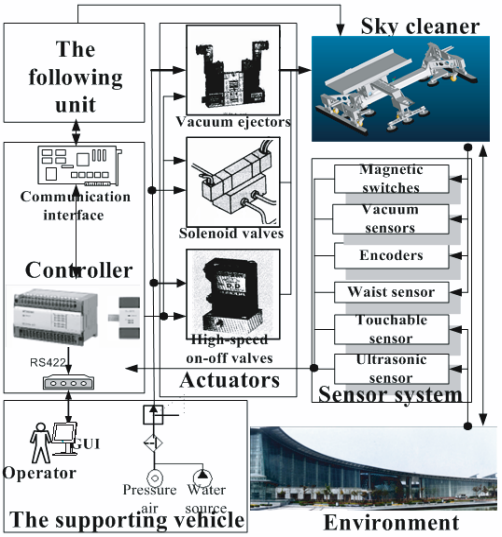


Fig. 8. Control system of the cleaning robot

4 Conclusion

This paper described a family of glass-wall cleaning robots totally actuated by pneumatic cylinders. The robots have to keep to and move on the arbitrarily sloped glass while accomplishing the cleaning tasks, they are portable, dexterous enough to adapt to the various geometries of the wall, intelligent enough to autonomously detect and cross the obstacles. The specification features of the robots are shown in Table. 1.

References

1. K. Berns, C. Hillenbrand, T. Luksch, "Climbing Robot for Commercial Applications – A Survey", Proceedings of the Sixth International Conference on CLAWAR 2003, Catania, Italy, September, 2003, pp. 771–776, 2003.
2. N. Elkmann, T. Felsch, M. Sack, J. Saenz, J. Hortig, "Innovative Service Robot Systems for Facade Cleaning of Difficult-to-Access Areas", Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland, October 2002, pp. 756–762, 2002.
3. H. Zhang, J. Zhang, R. Liu, G. Zong "A Novel Approach to Pneumatic Position Servo Control of a Glass Wall Cleaning Robot", 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2004, Sendai, Japan, Sept. 28–Oct. 2, 2004 (Accepted).
4. W. Wang, G. Zong, "Controlling and Sensing Strategy for Window Cleaning Robot", Hydraulics & Pneumatics, No.1, pp. 4–7, 2001.

5. H. Zhang, G. Zong, "Pneumatic Robot for Glass-Wall Cleaning", Chinese Hydraulics & Pneumatics, No. 11, pp. 5–8, 2001.
6. H. Zhang, J. Zhang, W. Wang, G. Zong, "Design of a Pneumatic Glass Wall Cleaning Robot for High-Rise Buildings", Proceedings of the ASER '04 2nd International Workshop on Advances in Service Robotics Stuttgart, Germany, May 21, pp. 21–26, 2004.

Design and Prototyping of a Hybrid Pole Climbing and Manipulating Robot with Minimum DOFs for Construction and Service Applications

M. Tavakoli[1], M.R. Zakerzadeh[2], G.R. Vossoughi[3], and S. Bagheri[4]

[1] Msc. Graduate student in mechanical engineering, Sharif University of Technology, Tehran, Iran,
m.tavakoli@mehr.sharif.edu
[2] Msc. Graduate student in mechanical engineering, Sharif University of Technology
[3] Associate Professor of mechanical engineering, Sharif University of Technology
[4] Assistance Professor of computer science, Sharif University of Technology

Abstract. In this paper, conception, design, modeling, and prototyping of a multi-task 4 DOF pole climbing/manipulating robot are discussed. The hybrid serial/parallel mechanism, with 2 translations and 2 rotations provides a good solution for a pole climbing and manipulating robot which can travel along tubular structures with bends, branches and step changes in cross section. It is also able to perform manipulation, repair and maintenance tasks after reaching the target point on the structure. After discussing conceptions of the mechanism, modeling and some aspects of the detailed design are presented. Then some of the issues with the prototyping of the robot mechanism are discussed.

1 Introduction

Climbing robots have received much attention in recent years due to their potential applications in the construction and tall building maintenance, agricultural harvesting, highways and bridge maintenance, shipyard production facilities... etc.

Use of serial multi-legged robots for climbing purposes requires greater degrees of freedom, without necessarily improving the ability of robots to progress in a complex workspace. It is also well known that serial configurations demand a greater amount of torque at the joints, thus calling for larger and heavier actuators and resulting in smaller payload to weight ratio, which is critical in climbing robots. In contrast using parallel platforms can result in the decrease of the weight/power ratio, thus allowing for larger payloads.

Earlier research in this area has focused on 6-DOF U-P-S (universal prismatic spherical) mechanisms (see [1] and [2]).

Saltarén et al. has modeled and simulated a parallel 6-DOF parallel robot with pneumatic actuators. The modeled robot has big payload capacity which is an important issue for industrial pole climbing robots [3].

Later R Aracil et al. fabricated a parallel robot for autonomous climbing along tubular structures. This robot uses the gough-stewart platform as a climbing robot. The platform actuators are 6 pneumatic cylinders with servo control. Their mechanism also used 6 cylinders as the grippers (3 cylinders for each gripper) using a total of 12 actuators not counting the actuators needed for the manipulator arm [4]. The mechanism is rather a complicated and has the ability of passing bends in any direction, making it suitable for traveling along trees and complex structures. So there is a need for a less complicated robot, which has the ability of traveling along human made and less complicated structures with minimum DOFs and minimum number of actuators. Furthermore using pneumatic cylinders in the mechanism has the problem of transferring compressed air from compressor to the cylinders.

But in recent years some industrial applications such as machine tools has resulted in more attention to parallel mechanisms with less than 6 degrees of freedom. Most of the research in recent years has focused on 3-DOF mechanisms [10, 11] and [12].

Traveling along a pole or tubular structures with bends and branches requires four degrees of freedom (2 translations and 2 rotations along and perpendicular to the tubular axis). These same degrees of freedom are also essential for most manipulation and repair tasks required in the pole climbing applications. More details on modeling of the mechanism and selection process of the planar parallel mechanism as the parallel part of the robot has been discussed in [5].

To the best knowledge of the authors, there is no 4-DOF mechanism providing 2 transnational and 2 rotational degrees of freedom suitable for such operations. The mechanism proposed in this paper takes advantage of a parallel/serial mechanism providing 2 degrees of translation and 2 degrees of rotation along the desired axes (which will be described later). The parallel/serial robots also have the advantages of high rigidity of fully parallel manipulators and extended workspace of serial manipulators [6]. Full kinematics analysis of the 4-Dof mechanism has been presented completely in [7].

The mechanism also takes advantage of a novel gripper design, making it suitable for safe pole climbing operations.

2 Concept

As mentioned earlier, locomotion along tubular structures, with bends and branches, requires a minimum of 4 degrees of freedom. These include Tz:

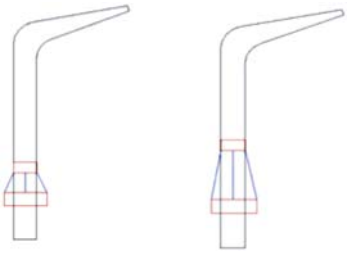


Fig. 1. Climbing Along the Pole

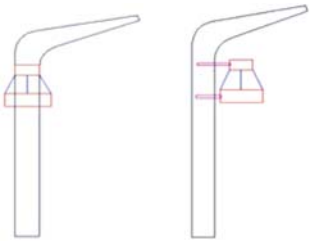


Fig. 2. Rotation Around the Pole axis

a translational degree of freedom for motion along the pole axis (Fig. 1), Rz: a rotational DOF for rotation around the pole (Fig. 2), Rx: a secondary rotational DOF for rotation around a radial direction of the pole (Fig. 3). Combination of the above 3 DOF with Tx a translational degree of freedom for motion along the pole radial direction provides the necessary manipulability to perform the many necessary operations after reaching the target point on the pole (i.e. repair, maintenance or even manufacturing operations such as welding) (Fig. 4).

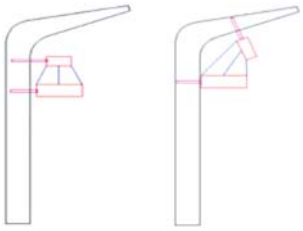


Fig. 3. Overtaking the Bent Section

Fig. 4. Robot Performing a Weldiing Operation

3 The Robot Design

The proposed pole climbing robot consists of three main parts (Fig. 5), which are: the 3-DOF planar parallel mechanism, the serial z axis rotating mechanism and the grippers.

Combining the 3-DOF planar parallel mechanism with a rotating mechanism around the pole axis provides two rotations and two translations, which is necessary to achieve the design objectives as explained in the last section. Furthermore, the linear cylinders used in the parallel manipulator are arranged to encircle the pole and thus reduce the grasp moments on the gripper.

One of the grippers is attached to manipulator, and the other one is attached to the base of the rotating platform. As a result, the grippers have four DOF with respect to each other, allowing for movements along the poles with different cross sections and geometric configurations.

3.1 The 3-DOF Planar 3-RPR Parallel 3-RPR Manipulator

A general planar three-legged platform with three-degree-of-freedom consists of a moving platform connected to a fixed base by three simple kinematics chains. Each chain consists of three independent one DOF joints, one of which is active [8]. Hayes et al. [9] showed that there are 1653 distinct general planar three-legged platforms with three DOF. For the proposed mechanism, the 3-RPR mechanism has been selected as the planar parallel part of the robot.

3.2 The Rotating Mechanism

The rotating mechanism consists of a guide, a sliding unit, a gear set and a motor. Figure 6 shows the guide and sliding unit. The guide is a T shape circular guide, which encircles the pole. The slider unit consists of a particular bearings arrangement, which can withstand the forces and torques generated during various maneuvers and maintains the robot stability in all its possible configurations. The slider holds the lower gripper and is driven by a motor with a simple gearing arrangement. By rotating the motor while keeping one of the grippers fixed (to the pole), the other gripper can rotate around the pole axis.

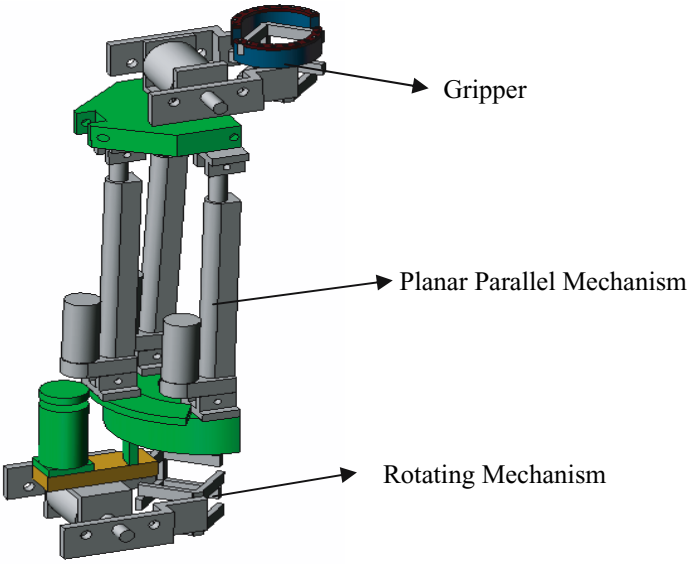


Fig. 5. The Pole Climbing Robot Model

3.3 The Grippers

The proposed gripper has a unique multi-fingered design, which is able to adapt to various pole cross sections and dimensions with only a single actuator. Each gripper consists of 2 v-shaped multi-fingered bodies, a double shaft motor, two right and left handed screws and two linear guides. Use of the Particular multiple finger arrangement not only increases the torque handling capability of the gripper but also improves the adaptability of the gripper to different pole dimensions without having fingers interfere/collide with each other. Using ballscrews with a friction coefficient of 0.1, and two linear bearing which stand the load of the robot during climbing process, the selected double shaft electric motor is rather small with respect to the weight of the robot. Figure 7 shows the fabricated gripper.

The combined actions of the various components in a typical pole climbing application are depicted in Figs. 8–10.

Table 1 shows the specifications of the prototype version and the estimated specifications of the industrial version of robot.

4 The Robot Prototype

Following the Kinematic analysis of the proposed mechanism [5, 7] a prototype unit was designed and built for a hypothetical municipal light bulb change operation.

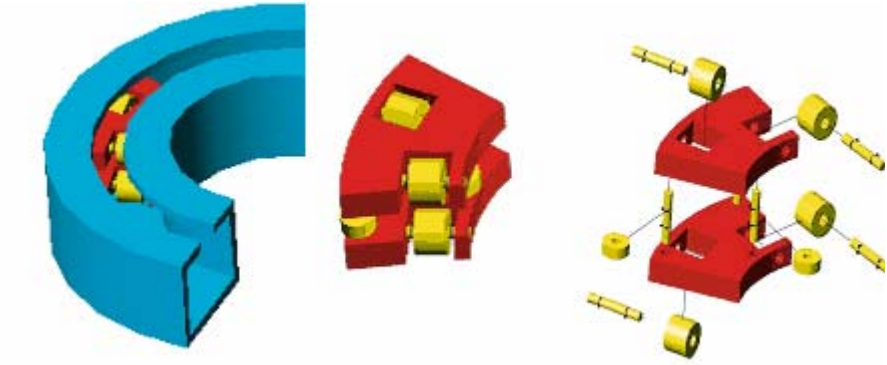


Fig. 6. The serial rotating mechanism

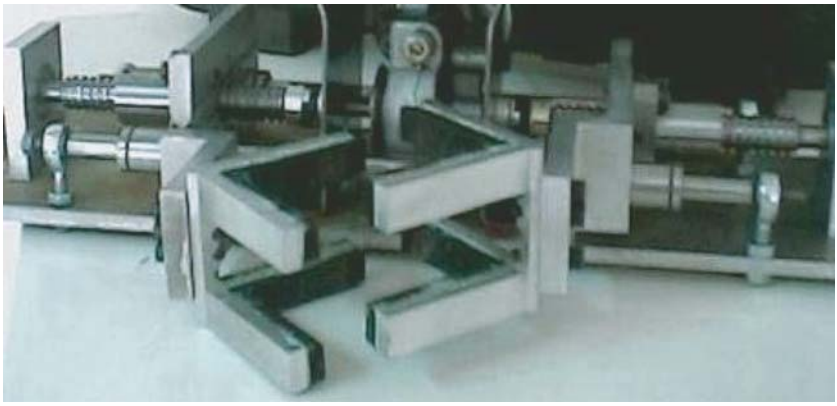


Fig. 7. The Robot Fabricated Gripper

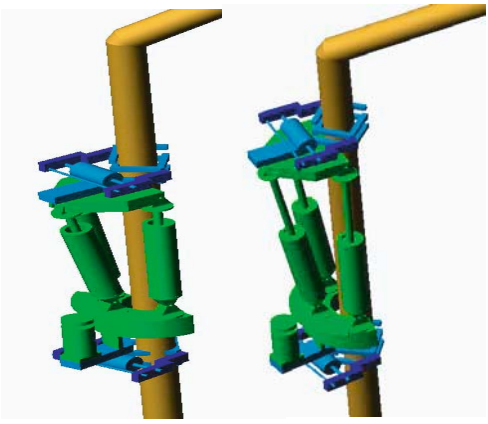


Fig. 8. The Robot Movement Along Pole Axis

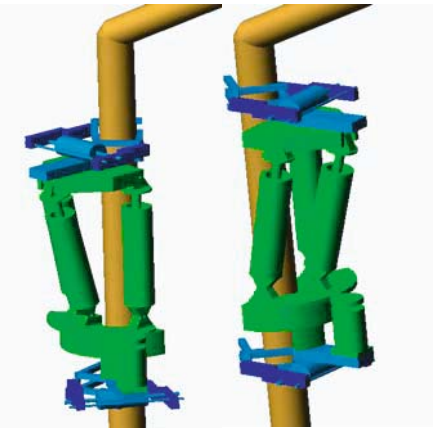


Fig. 9. The Robot rotation Around Pole Axis

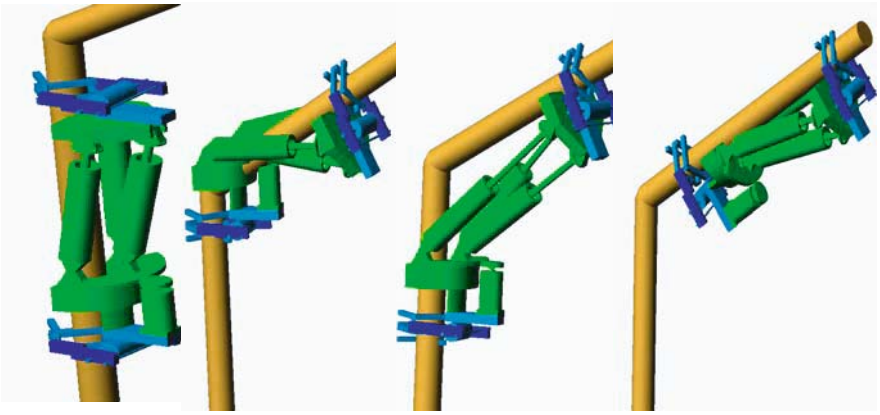


Fig. 10. The Robot Is Passing The Bent Section

The prototype of the robot weights 16 kg. The body of robot is fabricated from aluminum. The robot is driven by 3 dc motors and 3 electrical cylinders. Use of electrical cylinder rather than pneumatic or hydraulic cylinders simplifies the control of cylinders and increases the precision. Also there is no need of compressor or pump. This also eliminates hydraulic or pneumatic tubes, which are not safe in pole climbing applications.

The electrical cylinders weight 1.1 kg each. Each cylinder is able exert an 800 (N) force and has a stroke of 200 (mm) and speed of 0.6 (m/min).

Revolute joints in the planar parallel mechanism should be fabricated with a relatively high tolerance. Otherwise the planar parallel mechanism will either be overconstrained or exhibit extra degrees of freedom. In addition the assembly process precision is also highly important for the proper operation of the mechanism. To accommodate the light bulb change operations two

miniature grippers has been used. One to carry the new lamp, and the other to remove the old lamp. The grippers are two small pneumatic grippers. Also a small reservoir with capacity of 300 (cc) has been used. A dome remote control camera has been attached to the manipulator to assist in the bulb changing operation using a joystick as the robot remote control teach pendent unit. The camera has two degrees of freedom and can rotate around two perpendicular axes and is enveloped by a dome. Figure 10 shows the fabricated prototype.

5 Control of the Robot

As mentioned earlier, the prototype unit is actuated by 3 electrical cylinders and 3 dc motors. Each motor has a control driver board, which is attached to a central PC. The gripper motors are controlled using current feedback. Once the grippers touch the pole the current will increase to reach to a certain value thus exerting a proportional amount of force. Due to the large gear ratio of the gripper's dc motor the motor is not back-drivable. As a result in case of power failure, the gripper will continue to exerts the force continuously, making it fail-safe in case of power failure.

Electrical cylinders have combined of a dc motor, a gearing arrangement and an acme screw. Using a 500-pulse encoder on the shaft of the dc motor, the cylinders have a precision of 0.1 (mm) in linear movements. Also using a 100-pulse encoder on the shaft of the serial rotating mechanism's dc motor, the serial mechanism has a precision of 0.6 degree with the given gearing arrangement of the serial mechanism.

An array of touch switches, which have been assembled on the upper grippers, not only detect bend and other possible barriers on pole, but also can detect the angle of bent section of the pole with respect to the present direction of the robot gripper. This will allow the serial mechanism to rotate in a way that the robot mechanism is positioned properly for passing along the bent section.

The control system architecture includes a higher level inverse kinematic module and a lower lever PID based joint level position control system.

Table 1. Estimated Characteristic of The Prototype Version and the Industrial Version of Robot

	Number of Linear Actuators:	Number of Rotary Actuators:	Wt (Kg)	Dimensions (cm)
Prototype	3	3	16	18*25*60
Industrial	3	3	30	50*50*100

6 Conclusion

In this paper a solution to autonomous robot pole climbing problem is presented. Also, a unique multi-fingered gripper with the ability to adapt to various poles cross sections and dimensions with only a single actuator is also presented. Then some of the issues with the prototyping and control of the robot mechanism is discussed.

Acknowledgements

This work has been made possible by a grant from the Tavanir Electric research Center. The authors would like to thank Tavanir for supporting this research.

References

1. Merlet JP (2000) Parallel robots. Cluwer
2. Tönshoff HK (1998) A systematic comparison of parallel kinematics. Keynote in Proceedings of the First Forum on Parallel Kinematic Machines, Milan, Italy, August 31-September 1
3. Salataren R, Aracil R, Sabater JM, Reinoso O, Jimenez LM. Modeling, simulation and conception of parallel climbing robots for construction and service. 2nd international conference on climbing and walking robots, pp. 253–265
4. Aracil R, Saltarén R, Reinoso O (2003) Parallel robots for autonomous climbing along tubular structures. Robotics and Autonomous Systems 42, pp. 125–134, Novomber 2002
5. Vossoughi GR, Bagheri S, Tavakoli M, Zakerzadeh MR, Houseinzadeh M (2004) Design, Modeling and Kinematics Analysis of a Novel Serial/Parallel Pole Climbing and Manipulating Robot. 7th Biennial ASME Engineering Systems Design and Analysis, Manchester, UK July 19–22, 2004
6. Romdhane L (1999) Design and analysis of a hybrid serial parallel manipulator. Mechanism and Machine Theory 34, pp. 1037–1055.
7. Zakerzadeh ME, Vosoughi GR, Bagheri S, Tavakoli M, Salarieh H (2004) Kinematics Analysis of a New 4-DOF Hybrid (Serial-Parallel) Manipulator for Pole Climbing Robot. 12th conference of 12th Mediterranean Conference on Control and Automation
8. Gosselin CM, Sefrioui J, Richard MJ (1992) Polynominal solution to the direct Kinematic problem of planar three degree of-freedom parallel manipulators. Mechanism and Machines Theory, vol. 27 pp. 107–119
9. Hayes MJD, Hysty ML, Zsombor-Murray PJ (1999) Solving the forward kinematics of a planar three-legged platform with holonomic higher pairs. ASME J. Mech, vol. 121, pp. 212–219
10. Pierre F, Marquet F, Company O, Gil T (2001) H4 Parallel Robot: Modeling, Design and Preliminary Experiments. ICRA 2001: 3256–3261

11. Gosselin CM, et al. On the direct kinematics of general spherical 3-degree-of-freedom parallel manipulators. ASME Biennial Mechanisms Conference Proc., Scottsdale, Arizona, pp. 7–11
12. Tsai LW (1996) Kinematics of a three-dof platform with three extensible limbs. In Recent Advances in Robot Kinematics, pp. 401–410, Kluwer

Part X

Innovative Systems

Design and Control of a Manipulator for Landmine Detection

E. Garcia and P. Gonzalez de Santos

Industrial Automation Institute (CSIC) 28500 Madrid, Spain
egarcia@iai.csic.es

Summary. Antipersonnel mines infest fields all over the world. According to recent estimates, landmines are killing and maiming more than 2,000 innocent civilians per month [1]. The problem of landmine detection and removal requires the co-operation of various engineering fields. For this purpose, new technologies such as improved sensors, efficient manipulators and mobile robots are needed. This paper describes the configuration and control architecture of a scanning manipulator to detect antipersonnel landmines. The main features of the system that consists of a sensor head able to detect some kind of landmines and a manipulator to move the sensor head over large areas, conveniently sensorized to scan irregular terrains in the presence of obstacles are presented. Experiments show the performance of the whole system.

1 Introduction

Detection and removal of antipersonnel landmines is a serious problem of today. There is a real need to solve it, and solutions are being explored in different engineering fields. The finest solution is to apply fully automatic systems to this relevant task. However, this solution seems to be still very far away from succeeding. Efficient sensors and detectors are required to detect and, if possible, identify different mines.

Based on previous experience in robotics, the IAI-CSIC has configured the DYLEMA system based on a legged robot for detection and location of landmines (see Fig. 1) [2, 3]. DYLEMA is a Spanish acronym that means: Efficient Detection and Location of Antipersonnel Landmines. The main aim of this project is to develop a whole system to integrate relevant technologies in the fields of legged locomotion and sensor systems in order to identify the needs of this integration for humanitarian demining activities. This paper presents the on going results about the scanning manipulator carried by the walking robot. Section 2 describes the configuration of the manipulator and its sensor head. Section 3 presents the control architecture of the scanning

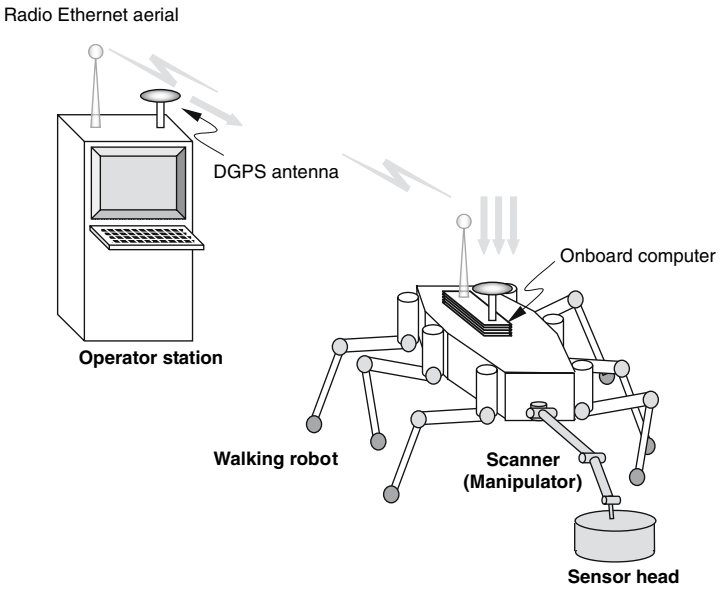


Fig. 1. DYLEMA demining system

manipulator based on sensor information and finally, some remarks and conclusions are given in Sect. 4.

2 Description of the Scanning Manipulator

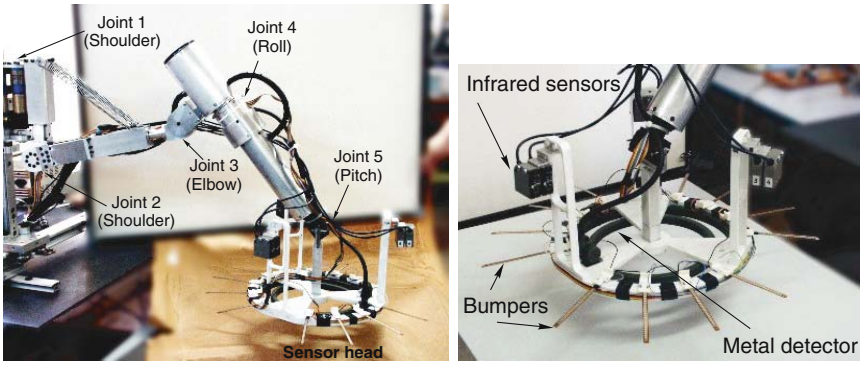
The scanning system is intended to detect antipersonnel landmines while a mobile robot traverses the infected area and is broken down into the following subsystems illustrated in Fig. 2(a):

2.1 Sensor Head

This subsystem contains a commercial mine detector and additional elements to detect the ground and objects in the way. The sensor head is configured to detect potential alarms, but also to allow the controller to maintain the sensor head at a given height above the ground using simple range sensors (infrared sensors coupled by pairs). Touch sensors are also provided to detect objects in the sensor head's trajectory. Figure 2(b) shows the sensor head.

2.2 Scanning Manipulator

The commercial mine detector is basically a local sensor. That means it is able to sense just one point. The efficiency of such a device can be improved



(a) (b)

Fig. 2. Scanning system: (**a**) Scanning manipulator; (**b**) Sensor head

by sweeping the sensor head through a large area. The simplest way to do this is by using a manipulator. Therefore, a 5-DOF manipulator is used to move the sensor head and to adapt the sensor head to terrain irregularities (see Fig. 2(a)).

3 Control Architecture

The scanning manipulator has to be controlled to seek for buried mines. The control architecture of the manipulator is shown in Fig. 3.

The manipulator-control architecture is a deliberative/reactive hybrid. Apart from the basic joint controller, three modules govern manipulator motion, which are:

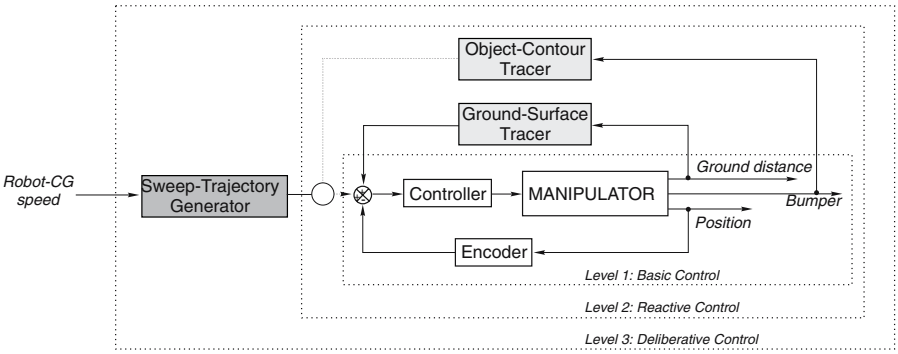


Fig. 3. The scanning-manipulator control architecture

The Sweep-Trajectory Generator: This is the deliberative module that generates the sweep trajectory to ensure complete coverage of the swept area.
The Object-Contour Tracer: This is a reactive module that moves the sensor head around an obstacle using information from the bumper.
The Ground-Surface Tracer: This is a reactive module that keeps the detector head at a constant distance from the ground, controlling its attitude as well.

3.1 Sweep-Trajectory Generator

The sweep-trajectory generator calculates the trajectory of the sensor head that ensures the complete coverage of the swept area. This is done by means of a crossed sweep, explained below.

The crossed sweep is the most efficient way to scan for buried mines. It avoids overlapping scanned areas while ensuring complete coverage. The sensor-head trajectory referred to the manipulator's base reference frame is depicted in Fig. 4(a). It scans an area that covers the width of the mobile robot that carries it ($2y_d$) and a length of x_d along the x-axis. To coordinate the manipulator's motion with the walking robot's motion, certain conditions must be met:

- The walking robot moves along the x' axis in an external reference frame $\{x', y'\}$ at a constant speed of v_{CG}. The external reference frame is centered initially at the robot's CG position, and the x' axis lies along the robot's longitudinal axis.
- The crossed-manipulation motion, combined with the CG robot motion, results in zig-zag trajectories parallel to the x' and y' axis as Fig. 4(b) shows.
- The complete crossed manipulator's motion takes place in a walking-robot gait-cycle time (T_c).

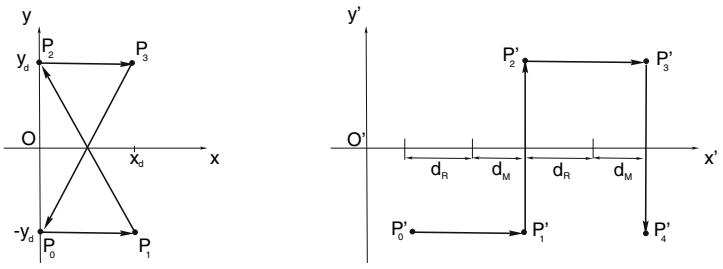


Fig. 4. Crossed-sweep trajectory: (**a**) In the manipulator's reference frame; (**b**) In the external reference frame

To accomplish the first and second conditions, the diagonal span of the manipulator trajectory from P_1 to P_2 (see Fig. 4(a)) needs to travel back the same distance x_d traveled by the robot in the same interval $(t_2 - t_1)$, where t_i is the time at when P_i is reached (for $i = 1..4$), that is:

$$x_d = v_{CG}(t_2 - t_1) \; . \tag{1}$$

To accomplish the third condition, the time interval of each trajectory span needs to be a fraction of the robot's cycle time. Additionally, to ensure complete coverage of the swept area, the distance from $P_0^{'}$ to $P_1^{'}$ has to equal the sensor head's diameter. So we have to determine the time intervals of the trajectory spans so that:

$$Dist(P_0^{'}, P_1^{'}) = D \; , \tag{2}$$

where D is the sensor head's diameter. Let us name $T_1 = t_1 - t_0$ and $T_2 = t_2 - t_1$. Let us also name d_R the distance traveled by the walking robot during T_1 and d_M the distance traveled by the manipulator during T_1. The problem equations are:

$$x_d = v_{CG} T_2 \tag{3}$$

$$D = d_R + d_M \tag{4}$$

$$T_c = 2(T_1 + T_2) \tag{5}$$

As a solution of this system of equations, we finally obtain the time intervals T_1 and T_2 that assure complete coverage of the scanned area:

$$T_1 = \left(1 - \frac{x_d}{D}\right) \frac{Tc}{2} \tag{6}$$

$$T_2 = \frac{x_d}{D} \frac{T_c}{2} \; . \tag{7}$$

Now, the manipulator trajectory can be described in four steps, starting at $P_0 \equiv (0, -y_d)$:

Step 1: Linear trajectory from $P_0 \equiv (0, -y_d)$ to $P_1 \equiv (x_d, -y_d)$ in the x-direction, that is

$$x = \frac{x_d}{T_1}(t - t_0); \quad y = -y_d \; . \tag{8}$$

Step 2: Linear trajectory from $P_1 \equiv (x_d, -y_d)$ to $P_2 \equiv (0, y_d)$ such that

$$x = x_d - \frac{x_d}{T_2}(t - t_1); \quad y = -y_d + \frac{2y_d}{T_2}(t - t_1) \; . \tag{9}$$

Step 3: Linear trajectory from $P_2 \equiv (0, y_d)$ to $P_3 \equiv (x_d, y_d)$ in the x-direction, that is

$$x = \frac{x_d}{T_1}(t - t_2); \quad y = y_d \; . \tag{10}$$

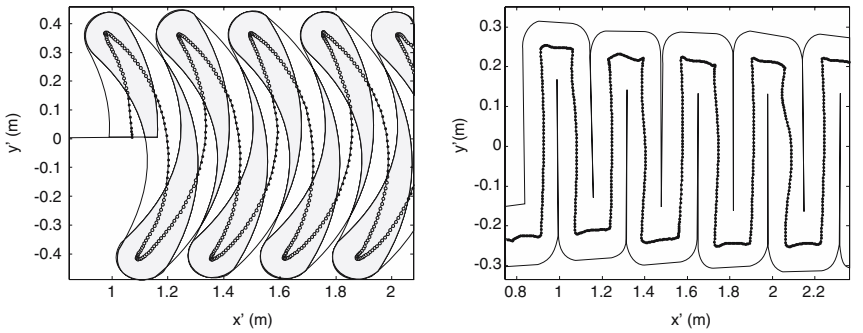


Fig. 5. Scanned area in the external reference frame using (**a**) a circular sweep; (**b**) a crossed sweep

Step 4: Linear trajectory from $P_3 \equiv (x_d, y_d)$ to $P_0 \equiv (0, -y_d)$ such that

$$x = x_d - \frac{x_d}{T_2}(t - t_3); \quad y = y_d - \frac{2y_d}{T_2}(t - t_3)\ . \tag{11}$$

This crossed-manipulation motion, combined with the CG robot motion, results in zig-zag trajectories parallel to the robot's x' and y' axes in the external reference frame as Fig. 4(b) shows. The crossed-sweep trajectory is generated as a function of the speed of the walking robot's CG (v_{CG}), and one crossed sweep is performed per robot cycle. To show the advantages of applying the proposed sweep motion, a comparision between a circular sweep and a crossed sweep has been performed. Figure 5(a) shows the experimental results of applying a complete-coverage circular sweep and compares it with the results of applying the crossed sweep (Fig. 5(b)) using the SILO6 robot. The dotted line shows the trajectory of the sensor head's base center, while solid lines depict the total scanned area. As can be observed, in order to obtain complete coverage with a circular sweep, overlapping areas must exist (shaded areas in Fig. 5(a)), which make the method inefficient. However, the crossed sweep generates complete-coverage trajectories without overlapping areas and is therefore demonstrated to be an efficient scanning method.

3.2 Object-Contour Tracer

An array of 12 bumpers around the sensor head detects obstacles in the mine field (see Fig. 6). This is done by moving the manipulator such that the bumper constantly detects the obstacle while the x coordinate (in the field reference frame) of the manipulator position varies. This module and the sweep-trajectory generator are mutually exclusive; that is, only one of them can be active at a time.

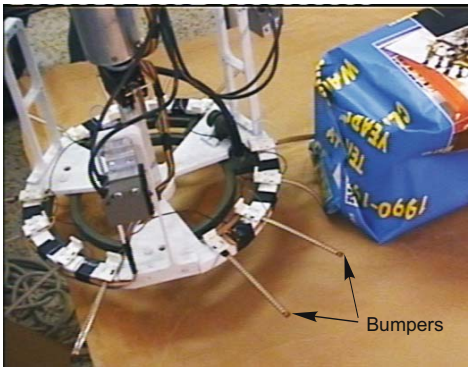


Fig. 6. Experimental obstacle detection

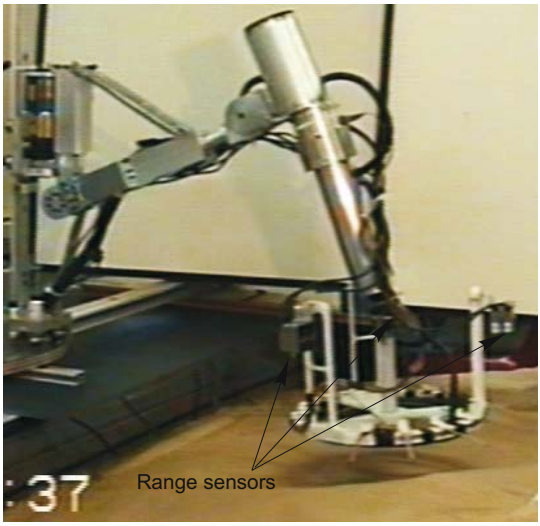


Fig. 7. Experimental ground adaptation

3.3 Ground-Surface Tracer

While performing a crossed-sweep motion, the sensor head adapts to the terrain profile reactively. Three range sensors arrayed at 120^o-degree intervals around the sensor head (see Fig. 7) measure the sensor-head distance and orientation to the ground. The ground-surface tracer computes the error from the desired distance and orientation and modifies the reference trajectory to compensate for it.

4 Conclusions

Human operators handling manual equipment are, at present, detecting and locating antipersonnel landmines. However, human community can obtain many benefits by the robotization of this activity. New sensors are required in order to detect landmines efficiently, but existing sensors can be carried by robots over infested fields.

This paper addresses the development of a manipulator able to scan areas with a sensor head based on a metal detector and other sensors required to scan irregular ground in the presence of obstacles. The scanning system has been described and the control architecture has been presented. This architecture allows the manipulator to adapt to terrain irregularities and avoiding obstacles in a reactive manner. A new sweep method enabling the scanning manipulator to search efficiently for buried mines has also been proposed. The improvement obtained with the crossed-sweep algorithm has been proved experimentally.

The scanning manipulator will contribute to the autonomous antipersonel-landmine detection process.

Acknowledgements

This work has been partially funded by CICYT (Spain) through Grant DPI2001-1595. The first author is supported by a postdoctoral CSIC-I3P contract granted by the European Social Fund.

References

1. P. Kopacek, "Demining robots: A tool for international stability," in *15th Triennial World Congress IFAC*, 2002, pp. 1–5, Barcelona, Spain.
2. E. Garcia, J. Estremera, and P. Gonzalez de Santos, "A control architecture for humanitarian-demining legged robots," in *Proc. Int. Conf. Climbing and Walking Robots*, September 2003, pp. 383–390, Catania, Italy.
3. P. Gonzalez de Santos, E. Garcia, J.A. Cobano, and A. Ramirez, "Silo6: A six-legged robot for humanitarian de-mining tasks," in *10th International Symposium on Robotics and Applications, World Automation Congress*, June 2004, Seville, Spain.

Interactions Between Human and Robot – Case Study: WorkPartner-Robot in the ISR 2004 Exhibition

S. Ylönen, M. Heikkilä, and P. Virekoski

Automation Technology Laboratory, Helsinki University of Technology, P.O. Box 5500, 02015 HUT, Finland
sami.ylonen@hut.fi

Abstract. WorkPartner is a mobile interactive service robot designed for lightweight outdoor tasks in co-operation with humans. WorkPartner participated in the ISR 2004 (35th International Symposium on Robotics) exhibition on CLAWAR stand in Paris 22–26 March 2004. During the five days a lot of information was collected about human-machine interaction. The robot communicated with humans using speech and gestures, and observed environment using vision system. The visitors seemed to get a very humane impression of the robot.

1 Introduction

Mobility of WorkPartner is based on a hybrid system, which combines the benefits of both legged and wheeled locomotion to provide at the same time good terrain negotiating capability and a large velocity range (see Fig. 1). The working tool is a two-hand human-like manipulator that can be used for manipulation or handling of tools. The user or operator can be physically present on the same site as the robot and communicate with it using speech and gestures, or he can use telepresence from another place and communicate via Internet.

The ultimate goal is a highly adaptive service robot. Some possible work tasks for the WorkPartner: garden work, guarding, picking trash, transferring lightweight obstacles, environment mapping. The WorkPartner project, its mechatronics design, hybrid locomotion and control system have been reported in six previous CLAWAR conferences [1–6]. The purpose of present paper is to continue the series by introducing development that is done for its human interaction. The project is public and can be followed on the Web site:

www.automation.hut.fi/IMSRI/workpartner.

Fig. 1. WorkPartner at the exhibition stand

The system diagram in Fig. 2 summarises the hardware structure of WorkPartner indicating also the main software environment and communication interconnections between the subsystems [7]

2 Interaction

WorkPartner was demonstrated on CLAWAR stand in the ISR 2004 exhibition. Demonstration consisted of movements and interaction like speech, gestures and eye contact. The robot was driven near the humans and it offered candies (Fig. 3). It was working under teleoperation. The operator drove the robot using a joystick and a teleoperation device for the humanlike manipulator (Fig. 4).

Appearance is very important for a robot that works interactively with humans. Many research organisations and companies are developing robots that look like humans, for example Honda has developed Asimo robot [8]. WorkPartner has a human like upper body, but for greater mobility, it has a platform with four wheeled legs.

2.1 Communication Methods of the Robot

Usability of the robot is one of the most important research areas in the WorkPartner project. A robot that is designed to be working interactively with humans has to be easy to use by different people. Therefore it should be able to communicate in a way that is natural for humans. WorkPartner uses speech, gestures and eye contact.

Speech

Different things can be expressed easily by speech and it is the most natural way of communication between humans. In the exhibition WorkPartner had

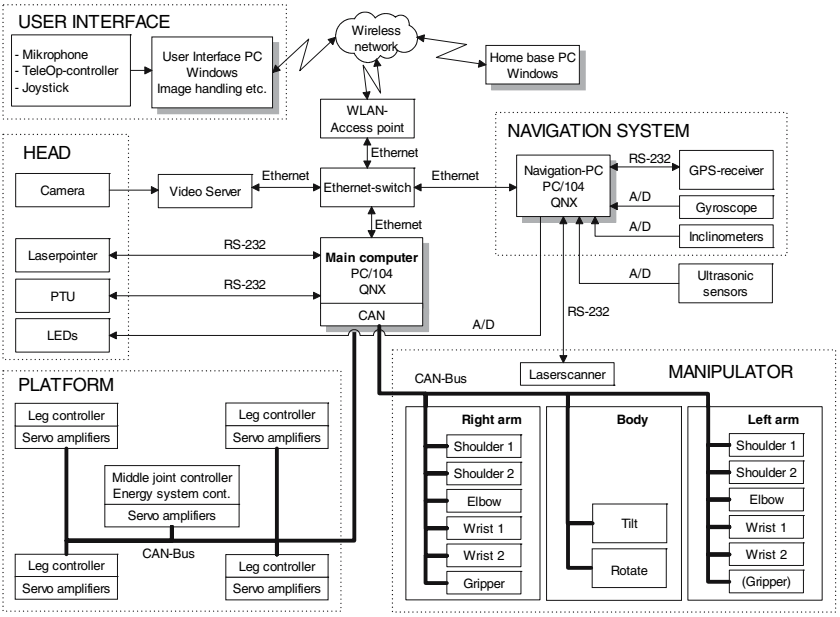


Fig. 2. System diagram of the WorkPartner hardware

Fig. 3. WorkPartner offering candies for the exhibition visitors

Fig. 4. Teleoperation of the manipulator

a couple of preprogrammed sentences that it spoke by a speech synthesis program. For example it said, "Take some candies", "Greetings from Finland" and "What is your name".

Gestures

The robot can make many different gestures using its arms, legs and head. For example it did dancing movements with the body, waved its hand and nodded its head.

Eye Contact

WorkPartner has a CCD-camera in its turning head. It was in color tracking mode. Human face color was selected for the tracking. This way the robot was looking for the faces of the visitors and an eye contact was formed between human and the robot.

2.2 Human Reactions

This chapter describes the most important results that we got in the exhibition. It was interesting to see how differently people reacted to the robot. Most humans were very interested, some were cautious and some were very familiar with it. Some communicated to the operator and some to the robot. Some acted with the robot like it was another human. Overall, the visitors seemed to get a very humane impression of the robot.

Humans were looking at the camera of the robot and most were smiling. This was like eye contact between humans and the robot. Figure 5 shows some human reactions seen by the camera of WorkPartner.

People said "thank you" to WorkPartner quite often after it had given candies. Gestures toward the robot included smiling, shaking hands, blowing

Fig. 5. Human reactions seen by the camera of the robot

a kiss and even kissing the head of the robot. It looks that children humanize the robot more often than adults. Women regarded to WorkPartner more emotionally and men more technically.

Taking Candies

Main work task of WorkPartner was giving candies for the visitors. The candies were in a paper bag that was held by the robot. Only a few visitors approached WorkPartner initially, but when it said, "Take some candies" and handed the bag towards them, many were encouraged to come and take sweets.

2.3 Media Interest

WorkPartner was filmed to the news of four main TV-channels in France. Some magazines were interested in WorkPartner too.

3 Future of the WorkPartner-Robot

The goal of the project is to have WorkPartner communicate autonomously with humans using natural methods such as speech, speech recognition and gestures and also to learn tasks. It has been a research project, but we are looking for partners to cooperate in prototyping and commercializing WorkPartner and/or its component systems.

4 Conclusions

The interactivity of the robot has a great impact on the behavior of humans. The robot appeared to be looking at humans by turning its head, made gestures by its arms and spoke. This induced some people to actually speak to the robot instead of its operator and also establish an eye contact with it.

It was a very nice experience to participate in the big exhibition with WorkPartner. We got much information concerning interactions between humans and robots. This information is utilized in our research. Collecting of interaction experiences will continue in the future.

References

1. Leppänen I, Salmi S, Halme A (1998) WorkPartner – HUT-Automations new hybrid walking machine, CLAWAR'98, Brussels 1998
2. Halme A, Leppänen I, Salmi S (1999) Development of WorkPartner-robot – design of actuating and motion control system, CLAWAR'99, Portsmouth 1999
3. Halme A, Leppänen I, Salmi S, Ylönen S (2000) Hybrid locomotion of a wheel-legged machine. CLAWAR'2000, Madrid 2000
4. Halme A, Leppänen I, Montonen M, Ylönen S (2001) Robot motion by simultaneous wheel and leg propulsion. 4th International Conference on Climbing and Walking Robots, Karlsruhe 2001
5. Ylönen S, Halme A (2002) Further development and testing of the hybrid locomotion of WorkPartner robot. 5th International Conference on Climbing and Walking Robots, Paris 2002
6. Luksch T, Ylönen S, Halme A (2003) Combined Motion Control of the Platform and the Manipulator of WorkPartner Robot, 6th International Conference on Climbing and Walking Robots, Catania 2003
7. Halme A, Leppänen I, Suomela J, Ylönen S, Kettunen I (2003) WorkPartner: Interactive Human-Like Service Robot for Outdoor Applications, International Journal of Robotics Research, Vol 22, July–August, 2003
8. Sakagami Y, Watanabe R, Aoyama C, Matsunaga S, Higaki N, Fujimura K (2002) The intelligent ASIMO: System overview and integration, IROS 2002, International Conference on Intelligent Robots and Systems, EPFL, Lausanne, Switzerland, pp. 2478–2483, October 2–4, 2002

Robust Platform for Humanitarian Demining

Lino Marques, Svetlana Larionova, and A.T. de Almeida

Institute of Systems and Robotics, University of Coimbra, Portugal
{lino,sveta,adealmeida}@isr.uc.pt

Abstract. This paper describes an advanced multisensor demining robot. The mechanical structure of the robot and its hardware and software architecture are described. Special structures of software and sensor fusion are proposed for practical implementation of the automated demining tasks. Identification results of possible regions-of-interest (ROI) with the proposed demining sensor fusion algorithm are shown.

1 Introduction

Landmines affect almost every aspect of life in states recovering from conflict. There are still more than 100 million mines spread across more than 60 countries [1]. Manual demining is a slow, dangerous and costly process that could be extraordinarily speed-up by the assistance of robotic technologies. There are two main problems that need to be overcome in order to implement autonomous demining robots: the first one is the development of platforms able to scan all points of a suspicious area with a landmine detecting system; the second aspect is the development of landmine detecting systems reliable enough to find all landmines in the terrain. A legged platform constructed with standard off-the-shelf pneumatic elements provides the required flexibility and roughness to move across natural rough terrains while guaranteeing the necessary availability, even in developing countries, of any critical motion element that might become damaged during the robot operation. Although the intense research in the subject of sensors for demining, it does not exist yet a single system able to surely detect any landmine. A possible solution to improve this scenario consists in using different technologies of commercial sensors and combining the information provided by all of them in order to obtain a better estimate of mine existence. Landmines have physical, chemical and geometric attributes. A common approach to identify landmines with multiple sensors consists in mapping the output of each sensor to a space referenced image and subsequently analyse each image. The on-line

construction of those images and the pre-processing of the images in order to identify areas with possible landmines are addressed in the last part of the paper.

2 Robot Design

The robot transport system is based on a simple 8-legs structure using pneumatic drive elements. The robot has robust design and can carry demining equipment up to 100 kg over rough terrains [5]. The mechanical design of the robot platform was optimised to perform linear movements across rough terrains, allowing to scan large areas with a landmine detecting block. Due to the adaptive possibilities of the pedipulators to obstacles, the robot can adjust the working position of the demining sensors while searching landmines. The detection block consists of a metal detector and an active infrared system.

The control architecture is hierarchical with high-level functions located in an on-board Linux PC connected by a serial link to the platform low-level controller. The high-level module is responsible by user interfacing, platform monitoring, task specification, sensor fusion and path planning. The low-level controller is responsible for trajectory control and demining sensors data acquisition.

2.1 Mechanical Structure

The design of the transport module of the robot is shown in Fig. 1. It consists of longitudinal and latitudinal 200 mm stroke pneumatic cylinders whose bodies are connected symmetrically allowing to perform perpendicular motions. Each transport cylinder has two pedipulators that are fixed at both ends of the piston rods. Each pedipulator consists of a lifting cylinder of 150 mm stroke to overcome stone obstacles of that magnitude and a foot with toothed contact surface to improve robot climbing possibilities. The detection block is connected to the front part of the robot. The position of each transport cylinder is measured by ultrasound distance sensors. These sensors provide position feedback for sub-stroke actuating motions and to measure the position of the sensor block during field scanning. The valve units are placed at both sides of the robot providing minimum tubing length. They are supplied from a supply rotation block connected to the main air pressure line. The rotation block allows free rotation of the robot in relation to the umbilical cord with the air supply pipe. The on-board control computer is placed in the center of the platform. An inertial unit with 3D accelerometers and 3D electronic compass is installed in the platform body to provide orientation information. The platform localization is obtained by optical triangulation to fixed beacons.

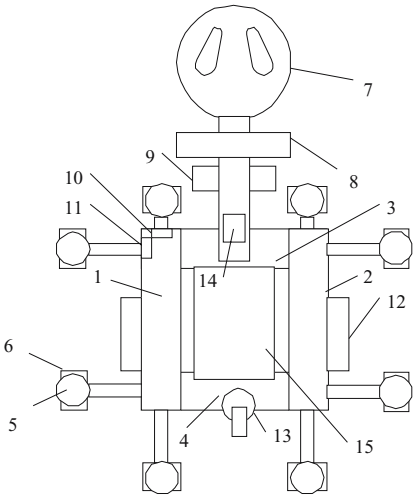


Fig. 1. Design of the transport module of the robot: 1, 2 – longitudinal pneumatic cylinders; 3, 4 – latitudinal pneumatic cylinders; 5 – lifting cylinder; 6 – foot; 7 – metal detector; 8 – IR detector; 10, 11 – linear position sensors for transport cylinders; 12 – valve units; 13 – supply rotation block; 14 – inertial unit; 15 – on-board computer

2.2 Control Hardware

The control hardware of the robot is modular (Fig. 3). Interfacing to the robot sensors and actuators is performed by separated modules connected via a SPI network to the low-level controller based on a HC12 Motorola micro controller. A Modbus-based protocol is used to connect the low-level controller to the on-board PC.

2.3 Motion Control

The main task of the robot is to find all landmines situated in a specified area. To obtain the required information a complete coverage of this area should be performed mapping the sensor values spatially. The robot can cover a minefield without obstacles using only linear motions. The transport and scanning trajectories of the robot along a minefield with obstacles demands to use rotation of the robot in order to orientate it to move around obstacles. It is possible to perform four different kinds of robot rotations, all of them based on a frictional principle. The rotations are mostly used to correct the trajectory. A drawback of using inexpensive pneumatic technology is the difficulties in accurately positioning the platform. A consequence of this problem is irregularity in the grid of sensor readings (Fig. 2(b)). The irregular grid is converted to a regular one by linear interpolation. During X-direction movement the samples of sensor values are taken continuously together

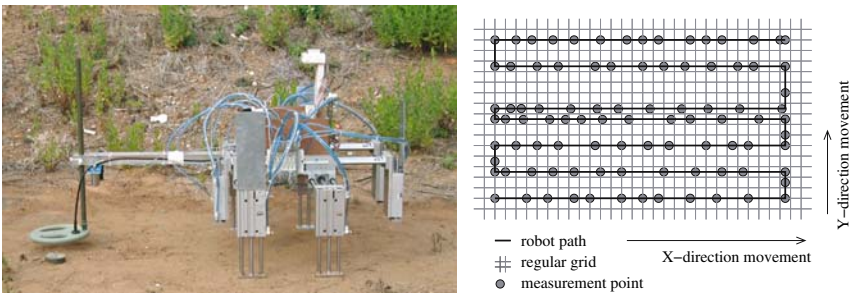


Fig. 2. Pneumatic demining robot (**a**) and a path obtained with it (**b**)

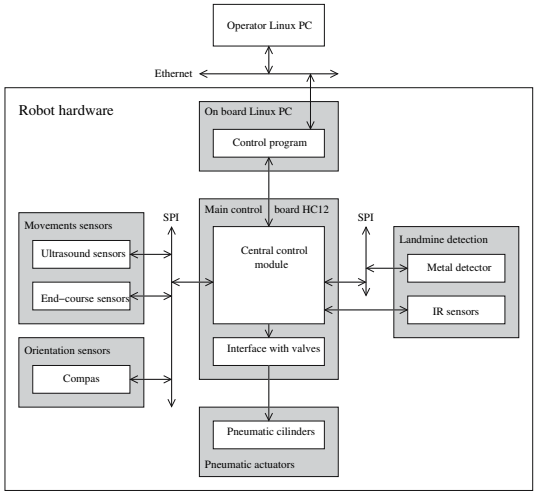


Fig. 3. Hardware structure

with the current coordinates of the sensor block. Y-direction movement is performed with a certain step (approximately 3 cm) by pulsing pneumatic valves at low frequency.

3 Software Arquitecture

The demining robot software must be able to perform variety of tasks from low-level drive control to a convenient representation of landmine detection results for the operator. To achieve the best performance, these tasks have to be splitted among different programs that communicate with each other. The software of the robot is distributed among its hardware. The main parts of it are: *Low-Level Control Program* running on a HC12 micro controller, *Control*

Program running on the robot on-board Linux PC and *Graphical Interface* running on the operator Linux PC.

3.1 Low-Level Control Program

The *Low-Level Control Program* performs basic operations to provide an interface to the sensors and actuators of the robot. It implements basic commands and performs simple trajectory control motions through medium level commands like: **step** in a certain direction, **scan** an area, **update** data from sensors, etc. The communication protocol that connects *Low-level Control Program* and *Control Program* is based on Modbus. Thus the robot is considered as a Modbus device with its internal memory structure. Commands "read multiple register" and "write multiple register" are implemented for updating data about the robot sensors and actuators. Other commands (e.g., step, scan, etc.) are "user-defined" according to the Modbus specification.

3.2 Control Program

The *Control Program* needs a special design because it deals with tasks with very different time constraints that reflect the specificity of automated demining: there are time critical modules, (e.g., controlling the robot and acquiring the sensors data) and time consuming modules (e.g., sensor fusion processing). Sensor fusion algorithms require a large amount of processing time as they use image processing and classification techniques. This does not allow to use a standard data-flow scheme as a base for the *Control Program* because the time consuming modules would slow down the rest of the time critical program. Each process that requires large amounts of processing time should be implemented in a separated thread. Simultaneously a time consuming algorithm (e.g., mapping, sensor fusion algorithm) must have access to the real-time sensor data, thus a suitable communication between slow and fast parts of the program should be available. Practical implementation of this software is done in C++ using standard Linux threads architecture and mutex mechanism. If the object implements a time-consuming algorithm then it is divided into two parts: the *slow part* can take as much processing time as it is required by the algorithm and the *fast part* provides access to the real-time data and the time it can spend is limited. Each object gets control for its *fast part* in the *working loop* of the main program thread. Besides path planning, sensor fusion and control algorithms, this module assures the communication with the *Graphical Operator Interface* and sensor data archiving during operation.

3.3 Graphical Operator Interface

A *Graphical Operator Interface* (Fig. 4) provides visual information to the operator about the robot current state and allows him to specify commands to

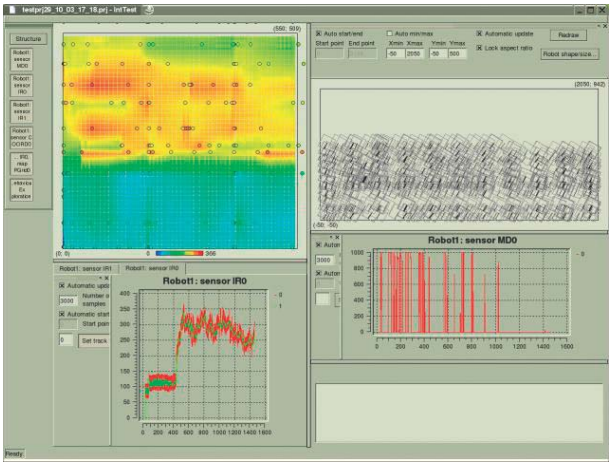


Fig. 4. Graphical Operator Interface

the robot. This module is developed as an easy configurable tool which is able to perform various tasks: graphical representation of the current robot position and sensors values, control of the robot by commands and joystick, review of the archived data obtained during past robot operation, representation of the internal information for debugging of the algorithms, interface to the unified landmine signatures database, interface for training and testing of sensor fusion classifiers.

4 Sensor Fusion for Landmine Detection

One of the main problems of a demining robot is the lack of landmine sensing devices able to provide enough information to guarantee the required clearance rate with a low false alarm rate. Using the current sensing technologies, it is considered that a combination of several different sensors and sensor fusion techniques are required to provide acceptable results. A sensor fusion system for demining robot must act as a single sensor which automatically provides online information. This will allow the robot to correct the planned path and avoid the danger as much as possible. In this work a feature-level sensor fusion technique is used as a base for the developed algorithms. Landmine detecting sensors allow to sense a heterogeneity of some physical characteristic against a background. This heterogeneity can be caused by a landmine, by other artificial objects (e.g., clutter), by natural objects (e.g., stones) or by changes in the environmental conditions (e.g., sun shining and humidity). In order to distinguish a landmine from other objects, its particular fingerprint in spatially mapped sensors readings should be identified. According to this specificity the following sensor fusion methodology is proposed: the

process is divided into two classification steps, the first step separates all objects from background, then during the second step a feature-level fusion allows to separate landmines from other objects (using classes *Mine* and *Another Object*). This methodology allows to focus the classifier only on the important features which distinguish a landmine from other objects. Moreover it allows to integrate data from different experiments proposing a more unified representation of landmine signatures. Databases of experimental data are already freely available through the Internet for public usage [2]. The proposed methodology allows using of this data to create a database of unified landmines signatures stored as collections of ROIs obtained from the ROIs extraction and objects association algorithms.

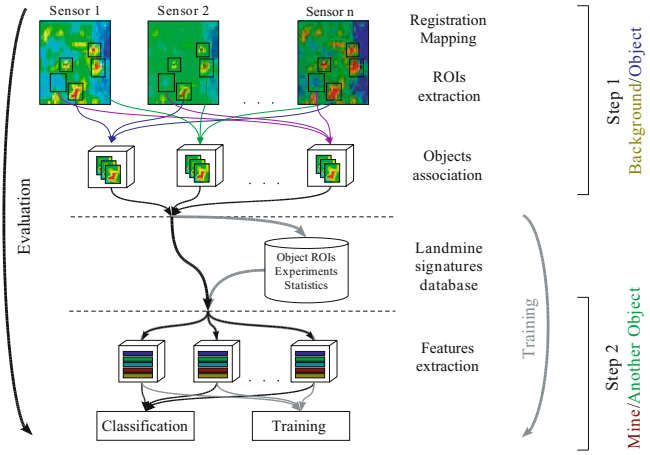
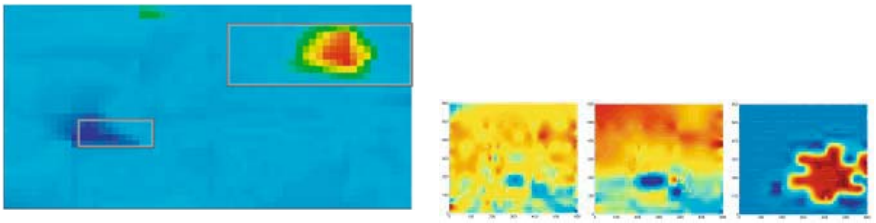


Fig. 5. Sensor fusion process

More detailed this process can be divided into the following stages:

- step 1: registration, filtering and mapping; Regions-Of-Interest (ROIs) extraction (segmentation); objects association;
- step 2: features extraction; classification;

The whole sensor fusion process is presented in Fig. 5 ([3]). First, the sensed data are mapped into a regular grid-map in order to build an image where ROIs might be found. The ROIs extraction algorithm is base on the assumption that ROIs can be found in places where heterogeneities in sensor readings are situated (a detailed explanation of the algorithm is presented in [4]). Then the ROIs from different sensors which represent the same object should be combined together for the further features extraction and classification (by other words, each ROI should be associated with some object). The objects association algorithm used in this work utilizes one-to-many association principle and uses the distance between ROIs as one of the



(a) (b)

Fig. 6. Result of ROIs extraction from experimental data obtained by one IR sensor of the robot (**a**) and Example of landmine signature measured by the robot sensors, from *left* to *right*: 6.5–14 μm IR, 8–14 μm IR, metal detector (**b**)

features for the classification. For each collection of ROIs obtained by the objects association algorithm a set of features is calculated. Two types of features are used: ROI features (statistical characteristics calculated for each ROI) and collection features (reflect the relationships between the ROIs). At the last classification is performed to distinguish landmines from other objects. The developed algorithms are included into the control software of the demining robot allowing its automated functioning.

5 Conclusions and Results

The proposed design of the demining robot is experimentally implemented (Fig. 2(a)). It was tested in real conditions of rough terrain, sample data obtained from antipersonnel landmines is presented in Fig. 6(b). A whole sensor fusion process was implemented and tested with the public experiment data [3]. The results of ROIs extraction from real data obtained by the robot are shown in Fig. 6(a). The creation of the unified landmines signatures database is being done which will allow the further research of features extraction and classification tasks.

References

1. Y. Baudoin. Humanitarian demining: Sensor systems, mechanical and robotic systems. In *Int. Conf. Climbing and Walking Robots (CLAWAR),* 2003.
2. JRC. Joint multi-sensor mine signature database. http://demining.jrc.it/msms/, 2004. Join Research Centre – Ispra.
3. S. Larionova, L. Marques, and A.T. de Almeida. Feature-level sensor fusion for a demining robot, In *IARP Int. Workshop Robotics and Mechanical Assistance in Humanitarian Demining and Similar Risky Interventions,* 2004.

4. S. Larionova, L. Marques, and A.T. de Almeida. Toward practical implementation of sensor fusion for a demining robot. In *Int. Conf. on Intelligent Robots and Systems (IROS),* 2004.
5. L. Marques, M. Rachkov, and A.T. de Almeida. Mobile pneumatic robot for demining. In *IEEE Int. Conf. On Robotics and Automation (ICRA),* pp. 3508–3513, 2002.

Co-Operative Smell-Based Navigation for Mobile Robots

C. Lytridis[1], G.S. Virk[2], and E.E. Kadar[3]

[1] Department of Electrical & Electronic Engineering, University of Portsmouth, Anglesea Building, Anglesea Road, PO1 3DJ, Portsmouth, UK
[2] School of Mechanical Engineering, University of Leeds, LS2 9JT, Leeds, UK
[3] Department of Psychology, University of Portsmouth, King Henry Building, King Henry I Street, Portsmouth, PO1 2DY, UK

Abstract. This paper presents the latest developments from research carried out by the Portsmouth/Leeds group regarding odour source localisation using co-operating mobile robots. Previous studies have demonstrated the capability of individual mobile robots to find an odour source using the biologically inspired navigational strategies (chemotaxis and biased random walking (BRW)). A combined chemo-BRW strategy has also been presented and experimental studies have shown that a multi-robot approach can yield good results by improving the efficiency and robustness of the searches. New experimental results are presented in this paper which show that the use of co-operation offers good potential for improving the overall search performance in the odour source localisation problem.

1 Introduction

The area of odour-based navigation is relatively new with only a handful of research groups actively working in it. The potential applications of odour-based navigation such as location of hazardous chemicals, humanitarian demining, etc, have motivated researchers to develop appropriate odour sensors and navigational algorithms. However, there are still two major problems that need solving before the mobile robots can perform effective odour source localisation in practical applications. Firstly, current odour sensors have limited sensitivity and slow response and recovery characteristics. Secondly, more research is needed on the odour-based navigational strategies which will be able to cope with the turbulences and instabilities of chemical fields in natural settings. The first work carried out in odour-based navigation was by Russell with a trail following robot [1]. Further studies by other research groups are presented in [2–4] with encouraging results, although in most cases the experiments are carried out in plume situations and in controlled experimental conditions. In many cases, it is beneficial to adopt a strategy which involves several robots working cooperatively to carry out

specific tasks. Such an approach offers several advantages such as averaging of temporal measurement errors, tolerance to unpredictable failures in the hardware and the fact that many low-cost simple machines can be used instead of a single complex (and potentially expensive) machine [5]. In the odour source localisation task such approaches have been presented in [6] and [7].

Early simulation studies carried out by the authors have demonstrated the potential of the chemotaxis and biased random walk (BRW) strategies [8]. These strategies have been tested in a variety of diffusion fields such as chemical trails [9] moving targets [10] and chemical plumes [11]. These studies have shown the efficiency of searching based on chemotaxis but has revealed the unreliability of this strategy in noisy fields. In contrast, BRW was found to be robust even under unstable and turbulent field conditions but performances can be improved in stable fields. For this reason a combined chemo-BRW strategy was developed which resulted in improved search performances. Simulations have also shown the potential of a multi-agent approach and a cooperation strategy has been proposed [12]. It was observed that the search efficiency in a plume environment is improved significantly when multiple co-operating agents are used, compared to searches involving a single agent or multiple independently working agents. The simulation results for the single agent strategies and the multi-agent non-cooperative searching scenario were verified with experimental studies using the BIRAW robots and were first presented in [13].

The latest experimental results presented here are extending the non-cooperative searching experiments described in [13] in order to conclude with a strategy to demonstrate the advantages of using cooperation in a multi-robot team for odour source localisation. The experiments are carried out in a plume environment as well as a Gaussian-shaped field where the chemical substance spreads evenly in all directions. The present paper is structured as follows: In Sect. 1 the fundamental search strategies BRW, chemotaxis and the combined method are briefly described and the proposed cooperation strategy is introduced. Section 2 deals with the experimental setup for the multi-robot searching experiments. In Sect. 3 the single robot and non-cooperative multi-robot searching experiments are summarised and are compared with the cooperative trials. Finally, in Sect. 4 the experimental results are discussed and recommendations for future work are made in Sect. 5.

2 Navigational Strategies

The biologically-inspired strategies that are used for the experimental work presented in this paper have been discussed extensively in previous studies [8, 11]. In chemotactic searches the robot turns by a constant angle towards the sensor that detects a stronger field and then moves to the new direction by a constant step length. Chemotaxis produces smooth paths but is unreliable in high levels of noise On the other hand, the BRW strategy is essentially

a uni-sensory strategy that uses Poisson statistics to determine a random turning angle and the step length. Longer step lengths are generated when the gradient at the direction of movement is positive and, over time, this causes the searching agent to drift towards the source. In contrast to the basic BRW algorithm two sensors are used in our implementation for the calculation of the forward gradient. BRW is robust and successful even in high levels of instability and noise. However, the method is much less efficient even in smooth fields since the random turns often produce longer and more meandering paths. The combined chemo-BRW strategy uses comparison of the bilateral sensors to determine the direction of the run as in chemotaxis and uses random step length generation as in BRW. It has been shown that this strategy is robust and almost as efficient as chemotaxis.

The above single robot search strategies are used in the multi-robot searches. To avoid collisions between robots a simple collision avoidance strategy was implemented. In brief, the searching strategies described above generate a turning angle and a step length at the end of their execution. The robot turns according to the turning angle indicated by the chosen strategy and samples the infrared sensors. When any of the infrared sensors detects an obstacle at distance less than 30 cm, then the average readings of the sensors on the left and the right side of the robot are calculated and depending on those averages the direction of the avoidance turn is determined. After the avoidance turn, the sensors are sampled again, and if there are no obstacles the robot moves forward using the step length generated by the navigational strategy. In this way the non-cooperative multi-robot searching experiments were carried out.

Cooperation is applied as an additional layer in the execution of the search strategies. The purpose of the proposed co-operation strategy is to bias a robot to others that detect a higher concentration so that its turns are effectively directed to regions of higher concentration. To achieve this, communication between robots is necessary. The BIRAW robots use a radio link for communication, and the data that are exchanged are the current position of the robot and the field strength at the current position. In this way the robots are able to compare their odour sensor readings with those of other robots and decide how to improve their orientation after the independent execution of the single robot strategies (chemotaxis, BRW or the combined strategy). The amount of directional bias that a robot has towards another depends on the difference of the field strengths measured by the two robots as well as their relative positions. Therefore the robot with the lower reading modifies its heading towards others which detect a stronger field by an amount proportional to the difference of the two field strengths.

3 Experimental Setup

Three BIRAW robots are used in the experiments, which were originally designed by Paul Fisher at the University of Portsmouth. These are of cylindrical shape, with a radius of 15 cm. They are equipped with five infrared sensors used for obstacle avoidance, evenly spaced, covering an angle of 120° in front of the robot. The robot has four Figaro TGS2620 olfactory sensors. An investigation on the performance of the particular odour sensors is presented in [14]. One of the BIRAW robots used in the experiments is shown in Fig. 1.

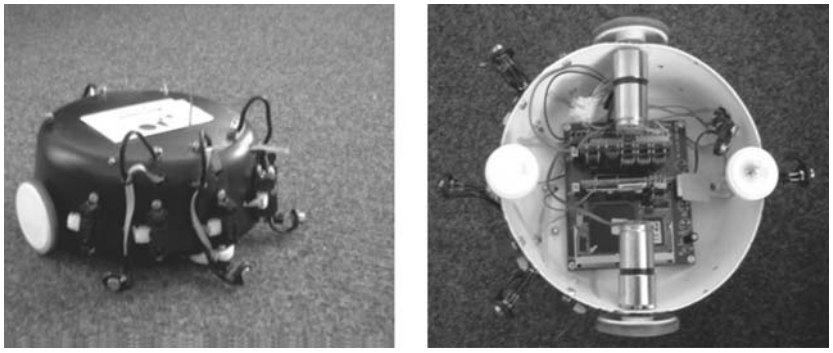


Fig. 1. A BIRAW olfactory mobile robot

The main processing unit of the robots is a Pentium-based DIMM-PC which runs at 133 MHz with a 32 MB solid state hard disk and 32 MB RAM. There are two add-in modules that contain PIC microcontrollers for motor control, RF communications and sensor sampling. The software that runs on the DIMM-PC sends high level commands to the PIC microcontrollers depending on the action the robot is to take (i.e. move, smell, detect obstacles, etc).

The odour source in the experiments is a flat dish with 200 ml of methylated spirits. Because the diffusion process is slow, a fan was used to speed up the spreading of the chemical field. In the case of the Gaussian shaped field the fan was placed on top of the dish and a downward flow was established. For the creation of the plume the fan was positioned such that lateral flow was created. Figure 2 shows the concentration distributions for the two field types, together with the positions of the sensors that were used to map the concentration distribution (indicated by filled circles).

Multi-robot searches were conducted by deploying two and three robots in each case. The initial distance of the robots from the odour source was set at 1.25 metres and the initial orientation was chosen randomly. Figure 3 shows the initial positioning of the robots for (a) two and (b) three agents.

For the measurement of the search performance of each trial two quantities were measured; the total distance travelled from the initial position to the

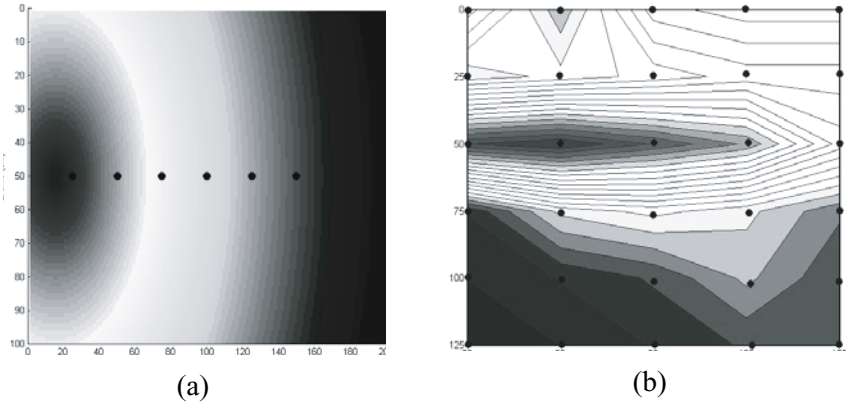


Fig. 2. **(a)** The Gaussian-shaped field and **(b)** the plume

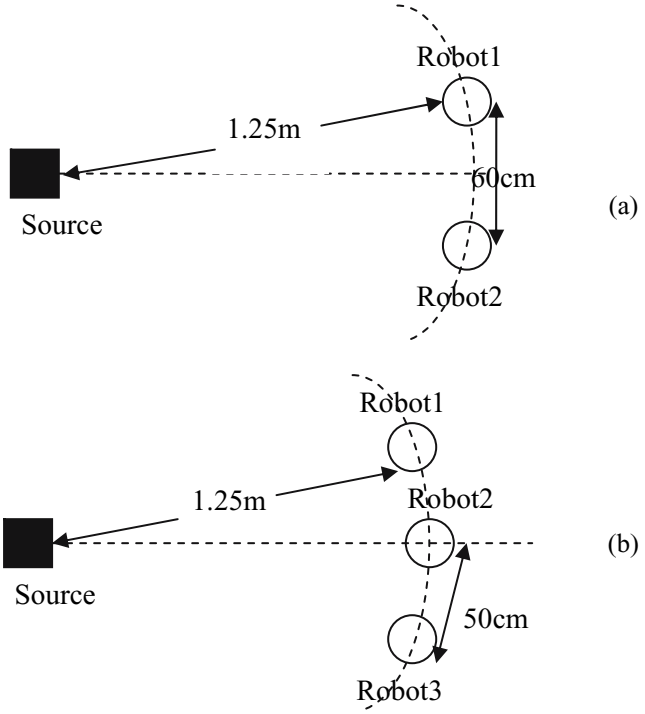


Fig. 3. Positioning of robots for multi-robot searches

source, and the total search duration. The results were averaged over 10 trials for each experimental condition. The performance measured for each trial is based on the total duration and the total distance travelled only for the first robot that reaches the target.

4 Results

Initially, the performance of the multi-robot searches was assessed and it was compared with the performance of the single robot searches. The performances for the three navigational strategies for the different team sizes are given in Fig. 4. The performances of the single robot searches are also shown for comparison.

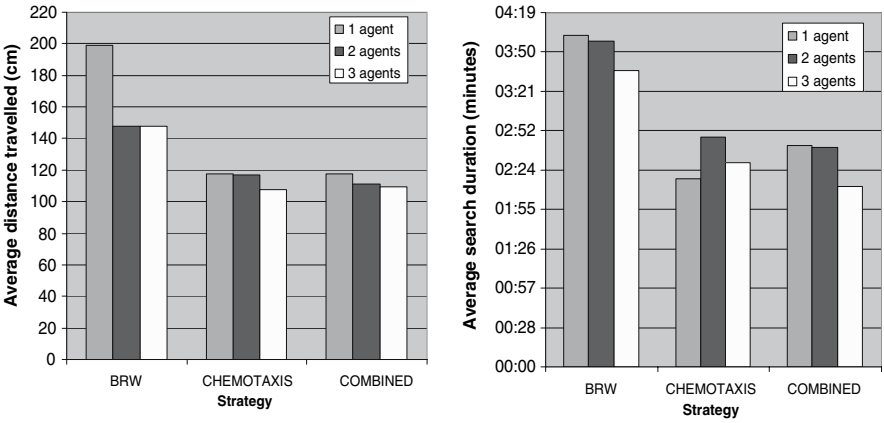
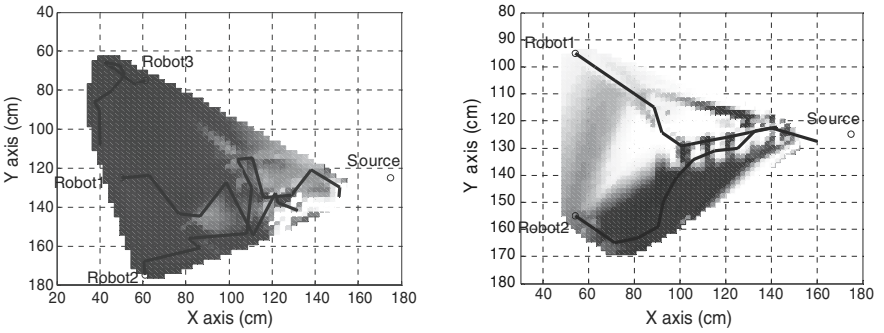


Fig. 4. Comparison between single robot searches and non-cooperative multi-robot searches

Although the results indicate that the use of multiple robots for the localisation of an odour source improves the overall searching performance, the non-cooperative searching experiments have given rise to the issue of how the spatial separation between the robots affects the efficiency of the search. Based on empirical observations during the trials, the avoidance manoeuvres between agents hindered the search and reduced the efficiency. This is more noticeable in the chemotaxis trials where the average search duration does not improve when more robots are used, even though the average distance travelled is reduced. This increase in the search duration occurs due to the clustering of the robots in a small area as the robots move towards higher concentrations and results in the collision avoidance algorithm taking over from the target searching behaviour more frequently. Specifically, the obstacle avoidance causes the robots to turn away from the source for short periods of



(a) BRW search with three agents (b) Chemotaxis search with two agents

Fig. 5. Typical cooperative searches

time in order to avoid collisions with each other. When avoidance is completed, the robots resume the search behaviour.

With the same experimental setup, cooperative searches have been carried out. Figure 5 shows two typical cooperative searches. Note that the robots are converging to the odour source faster.

In general, the cooperation strategy improved the orientations of the robots to lead them to the source more efficiently with smoother paths. In Fig. 6 the performances measured for the non-cooperative and the cooperative trials are compared.

Figure 6 shows a clear improvement of the average distance travelled in all strategies, but the average search duration increased. Moreover, the increase is related to the number of robots. Due to hardware and sensor limitations, the robots are moving sequentially; at the beginning of each step, the robots simultaneously sample the field (after a preset sensor settling delay), and exchange position and field data. The robots then modify the turning angle produced by the navigational strategy and move forward. Since the duration of the motion at each step is variable, to ensure that each robot is ready to receive the appropriate position and sensor data from other robots, motion is arranged in a sequential fashion (Robot 1 first) with an end-of-move signal allowing the next robot in the sequence to start moving.

To account for the way the cooperative trials have been carried out, the average search durations measured are normalised in order to predict this quantity theoretically, assuming that the cooperative searches were carried out asynchronously as were the non-cooperative searches. Normalisation was carried out as follows: For each trial, the duration of motion of each robot was calculated and subtracted from the total duration that was measured. The communication overheads and the delay before the odour sensors' sampling are still included in the normalised duration as the former would still exist in the case of parallel search and the latter is actually performed in parallel.

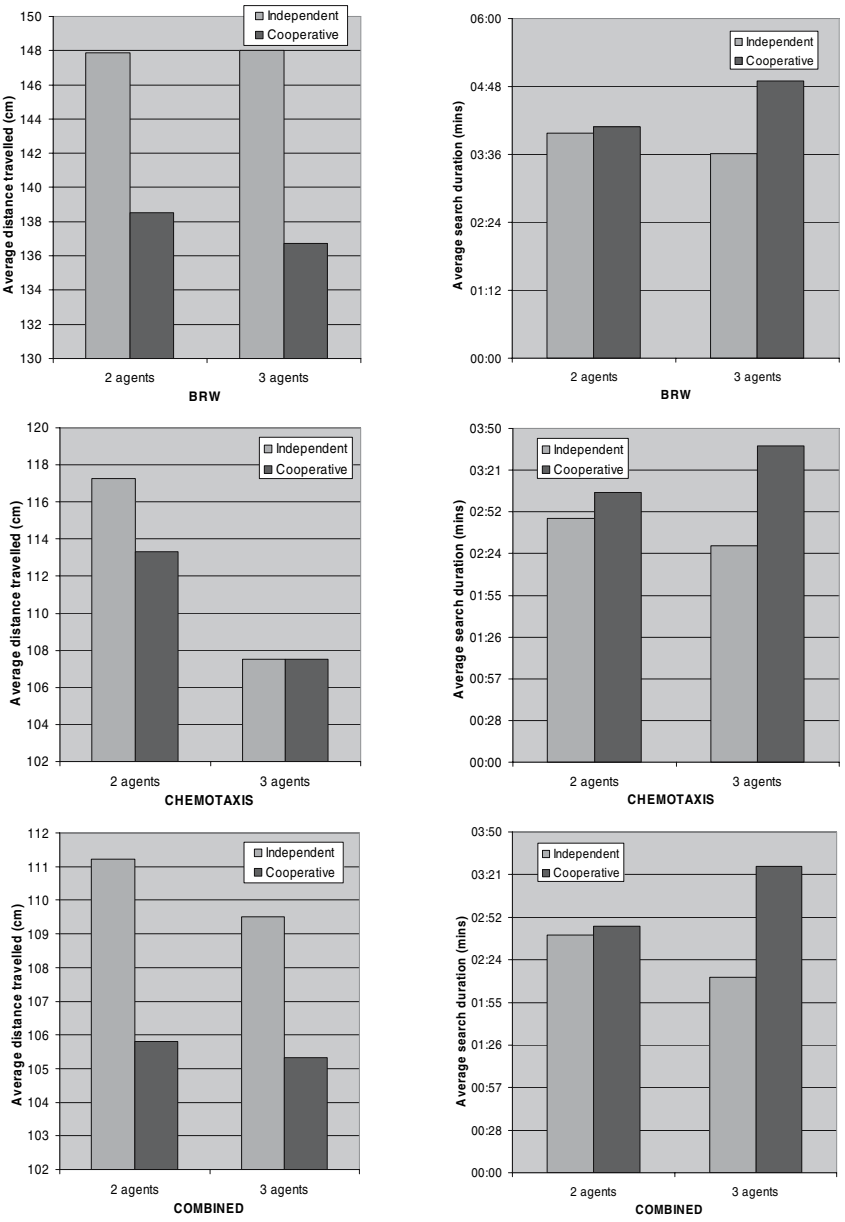


Fig. 6. Comparison of non-cooperative and cooperative search performance

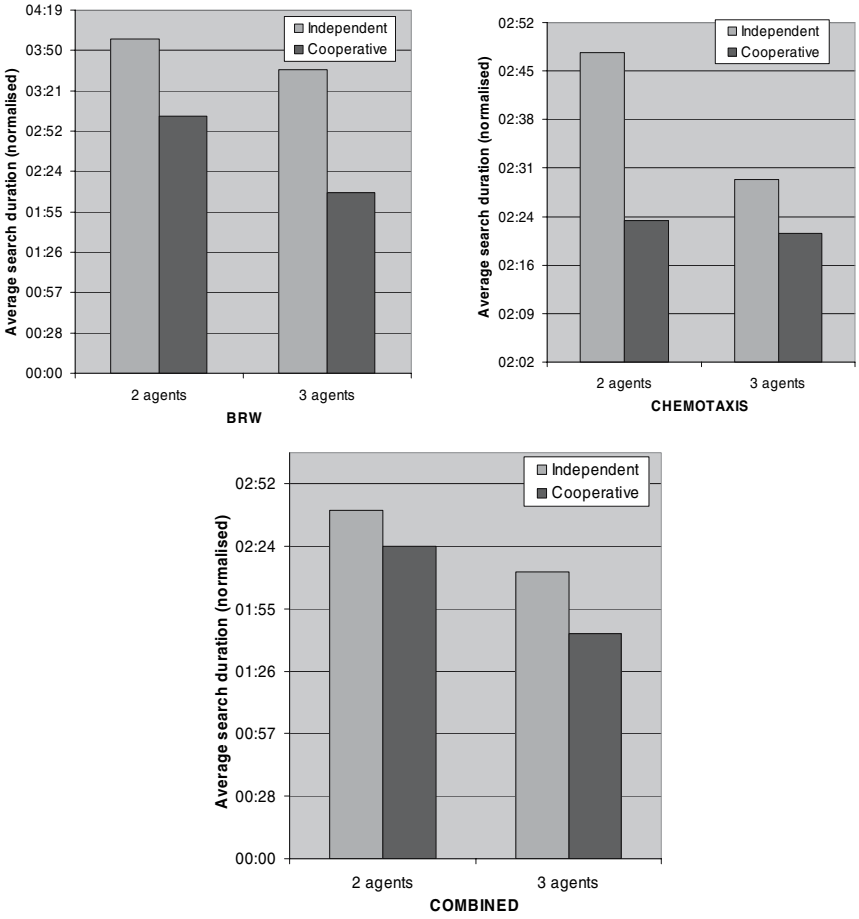


Fig. 7. Duration of non-cooperative trials versus normalised duration of cooperative trials

Fig. 7 compares the search duration measured in the non-cooperative trials and the normalised duration of the cooperative trials.

The comparison between the cooperative normalised search duration and the average search duration measured in non-cooperative search trials shows that the average search duration as well as the average distance travelled in cooperative searches are decreasing for increasing team sizes in a parallel search implementation.

5 Conclusions

In summary, the results have demonstrated the usefulness and the potential of the proposed cooperation method, as a valuable addition to the methods available for solving odour source localisation tasks. The search performance is improved even though the cooperative searches are heavily influenced by the frequent activation of the obstacle avoidance behaviour. The main limitation of the experimental trials was the odour sensors. The sensors' slow response had a significant impact on the overall search performances. Also, due to the limited detection range of the odour sensors, the multi-robot experiments described in this paper were constrained to small searching environments, at the cost of frequent collision avoidance manoeuvres taking place, mostly during the latter stages of each trial. The benefits of the multi-robot approach and the cooperation strategy would be more pronounced in large-scale search tasks, where collision avoidance manoeuvres would be rare. Ultimately, future research will focus on the development of effective search strategies for robots that are designed to carry out odour source localisation in natural outdoor environments.

References

1. Russell RA (1995) Laying and sensing odour markings as a strategy for assisting mobile robot navigation tasks. IEEE Robotics & Automation Magazine, 2(3), 3–9
2. Ishida H, Hayashi K, Takakusaki M, Nakamoto T, Moriizumi T, Kanzaki R (1995) Odour-source localization system mimicking behaviour of silkworm moth. Sensors and Actuators A-Physical, 51(2–3), 225–230
3. Duckett T, Axelsson M, Saffiotti A (2001) Learning to locate an odour source with a mobile robot. IEEE International Conference on Robotics and Automation (pp. 4017–4022)
4. Grasso FW, Consi TR, Mountain DC, Atema J (2000) Biomimetic robot lobster performs chemo-orientation in turbulence using a pair of spatially separated sensors: Progress and challenges. Robotics and Autonomous Systems, 30(1–2), 115–131
5. Cao YU, Fukunaga AS, Kahng AB (1997) Cooperative mobile robotics: antecedents and directions. Autonomous Robots, 4(1), 7–27
6. Sandini G, Lucarini G, Varoli M (1993) Gradient driven self-organizing systems. IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1–3 (pp. 429–432)
7. Hayes AT, Martinoli A, Goodman RM (2003) Swarm robotic odour Localization: Off-line optimization and validation with real robots. Robotica, 21(4), 427–441
8. Kadar EE, Virk GS (1998) Field theory based navigation for autonomous mobile machines. A. Ollero, (Ed.), Proceedings of the IFAC Workshop on Intelligent Components for Vehicles (ICV '98) Amsterdam, The Netherlands
9. Virk GS, Kadar EE (2000) Trail following navigational strategies. Proceedings of the 3rd International Conference on Climbing and Walking Robots (CLAWAR 2000), Madrid, Spain, 605–613

10. Kadar EE, Virk GS (1998) Field theory based navigation towards a moving target. Advanced Robotics: Beyond 2000: 29th International Symposium on Robotics
11. Lytridis C, Virk GS, Rebour Y, Kadar EE (2002) Odor-based navigational strategies for mobile agents. Adaptive Behaviour, 9(3–4), 171–187
12. Virk GS, Lytridis C, Kadar EE, Fisher P (2001) Cooperative target searching in a diffusion field. European Control Conference (ECC 2001), Porto, Portugal
13. Lytridis C, Fisher PVGS, Kadar EE (2003) Odour source localization using co-operating mobile robots. Proceedings of the 6th International Conference on Climbing and Walking Robots (CLAWAR 2003), Catania, Italy, 967–974
14. Lytridis C, Virk GS, Kadar EE, Fisher P (2002) Gas sensors for mobile robot navigation. In: Proceedings of the 5th International Conference on Climbing and Walking Robots (CLAWAR 2002), Paris, France, 233–239

A Localization Algorithm for Outdoor Trajectory Tracking with Legged Robots

J.A. Cobano, J. Estremera, and P. Gonzalez de Santos

Industrial Automation Institute (CSIC), Madrid, Spain

Abstract. This paper describes a method developed to estimate the position of legged robots in outdoors environments. This method combines a dead-reckoning estimation with data provided by a differential global positioning system through an extended Kalman filter algorithm. This localization system is expected to permit accurate trajectory tracking with the SILO6 hexapod robot in humanitarian demining tasks. Preliminary experiments performed with the SILO4 quadruped have shown an adequate performance of this localization system.

1 Introduction

Natural outdoor environment is the typical scenario for the operation of legged robots, as in this situation they are theoretically superior to conventional vehicles. Some suitable applications for legged robots in outdoor environment are the inspection of contaminated areas, volcano exploration, and humanitarian demining. Whether these tasks are accomplished by autonomous or teleoperated robots, precise localization of the robot is required to complete an efficient exploration. Particularly, in humanitarian demining tasks an accurate estimation of the robot position is needed in order to locate mines and to track trajectories that guarantee that the entire minefield is explored.

The work presented in this paper is framed in the DYLEMA project oriented towards the development and experimentation of localization, control and sensor techniques allowing the efficient application of robots in humanitarian demining. The overall DYLEMA demining system [1] is composed of a hexapod legged robot in charge of transporting the different subsystems across the minefield, a sensor head equipped with the landmine detector, a locator system based on a differential global positioning system (DGPS), and a remote operation station (see Fig. 1(a). When exploring a minefield the navigator module generates on–line a complete coverage trajectory despite of obstacles [2] (see Fig. 1(b). In order to assure that the entire minefield is explored the trajectory must be tracked accurately across an unstructured

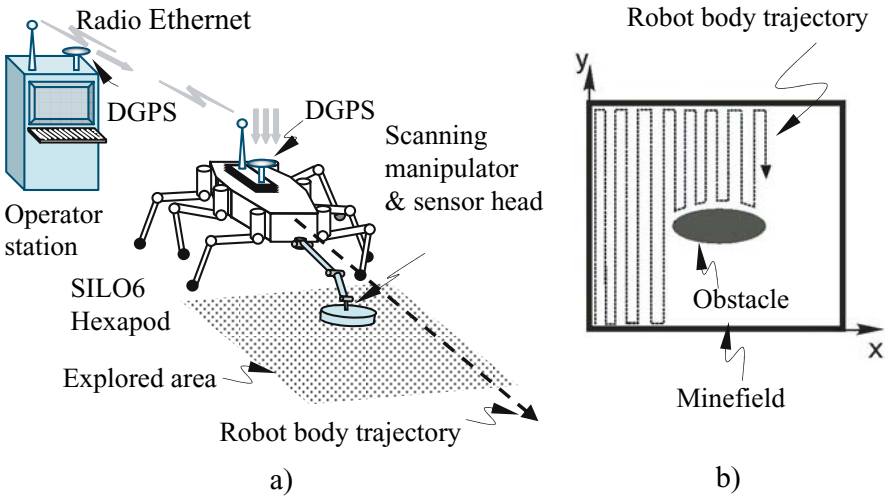


Fig. 1. (**a**) DYLEMA demining system (**b**) Robot trajectory for complete coverage of the minefield

terrain. Additionally, when a potential landmine is detected, it must be precisely located in a database for posterior deactivation.

Regarding these requirements, a DGPS has been selected as a suitable method to locate the robot. Nevertheless dead-reckoning is required when DGPS readings are faulty, inaccurate or inexistent, and a sensor fusion technique is thus needed. Extended Kalman filter based sensor fusion is a technique that has been widely tested for localization of conventional wheeled and tracked robots [3, 4]. However, the application of this method to legged robots presents some particularities which are studied in this work. This paper is focused in the description of the hardware (Sect. 2) and the Extended Kalman filter algorithm (Sect. 3) that compose the localization system SILO6 legged robot (see Fig. 2), as well as the preliminary experiments conducted with the SILO4 legged robot (Sect. 4).

2 System Description

Walking Robot: The SILO6 hexapod and the SILO4 quadruped are medium sized robots (50 and 30 Kg respectively) with similar characteristics (see Fig. 2). They have 3 DOF insect type legs driven by DC motors. Adaptive crab gaits are employed to achieve terrain adaptability. The crab angle is varied in order to track the rectilinear trajectory computed by the navigator; this correction is based on the estimated position of the robot computed by the localization algorithm. A rotation along the vertical axis is superimposed to the gait to assure that the longitudinal axis of the robot is parallel to the

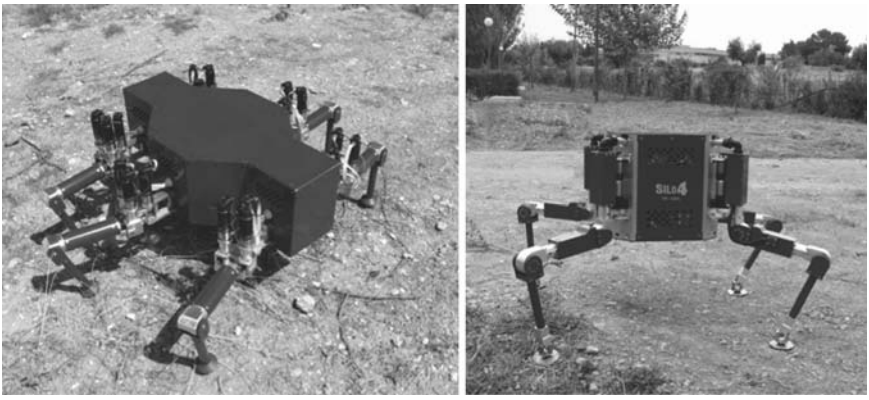


Fig. 2. SILO6 (*left*) and SILO4 (*right*) walking robots

trajectory despite the foot slippage, soil deformation, etc; this correction is based on compass data. A dead-reckoning estimation of the body position is computed every 0.2 seconds (i.e., every 2 mm) based on the average position increment of the supporting feet.

Compass: A magnetic compass model 1525 Analog Sensor is employed to measure the absolute orientation of the robot about its vertical axis. Its precision is about 1°; however, this performance is severely affected by the robot electro-mechanical equipment and metallic mobile masses, as well as external disturbances. Leg transferences cause strong electromagnetic interferences, as well as mechanic shocks and vibrations that affect the measurements, so compass readings are performed only during all-legs support periods. Additionally, the compass must be kept vertically to measure properly the orientation, so the robot walks with its body leveled. **Differential Global Positioning System:** This system is used to determine the absolute position of the SILO4 mobile robot. It is composed of a Trimble 5700 GPS Receptor and Zephyr antenna mounted on board. The DGPS can resolve the position with an accuracy of ±20 mm. However, the accuracy depends on the quality of the differential corrections received from the operator station. In some situations the error of the DGPS may increase up to a few meters when obstacles are present, when multipath effect appears or when the visibility of some satellites is obstructed.

3 Extended Kalman Filter Based Localization Algorithm

An extended Kalman filter (EKF) [5] is employed to determine the position of the legged robot. The derivative of the state transition matrix is employed to linearize the system. The localization algorithm is a classical EKF problem in which the estimation phase is performed using odometry and compass measurements (dead-reckoning) and the update phase employs the DGPS

data. If dead-reckoning estimation differs from DGPS data more than a fixed threshold (50 mm) the DGPS data are discarded.

1. Estimation Phase: The horizontal coordinates of the position of the robot in a fixed reference frame (represented by X_1 and X_2) are estimated from dead-reckoning. The orientation of the robot (represented by angle X_3) is obtained filtering compass readings (α). The estimated position and orientation are then expressed as follows:

$$X(k+1,\,k) = \begin{pmatrix} X_1(k+1,k) \\ X_2(k+1,k) \\ X_3(k+1,k) \end{pmatrix} = \begin{pmatrix} X_1(k) + \Delta X_1(k+1) \\ X_2(k) + \Delta X_2(k+1) \\ (1-\lambda)\alpha(k+1) + \lambda X_3(k) \end{pmatrix} \quad (1)$$

In this expression the parameter λ has been fixed to 0.9 empirically to obtain a suitable filtered signal. ΔX_1, ΔX_2 are the position increments in the fixed frame computed from odometry and compass data. The estimated covariance matrix of the position and orientation error (P) is given by:

$$P(k+1,\,k) = J(k+1,k) \times P(k) \times J^T(k+1,k) + C_o(k+1) \quad (2)$$

where J is the jacobian matrix of the system, employed to linearize the system. The jacobian matrix is computed as:

$$J(k+1,\,k) = \frac{\partial \Phi(k+1,k)}{\partial t} = \begin{pmatrix} 1 & 0 & -T \cdot v(k) \cdot \sin(X_3(k+1,k)) \\ 0 & 1 & T \cdot v(k) \cdot \cos(X_3(k+1,k)) \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

where $\Phi(k, k+1)$ is the transition matrix from the state k to the state $k+1$, $v(k)$ is the speed of the robot and T is the sample time (one second). In (2) C_0 represents the dead-reckoning error, obtained empirically considering the standard deviation measured for each coordinate X_1 and X_2 (Sect. 4) and the compass precision. The correlation error between the different coordinates is not considered, and thus the matrix $C_\circ$ is expressed as follows:

$$C_o = \begin{pmatrix} 0.005 & 0 & 0 \\ 0 & 0.017 & 0 \\ 0 & 0 & 0.01745 \end{pmatrix} \quad (4)$$

2. Update Phase: In this phase the position $Z(k+1)$ given by the coordinates Z_1, Z_2 of the robot is measured with the DGPS and also the estimated measurement vector $Z(k+1,k)$ is calculated as:

$$Z(k+1,\,k) = H \times X(k+1,k) \quad (5)$$

where H is the following matrix:

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (6)$$

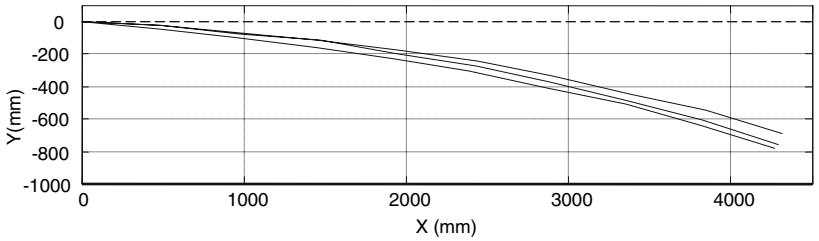


Fig. 3. Desired trajectory (*dashed line*) and trajectories described by the robot in open loop control (*solid lines*) in several experiments

The Kalman matrix (K) is the gain or blending factor that minimizes the *a posteriori* error covariance:

$$K(k+1) = P(k+1,k) \times H^T \times \left[H \times P(k+1,k) \times H^T + C_g(k+1)\right]^{-1} \quad (7)$$

In this expression C_{g} is the matrix representing the error introduced by the DGPS measurement:

$$C_g = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \quad (8)$$

where σ_1 and σ_2 are the standard deviation of each coordinate, which depend on the DGPS measurement quality indicator.

Finally the robot position is computed and the error covariance matrix is updated with the Kalman matrix:

$$X(k+1) = X(k+1,\, k) + K(k+1) \times [Z(k+1,k) - Z(k+1)] \quad (9)$$

$$P(k+1) = [I - K(k+1) \times H] \times P(k+1,k) \quad (10)$$

where I is the identity matrix.

4 Experiments

The above described localization system has been tested in the SILO4 quadruped robot. In the experiments performed the robot must follow a rectilinear trajectory (defined by $y = 0$ in the fixed reference frame) in different conditions. Initially the robot's longitudinal axis is aligned with the trajectory. The body speed was fixed to 10 mm/s and the crab angle to 0°.

In the first experiment, the robot was intended to walk along the desired trajectory in open loop control in order to calculate the odometry errors in expression (4). The robot trajectories measured in three different experiments show a strong tendency of the robot to turn right (see Fig. 3).

In the second experiment, the robot walks again in open loop control, while the EKF is employed to estimate the position of the robot. Figure 4(a) shows

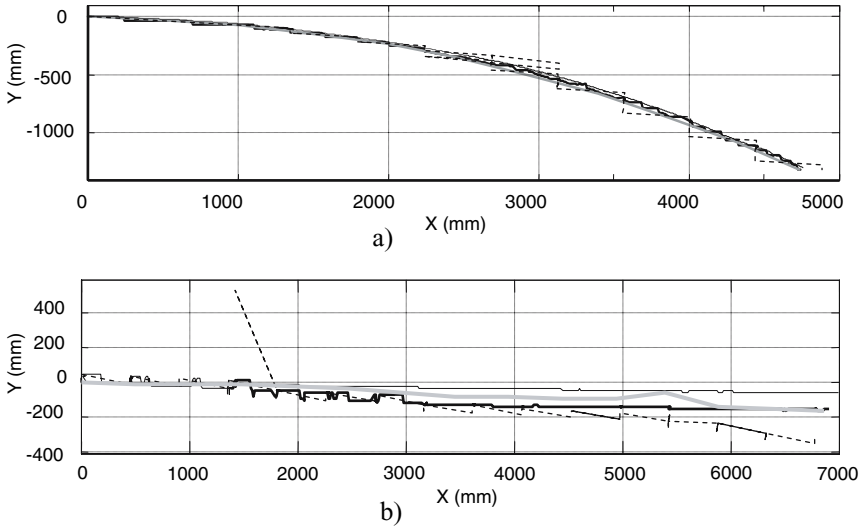


Fig. 4. Results obtained with open loop control (**a**) and with orientation control (**b**): Trajectory followed by the robot (*solid gray line*), dead-reckoning estimation (*solid black thin line*), DGPS data (*dashed line*) and EKF estimation (*solid black thick line*)

the real trajectory followed by the robot compared with the dead-reckoning estimation, the DGPS data and the EKF estimation.

In the third experiment, the robot employs the compass data to control its orientation in order to keep it constant. Figure 4(b) shows the results obtained in this case. Figure 5 depicts the compass raw and filtered data obtained in this experiment.

5 Discussion

One of the main characteristic of walking robots in general, and the SILO4 in particular, is their low speed. This makes impractical the method described in [6] to compute the robot orientation from two consecutive DGPS measurements, since usually DGPS data remain constant during several readings (see the "steps" in the DGPS signal in Fig. 4(a)). This also causes the DGPS data to differ excessively from dead-reckoning estimation in a periodic way (DGPS errors of about 100 mm can be seen periodically in Fig. 4(a)). For this reason DGPS data are considered only if they differ less than a fixed threshold from the dead-reckoning estimation (see Sect. 3). Additionally, given that speed is low, a large number of compass readings can be made and strongly low-pass filtered to obtain utilizable orientation data despite of high noise in this sensor (see Fig. 5). Compass data are employed jointly with odometry data to obtain an accurate dead-reckoning estimation of the position of

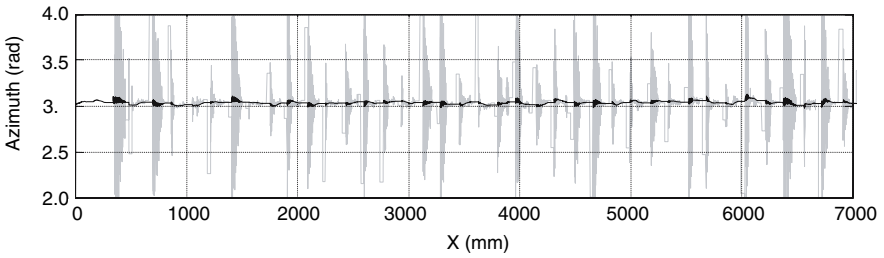


Fig. 5. Raw compass data (*gray line*) and filtered data employed for orientation control (*black line*)

the robot (Fig. 4(a) shows a precision of about 26 mm when the robot has covered 5 m). Compass data are also used to achieve an effective body orientation control, which is fast enough to correct minor azimuth deviations (about 0.1 radians, see Fig. 5) and to considerably reduce the tracking errors (see Fig. 4(b) despite of the lack of a true trajectory tracking control at this stage.

The dead-reckoning estimation helps to achieve a smooth evolution of the EKF position if compared with the DGPS data. This will facilitate the control of the crab angle to achieve true trajectory tracking in the future. However, localization errors due to a misalignment between the compass and DGPS reference frames and due to compass errors have been observed (see Fig. 4(b). Considering all the available sensor sources jointly, the position of the robot was estimated with an error of less than 50 mm when the robot had covered 7 m. Nevertheless, the localization system must be tested in longer experiments in which the dead-reckoning estimation becomes inaccurate enough to test the effectiveness of the correction introduced by the DGPS.

6 Conclusions

A localization system for legged robots has been described in this paper. Both the hardware and algorithms have proven experimentally the efficacy of the system to locate the robot and to make possible the control of its motion. Current work is oriented to control the crab angle of the walking machine in order to accurately track the desired trajectory. New strategies are also needed to estimate the position of the robot when DGPS data quality is lower, in order to achieve a realistic localization system.

Acknowledgement

This work was supported by CICYT under Grant DPI2001–1595.

References

1. Gonzalez de Santos, P., Garcia, E., Estremera J. and Armada, M.A., "SILO6: Design and configuration of a legged robot for humanitarian demining", *IARP Workshop on Robots for Humanitarian Demining*, Vienna, Austria, 2002
2. Garcia, E. and Gonzalez de Santos, P., "Mobile Robot Navigation with Complete Coverage of Unstructured Environments", Robotics and Autonomous Systems, vol. 46, no. 4, pp. 195–204, 2004.
3. Goel, P., Roumeliotis, S.I. and Sukhatme, G.S., "Robust Localization Using Relative and Absolute Position Estimates", In Proc. 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Kyongju, Korea, 1999, pp. 1134–1140.
4. Roumeliotis, S.I., Bekey, G.A., "An extended Kalman filter for frequent local and infrequent global sensor data fusion", In Proc. Of the SPIE (Sensor Fusion and Decentralized Control in Autonomous Robotic Sytems), Pittsburgh, Pennsylvania, USA, 1997, pp. 11–22.
5. Gelb, A., Kasper, J. F: Jr., Nash, R. A. Jr., Price, C. F., Sutherland A. A. Jr., 1986, "Applied Optimal Estimation", The M.I.T Press.
6. Caltabiano, D. and Muscato, G., "An new localization algorithm for mobile robots in outdoor environments", In Proc. Of the 6th International Conference on Climbing and Walking Robots, Catania, Italy, 2003, pp. 925–932.

Sit to Stand Transfer Assisting by an Intelligent Walking-Aid

P. Médèric, V. Pasqui, F. Plumet, and P. Bidaud

Laboratoire de Robotique de Paris (LRP), CNRS FRE 2507 – Universit'e Pierre et Marie Curie, Paris 6, 18, route du Panorama, BP 61, 92265 Fontenay-aux-Roses, France
{mederic,pasqui,plumet,bidaud}@robot.jussieu.fr

Abstract. The sit to transfer assisting for elderly patients using an intelligent walking-aid is presented. This assistive device provides physical support to aid elderly with physical impairment during both the walk and the sit-to-stand transfer. In this paper, we will focus on the sit-to-stand transfer function and more precisely on the trajectory generation of the assistive device handles using interpolating cubic splines.

1 Introduction

Instability and falling are among the most serious problems associated with aging. Age-related changes in the neural, sensory and musculoskeletal systems can lead to balance impairments that have a tremendous impact on the ability to move safely, and the consequences of instability and falling, in terms of health care costs and quality of life, are more than significant.

A growing interest in developing intelligent assistive devices for the elderly can be noted in the past few years. Robotics technologies have been investigated to provide physical support of patients and to promote safe mobility [1–4]. The Care-O-Bot [5, 6] and the Nursebot [7] are personal service robots projects mainly focused on the man-machine interaction. A power-assistance device has been developed by the Ritsumeikan University [8] to provide physical support during the walk and the sit to stand transfer. This system is fixed and must be installed in a given hospital room.

The paper is organized as follow: we first present the assistive device developed in our laboratory. This assistive device provide support during the walk and also during the sit to stand transfer and is primarly intended for elderly patients. Then, the synthesis of the handles trajectory for the sit to stand and stand to sit transfer is presented. This synthesis is done using temporal cubic splines and is based on an analysis of experimental sit to stand transfer curves recorded on a set of elderly patients.

2 Assistive Device

Functional decline due to aging and fall, with its functional and psychological consequences, are responsible for the most common walking troubles associated with aging. Injuries with bone fractures and fear of fall ("post-fall" syndrome) are the main pathologies appearing after a fall. After a bone fracture on the lower limb, the medical staff must thus spent a lot of time and energy helping patients to stand up and to walk by their own self with intensive rehabilitation exercise. The "post-fall" syndrome is one of the psychological consequences of the fall. This syndrome leads to a disturbance of posture: the retropulsion. During the walk, the torso is lined backward with shuffling gait and the risk of fall increases. The elderly must also be assisted in the sit to stand transfer since patient sited-down in a retropulsion configuration cannot use properly his body to get into an antepulsion posture to provide propulsion in the direction of the motion.

Active devices for postural compensation can then free medical staff for other tasks, and help elderly people to do rehabilitation exercises with various difficulty levels. Based on this analysis of the most common walking trouble associated with aging, we design the overall kinematic of our assistive device. This assistive device has two main functions:

- Postural stability, to provide support and avoid fall during the walk.
- Verticalisation, to help elderly during stand-up and sit-down transitions.

The designed robotic system is basically a two degrees of freedom mechanism mounted on an active mobile plateform (see Fig. 1).

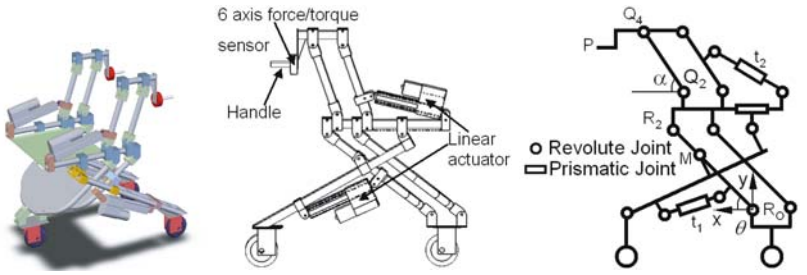


Fig. 1. Description of the assisting device

For the sit to stand transfer, the handles must first pull slowly the patient to an antepulsion configuration. Then, the handles go from this down position to the up position, which is the position used for walking. Obviously, the handles must remain horizontal during the whole transition. This is obtained using two parallel and independent mechanisms combined in a serial way. Details on the design of this assistive device can be found in [9].

To assist the user during the transfer the handles of the assisting device must be controlled, this is made by using the inverse kinematic model of our structure. The inverse kinematic model define the position of the tip of the handles $\mathbf{x_P}$ in the frame R_0 for a given configuration of the linear actuator (t_1 and t_2). We get this model by considering the device as a planar 2R (see Fig. 1) with $\phi = \theta - \alpha$. The position of the point P is given by:

$$\mathbf{x_P} = \begin{pmatrix} X_P \\ Y_P \end{pmatrix} = \begin{pmatrix} OR_2 \cos\theta - R_2Q_{2x} + Q_2Q_4 \cos\phi + Q_4P_x \\ OR_2 \sin\theta + R_2Q_{2y} + Q_2Q_4 \sin\phi - Q_4P_y \end{pmatrix} \tag{1}$$

We first get the value of ϕ for a given position $\mathbf{x_P}$ of the handle, in (1) we group the constant terms in a vector $\mathbf{x'_P}$:

$$\mathbf{x'_P} = \begin{pmatrix} X'_P \\ Y'_P \end{pmatrix} = \begin{pmatrix} OR_2 \cos\theta + Q_2Q_4 \cos\phi \\ OR_2 \sin\theta + Q_2Q_4 \sin\phi \end{pmatrix} \tag{2}$$

With $\phi \in \left[0, \frac{\pi}{2}\right]$ the relation between ϕ and $\mathbf{x'_P}$ is given by:

$$\phi = \arccos\left(\frac{(X'_P)^2 + (Y'_P)^2 - \left(OR_2^2 + Q_2Q_4^2\right)}{2OR_2Q_2Q_4}\right) \tag{3}$$

With ϕ known we have the following system:

$$\mathbf{x'_P} = \mathbf{A}\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \tag{4}$$

The matrix $\mathbf{A}$ is given by:

$$\mathbf{A} = \begin{pmatrix} OR_2 + Q_2Q_4 \cos\phi & Q_2Q_4 \sin\phi \\ Q_2Q_4 \sin\phi & OR_2 + Q_2Q_4 \cos\phi \end{pmatrix} \tag{5}$$

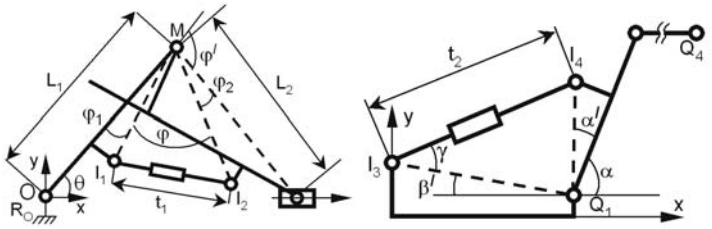
After solving (3) and (4) we need the relation between the displacement of the linear actuator t_1 and the angular variation θ and also the relation between t_2 and α. For the lower part (see Fig. 2(a)) we get the following expression:

$$t_1 = \sqrt{MI_1^2 + MI_2^2 - 2MI_1MI_2 \cos\left(-\pi + \varphi_1 + \varphi_2 + \arcsin\left(\frac{L_1}{L_2}\sin\theta\right)\right)} \tag{6}$$

For the upper part (see Fig. 2(b)) the relation with the angle α and the linear displacement t_2 is given by :

$$t_2 = \sqrt{Q_1I_3^2 + (Q_1I_4^2 + 2Q_1I_3Q_1I_4 \cos(\alpha + \beta'))\cos(\alpha')} \tag{7}$$

In the next section, we describe the handle's trajectory synthesis needed to perform the transfer assistance.



(a) Lower part (b) Upper part

Fig. 2. Kinematic descriptions of the *Lower* and *Upper* part

3 Trajectory Synthesis

In order to implement the trajectory generation (described in Sect. 3.2) for the handles of our prototype, we first do some experiments with patients and record the patient's hands trajectories and forces during assisted sit to stand and stand to sit transfers.

3.1 Experimental Trajectory Analysis

Experiments has been conducted on a set of 7 elderly patients in Charles-Foix Hospital under the supervision of gerontologists and physiotherapists.

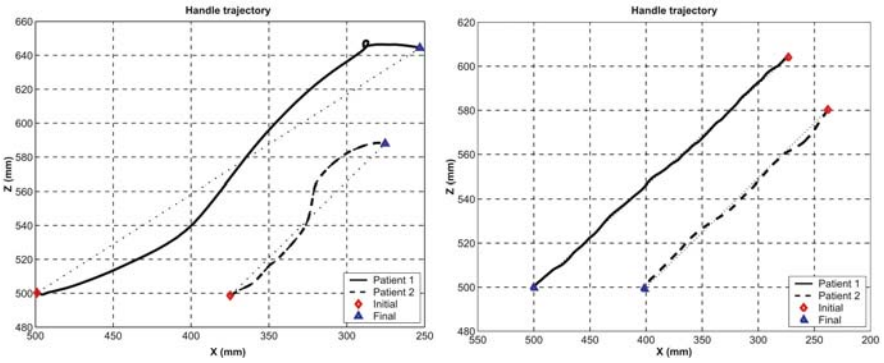
A test platform has been developed to simultaneously record the trajectory and force interaction of elderly's hand during sit-to-stand and stand-to-sit transfer [9]. For the sit to stand transfer, the patient is seated on a chair and holds the central handle. The two caregivers hold their handles and impose the motion of the test platform according to the gerontologists and physiotherapists instructions until the patient is in a quiet standing position. The initial (down) position of the patient handle is chosen close to the knee position of the patient, forcing the patient to leave off his rather usual retropulsion configuration for an antepulsion configuration.

The reverse procedure is applied for the stand to sit transfer.

For each patient, several sit to stand and stand to sit transfer trajectories was recorded and some examples of these trajectories are given Fig. 3.

Analysis of the transfer trajectories records show some interesting points:

- The sit to stand transfer trajectory follow roughly a straight line between the two end points with a normal deviation that never exceed 12% of the total trajectory length for all the patients. The sit to stand transfer trajectory is similar but with a normal deviation that do exceed 5% of the total length.
- The global shape of the trajectory is not directly related to the age, height or pathology of the patient. Moreover, the difference between the initial and final point (down and up positions of the handle) is not directly related to the height of each patient but seems to be correlated with its own personal strategy to stand up or sit down.



Sit to Stand transfer Stand to Sit transfer

Fig. 3. Trajectories for different Patient

These items can't be considered as rules (since the number of patients is not statistically significant and since the observed trajectories are the result of an interaction between the caregivers and the patient) but rather like practical guidelines for the trajectory generation described in the next section.

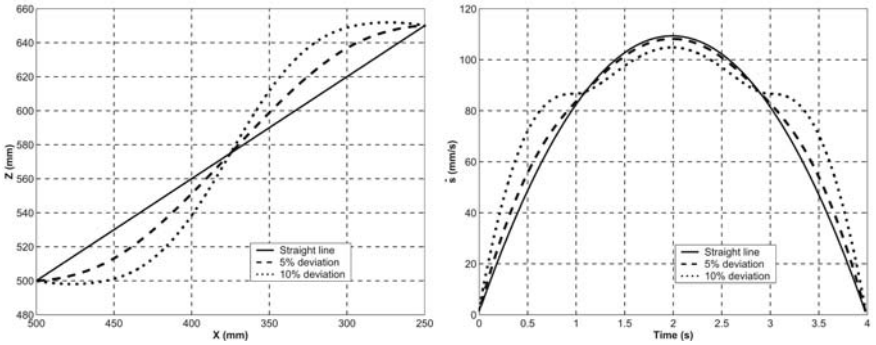
3.2 Trajectory Generation

A trajectory generation is needed to compute the desired motion of the handles of the assistive device in real time. These trajectories have no need to precisely interpolate the experimental trajectories but they rather must reproduce the global shape of the trajectories depicted in Sect. 3.1.

The trajectory generation is based on interpolating cubic splines for their interesting properties which are mainly the C^2 continuity, the local deformation properties and the computational efficiency of this method (given a set of control points, interpolating cubic spline is a piecewise 3rd order polynomial curve that interpolates these control points i.e. passes through them all).

From a functional point of view, the patient first choose the initial $\mathbf{X}_{Ini}$ and final $\mathbf{X}_{End}$ points of the transfer trajectory (up and down positions of the handles in the X-Z plane). The trajectory generation then automatically add 3 other interpolating points between $\mathbf{X}_{Ini}$ and $\mathbf{X}_{End}$ to control the local deformation of the curve and compute the coefficients of the 3rd order piecewise polynomial (details on the computation of these coefficients are given in the Appendix). According to the experiments done in Sect. 3.1, three points is enough to control the global shape of the transfer trajectory.

Finally, and since we use a temporal spline (i.e the $\mathbf{u}$ parameter is a linear function of time: $\mathbf{u} = \alpha_i \, \mathrm{t}$ with $\alpha_i = 1/\mathrm{T}_i$ and T_i the time to go from the control point $\mathbf{X}_i$ to the control point $\mathbf{X}_{i+1}$), the real time computation of the



Spline trajectories (X-Y Plane) Curvilinear speed

Fig. 4. Interpolating trajectories exemples

trajectory begin i.e. a new point can be computed at each sampling time using a increment $\mathrm{du} = \mathrm{T}_e/\mathrm{T}_i$ of the parameter $\mathbf{u}$.

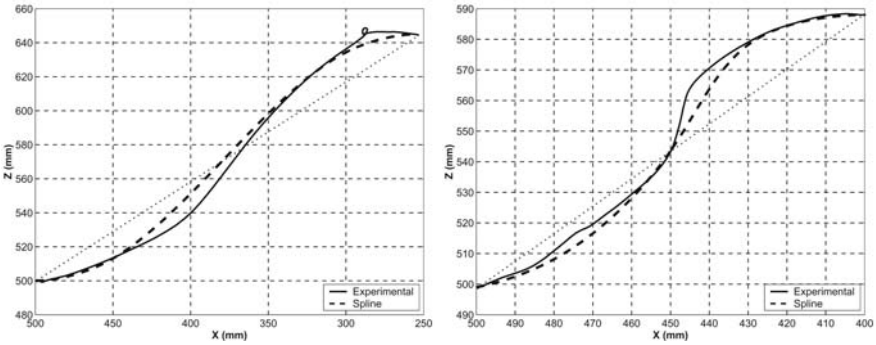
However, as mentioned early, the cubic spline is continuous up to the 2nd derivative of $\mathbf{u}$. Recalling that $\mathbf{u}$ is a linear function of time, the trajectory is also continuous up to the 2nd derivative of time only if $\mathbf{u}$ is the same linear function of time for the whole spline. An immediate and important consequence is that the time to go from $\mathbf{X}_i$ to $\mathbf{X}_{i+1}$ must be equal i.e each control points must be equally spaced in time. In other words, the choice of the added interpolating points not only control the local deformation of the curve but also the velocity profile along the trajectory.

To help elderly patient in the sit to stand (or stand to sit) transfer, we need to generate smooth trajectory both in position and time i.e. the patient's hands must be accelerated gently to attain a maximum speed and then decelerated to reach the final position. For example, to get a parabolic velocity profile for the entire spline, elementary calculation gives that the 3 additional points $\{X_1, X_2, X_3\}$ must be chosen such that:

$$X_1 - X_{Ini} = \frac{5}{32}L \qquad X_2 - X_{Ini} = \frac{L}{2} \qquad X_{End} - X_3 = \frac{5}{32}L$$

with L the total length of the trajectory. Some examples of the interpolating spline trajectory for 5% and 10% of the total length normal deviation are given Fig. 4.

A comparison between experimental trajectories and the resulting spline trajectories for the sit to stand transfer is given Fig. 5.



Patient 1 Patient 2

Fig. 5. Experimental and interpolating spline trajectories

4 Conclusion

An assistive device for elderly patients has been developed to provide support during the walk and also during the sit to stand transfer. The synthesis of the handles trajectory for the sit to stand and stand to sit transfer has been presented. This synthesis is done using temporal cubic splines and is based on an analysis of experimental sit to stand transfer curves recorded on a set of elderly patients. This results in a automatic and versatile trajectory generator that can be adapted to the personal feeling and strategy to stand up and sit down of the patient.

Since force sensor are integrated into the handles, this device can also be a useful tools for gerontologist to better understand walking troubles and to prevent falls as well as a powerful programmable device for the monitoring and rehabilitation of the lower limbs.

Acknowledgments

This work is partly supported by the French RNTS (Réseau National des Technologies de la Santé) program under grant N° 02B0414 (Monimad Project).

Appendix

Let $\mathbf{X} = \{x_o, x_1, \ldots x_n\}$ be a set of (n+1) points equally spaced in time and S the interpolating cubic spline passing trough each point x_i. On each segment $[x_i\ x_{i+1}]$, the 3rd order spline polynomial is written as:

$$S_i(u) = a_i u^3 + b_i u^2 + c_i u + d_i \qquad i = 0 \ldots (n-1)$$

with u = [0 ... 1] on each segment.

The first derivative S_i' and second derivative S_i'' of S_i with respect to u are given by :

$$S_i'(u) = 3a_i u^2 + 2b_i u + c_i \quad \text{and} \quad S_i''(u) = 6a_i u + 2b_i \qquad i = 0 \ldots (n-1)$$

For (n) segments, we must then compute (4n) unknown coefficients. The interpolation conditions give (n+1) equations and the continuity conditions on S, S′ and S′ give 3(n–1) equations which leads to (4n–2) equations. The two remaining equations are given by the boundary conditions of the S curve i.e the initial velocity V_{Ini} on the first segment and the final velocity V_{End} on the last segment.

- The interpolation condition is: $S_i(u = 0) = x_i$ which leads to:

$$d_i = x_i \qquad i = 0 \ldots (n-1) \tag{8}$$

- The continuity condition on S_i'' is: $S_i''(u = 1) = S_{i+1}''(u = 0)$ which leads to: $6a_i + 2b_i = 2b_{i+1}$ and finally:

$$a_i = \frac{b_{i+1} - b_i}{3} \qquad i = 0 \ldots (n-1) \tag{9}$$

  Note that an unknown additional parameter b_n must be added to get a valid equation for i = n−1.
- The continuity condition on S_i is: $S_i(u = 1) = S_{i+1}(u = 0)$ which leads to: $a_i + b_i + c_i + d_i = d_{i+1}$ and finally:

$$c_i = (x_{i+1} - x_i) - \frac{b_{i+1}}{3} - \frac{2b_i}{3} \qquad i = 0 \ldots (n-1) \tag{10}$$

- The continuity condition on S_i' is: $S_i'(u = 1) = S_{i+1}'(u = 0)$ which leads to: $3a_i + 2b_i + c_i = c_{i+1}$. Replacing a_i and c_i by their values given in (9) and (10), we have:

$$b_i + 4b_{i+1} + 4b_{i+2} = 3\left[(x_{i+2} - x_{i+1}) - (x_{i+1} - x_i)\right] \qquad i = 0 \ldots (n-2) \tag{11}$$

- The two boundary conditions give two other equations.
  $S_0'(u = 0) = V_{Ini} = c_0$. Replacing in (10), this gives:

$$2b_0 + b_1 = 3\left[(x_1 - x_0) - V_{Ini}\right] \tag{12}$$

  And the boundary condition on the last segment:
  $S_{n-1}'(u = 1) = V_{Fin} = 3a_{n-1} + 2b_{n-1} + c_{n-1}$ gives in a similar manner:

$$b_{n-1} + b_n = 3\left[V_{Fin} - (x_n - x_{n-1})\right] \tag{13}$$

Equation (11), (12) and (13) can be put in a matrix form:

$$\begin{bmatrix} 2 & 1 & 0 & 0 & \dots & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & \dots & 0 & 1 & 4 & 1 \\ 0 & \dots & 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} = \begin{bmatrix} 3\left(\Delta x_1 - V_{Ini}\right) \\ 3\left(\Delta x_2 - \Delta x_1\right) \\ 3\left(\Delta x_3 - \Delta x_2\right) \\ \vdots \\ 3\left(\Delta x_n - \Delta x_{n-1}\right) \\ 3\left(V_{Ini} - \Delta x_n\right) \end{bmatrix} \tag{14}$$

with $\Delta x_i = x_i - x_{i-1}$.

Finally, the algorithm to compute the (4n) coefficients of the spline is as follow:

- Compute $d_i (i = 0 \dots n-1)$ using (8)
- Compute $b_i (i = 0 \dots n)$ by inverting (14) with LU decomposition or similar algorithm.
- Compute $a_i (i = 0 \dots n-1)$ using (9)
- Compute $c_i (i = 0 \dots n-1)$ using (10)

References

1. G. Lacey, S. Mac Namara, and K. M. Dawson-Howe. Personal adaptive mobility aid for the infirm and elderly blind. *Lecture Notes in Computer Science*, 1458:211–220, 1998.
2. S. Dubowsky, F. Genot, S. Godding, H. Kozono, A. Skwersky, H. Yu, and L.S. Yu. Pamm – a robotic aid to the elderly for mobility assistance and monitoring: A "helping-hand" for the elderly. In *IEEE Int. Conference on Robotics and Automation*, pp. 570–576, 2000.
3. H. Yu, M. Spenko, and S. Dubowsky. An adaptive shared control system for an intelligent mobility aid for the elderly. *Autonomous Robots*, 15:53–66, 2003.
4. C. Lee, S. Oh K. Kim, and J. Lee. A system for gait rehabilitation: mobile manipulator approach. In *IEEE Int. Conference on Robotics and Automation*, pp. 3254–3259, 2002.
5. R.D. Schraft, C. Schaeffer, and T. May. Care-o-bot™: The concept of a system for assisting elderly or disabled persons in home environments. In *24th Annual Conf. of the IEEE Industrial Electronics Society (IECON'98)*, pp. 2476–2481, 1998.
6. B. Graf and M. Hägele. Dependable interaction with an intelligent home care robot. In *IEEE Int. Conference on Robotics and Automation*, pp. 21–26, 2001.
7. G. Baltus et al. Towards personal service robots for the elderly. In *Workshop on Interactive Robotics and Entertainment (WIRE'00)*, 2000.
8. K. Nagai, I. Nakanishi, and H. Hanafusa. Assistance of self-transfer of patients using a power-assisting device. In *IEEE Int. Conference on Robotics and Automation*, pp. 4008–4015, 2003.
9. P. Médéric, J. Lozada, V. Pasqui, F. Plumet, P. Bidaud, and J. C. Guinot. An optimized design for an intelligent walking-aid. In *6th Int. Conference on Climbing on Walking Robots (CLAWAR'03)*, pp. 53–60, 2003.

Part XI

CLAWAR Network Workpackages

CLAWAR Modularity – Design Tools

G.S. Virk

School of Mechanical Eng, University of Leeds, Leeds, LS2 9JT, U.K.
g.s.virk@leeds.ac.uk

Abstract. The paper presents the latest work on modularity carried out by the EC funded CLAWAR project partners during the second year of the project. This work has focused on specifying the design tools needed to support the overall open modular concepts being proposed to encourage the creation of a component based research and development community for robotics.

1 Introduction

From its start in 1998, CLAWAR has stressed the importance of open modularity and the development of open standards for robot components so that the problems of "reinventing-the-wheel" scenarios could be reduced and we can create and sustain a viable and vibrant robotics research community and robot products industry. CLAWAR has taken a holistic view by including all the stakeholders in the planning phases to produce a generic modular design philosophy that sub-divides the overall robot design process into smaller sections where the individual components (modules) can link up to other modules to form the overall system using an "interaction space highway" type of data bus. This involves determining how the modules need to link up. After considerable investigations and discussions it has been established that six interaction variables are needed for this inter-connectivity, namely, (1) power, (2) computer data bus, (3) mechanical linkages, (4) analogue signals, (5) digital signals and (6) working environment (see Virk [1–6]).

The latest work on modularity has focused on the design tools that could be useful to support the overall component based approach. The design tools need to cover the entire process from starting the requirement formulation phase of the design to delivering the final robot using it and recycling the materials at the end of the robot's useful life. The important aspects need to be included:

(1) Gathering the information in the specifying requirement stage;

(2) Creating a simulation environment and the tasks to be carried out;
(3) Creation of design concepts for the possible solutions;
(4) Virtual prototyping and testing of the designs at modular and system levels;
(5) Assembly of the modules to give sub-systems, super-modules and the full system;
(6) Engineering design and selection of the modules (via the use of existing knowledge of methods and tools;
(7) Physical prototyping and testing at the module, super module and system levels is needed.

The paper presents a summary of the results to specify the design tools needed; the full results are presented in the Year 2 report on Workpackage 2 that has been prepared and submitted to the EC (see Virk [7]).

2 Design Tools

The work on specifying the design tools has involved looking at several different areas, but the philosophy has been to specify the generic tools needed to cover the overall design process mirroring the entire lifecycle phases of particular systems. These include identifying a need, specifying its requirements, specifying the design parameters, creating the design specifications, actually carrying out the design (at module and system levels), developing, building and testing the prototypes, putting the designs into production, commercialising and then recycling.

In order to support this entire process, it has been agreed to produce the following design tools:

- Tools to specify the design requirements. These have been broken down into different environments and application sectors. What tasks need to be carried out; how the designs are assessed; how the designs are operated; what standards to use are aspects that all need to be included in formulating the tools. These tools should allow static and dynamic analysis and simulation within the various environments and the assessment of designs using different metrics. These include cost, specific task needs, speed, reliability, ease of use, etc.
- Tools needed to assist in the creation of design concepts (via expert knowledge)
- Tools for virtual and rapid prototyping
- Testing and analysis tools
- Creation of engineering design concepts module library; this should allow specific machines to be tailored to individual requirements
- Physical prototyping and testing

We will focus on the initial phases of the above full life cycle so that the open modular concepts can be initiated and the final phases can be included in the future as the needs arise and when the modularity concepts have been integrated into the community.

2.1 Tools for Studying Environments, Tasks and Sector Specific Needs

The requirements have been broken down into areas that individual partners can contribute to based on their direct expertise and experience. The following sectors have been studied:

- Urban environments (Cybernetix)
- Domestic environments (Shadow Robot Company)
- Industrial environments (Caterpillar)
- Underground environments (University of Genova)
- Quarry environments (University of Genova)
- Construction environments (University Carlos III, Madrid and Shadow Robot Company)
- Space environments (Space Applications Services)
- Pipe and duct environments (ISQ and Fraunhofer-IFF)
- Land environments (QinetiQ)
- Sea environments (Helsinki University of Technology)
- Air environments (BAE Systems)
- Underwater environments (CSIC-IAI)
- Archaeological site environments (Politecnico di Torino)
- Mining environments (Őrebro University)
- Nuclear environments (CEA)
- Fire fighting environments (University of Genova)
- Facade cleaning environments (Fraunhofer-IPA)

Detailed reports on these environments and sectors have been produced (see Virk [7]), and it is clear from these that the range of operational sectors and tasks to be carried out is enormous. It is impossible to design and build individual and specific machines for all the possible uses if all the designs have to start from scratch (as is the current practise). This reinforces the need for a modular approach so that specific designs can be realized by plugging together the components needed to carry out the task in an individual application. Also, in this way, effort can be spent on the missing components that need to be developed and added to the existing module library.

2.2 Tools to Create Design Concepts

When mobile robots have been developed, it has usually been assumed that the design principles are known and the problem is one of applying them. In

reality, it is fair to say that there is no rigorous design methodology for mobile robots. In other words, given a formal description of the terrain that must be crossed, there are no general methods for specifying appropriate machine geometry, body articulation, suspension characteristics, traction mechanisms (wheels, tracks or legs), surface adherence mechanisms and actuation systems. This is a particular problem in the design of high-agility mobile robots for complex terrains. Most prototypes have either been the result of trial-and-error design methods or, in the case of legged machines, have been copied from nature with no scientific justification.

The most common approach has been the informal application of the designer's experience, pre-conceived ideas and preferences, in other words, a non-scientific approach. In many cases, more formal, but still subjective, design methods have been applied. These are usually based on some form of design matrix (e.g. showing design options versus performance metrics) and a subjective scoring system. A good example of this approach, drawn from the CLAWAR partners, is the methodology used to select the locomotion principle in the EU-IST ROBOVOLC project [8]. They used a design matrix showing locomotion options versus locomotion performance metrics. All of the consortium partners applied a subjective scoring system to complete their matrix and, then, the individual partner matrices were combined to get overall scores for the different locomotion options. This more formal approach reduces the chance of neglecting important aspects of the design problem, and the use of multiple opinions can increase the confidence in the result. However, the subjective scores still reflect the designers' pre-conceived ideas and preferences. A more scientific approach is required if mobile robot design is to move forward, leading to better designs, more able to achieve the tasks required of them.

In order to do this more objectively, tools are needed to formally define the environment and tasks to be carried out (using accepted standards) and develop expert systems with the knowledge to guide designers through the various phases of the design. Keys stages of the design include focussing on: machine geometry, traction methods, suspension characteristics, control systems, user-machine interface, sensing requirements and actuation and powering requirements,

2.3 Tools for Virtual Prototyping

In order to assess any design it is vital to see the concepts in as full a manner as possible. This can range from simple time response simulation to full VR immersion type of tests. The components that need to be realized to produce the concepts can also be studied and assessed in a similar manner. Hence tools to support the virtual prototyping of the components and the full systems are needed. An integrated design methodology for doing this is useful and hence the tools need to be fully structured. Solid modelling packages need to be coupled with tools for dynamic and structural

analysis as well as to the tools to generate digital mock-ups. In this way it is possible to allow a concurrent engineering approach so that early tests on virtual prototypes in their environments can be carried out. Figure 1 shows an example prototype robot that has been recently developed at PMAR Lab, Univ of Genova in the EC funded ROBOCLIMBER project (www.dimec.unige.it/PMAR/pages/research/projects/roboclimber.htm).

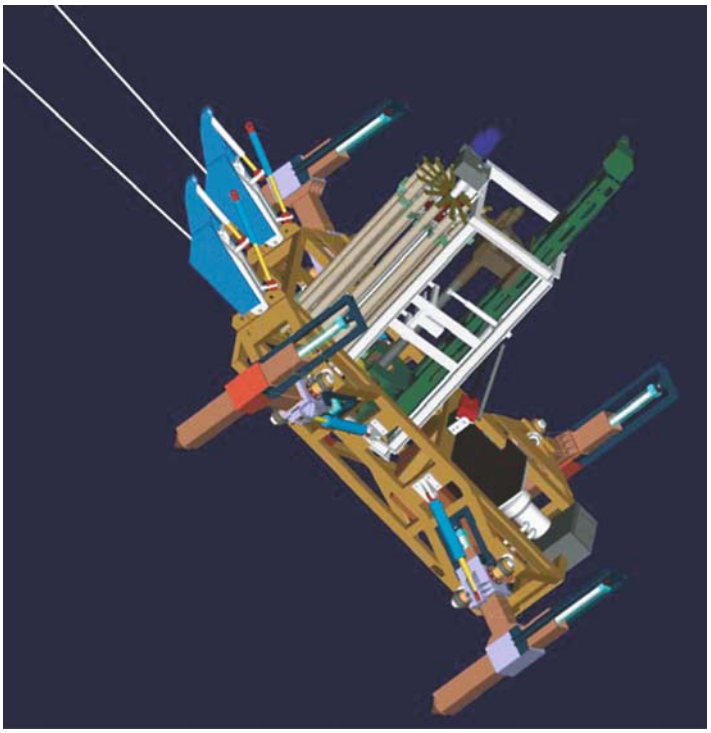


Fig. 1. University of Genova's ROBOCLIMBER prototype

2.4 Tools for Testing and Analysis

Once robot designs have been produced, tools are needed to assess them in some formal and objective manner. In this respect, the designs need to be assessed from several viewpoints. These include its mechanical characteristics (mobility, manoeuvrability, traction, obstacle clearance, etc) and reliability and robustness, usability, cost, etc).

2.5 Software Platforms

There are many software platforms available to develop the design tools as specified in this work. These range from freely available packages to expensive specialized products for specific sectors. In order to develop an open modular tool set it is clear that free shareware software needs to be used as the kernel. Some of the software packages that could be useful are the following:

- Player/Stage/Gazebo: This is a freeware integrated software environment that allows simulation of robotic systems. Stage is a 2D simulator capable of simulating large numbers of robots and Gazebo is a full physics 3D simulator. Player is a robot interface providing a network interface to a variety of robot and sensor hardware. The server/client architecture allows the control programs to be written in any language. These can be used for simulating virtual robots in Stage, Gazebo as well as real ones with no or little modification.
- CONFIG: prototype software tool integrating modelling, simulation, and analysis of several types of dynamic reconfigurable systems
- DynaMechs: A multibody dynamic simulation library (real-time dynamic and hydrodynamic simulation system
- Dynawiz XMR: A multibody dynamic simulation program for mechanisms and robotics design and analysis
- ODE: Useful for simulating articulated rigid body dynamics – for example ground vehicles, legged creatures, and moving objects in VR environments. It is fast, flexible, robust and platform independent, with advanced joints, contact with friction, and built-in collision detection
- JUICE: An animation workshop, with realistic physics for legged machines
- ADAMS : A virtual prototyping software
- ORCCAD: A software environment dedicated to the design and the implementation of advanced robotics control systems. It also allows the specification and the validation of missions and is mainly oriented towards critical real-time applications in robotics. Here, automatic control issues are strongly connected with discrete-event aspects. ORCCAD is particularly concerned with systems which strongly interact with their environment, through several actuators and sensors. ORCCAD has capabilities of validation of a mission, either by formal verification or by extensive simulation analyses

3 Conclusions

The paper has presented the latest work in Work Package 2 on modularity carried out by the CLAWAR partners. This work has specified the design tools needed to support open modular research and development in the wider community. The tools required should cover the entire cycle phases

from realising the need, formulating the requirements, designing and testing prototypes to producing commercial products and recycling them after their useful life is over. The software tools that are needed should be based around free shareware packages that have been produced. Over 150 such software tools have already been identified and from these, a few could be very useful to provide the functionalities needed to realise the component based design tool set to support and sustain the development of robot systems needed by the society and citizens of the future.

Acknowledgements

The author would like to acknowledge the support of the European Commission for funding the CLAWAR Thematic Network under contract G1RT-CT-2002-05080. He would also like to express his gratitude to the WP2 task members and in particular to those who have submitted reports that have contributed to this paper.

References

1. Virk GS, CLAWAR Technical Reports on Modularity, Tasks 1, 6, 11 and 16, EC Contract BRRT-CT97-5030, University of Portsmouth, 1999–2002
2. Virk GS (1999) Modularity for CLAWAR Machines – Specs and possible solutions, Proceedings of the 2nd Int Conf on Climbing and Walking Robots (CLAWAR'99), Portsmouth, UK, pp. 737–747, Professional Engineering Publishing, 14–15 Sept 1999
3. Virk GS (2000) Modularity of CLAWAR machines – Practical solutions, Proceedings of the 3rd Int Conf on Climbing and Walking Robots (CLAWAR 2000), Madrid, Spain, pp. 149–155, Professional Engineering Publishing, 2–4 Oct 2000
4. Virk GS (2001) Functionality modules – specifications and details, Proceedings of the 4th Int Conf on Climbing and Walking Robots (CLAWAR 2001), Karlruhe, Germany, pp. 275–282, Professional Engineering Publishing, 24–26 September 2001
5. Virk GS (2002) CLAWAR – robot component modularity, Proceedings of the 5th Int Conf on Climbing and Walking Robots (CLAWAR 2002), Paris, France, pp. 875–880, Professional Engineering Publishing, 25–27 September 2002
6. Virk GS (2003) CLAWAR modularity – the guiding principles, Proceedings of the 6th Int Conf on Climbing and Walking Robots (CLAWAR 2003), Catania, Italy, pp. 1025–1031, Professional Engineering Publishing, 17–19 September 2003
7. Virk GS (2004) CLAWAR Technical Report on Modularity – Design tools, EC Contract G1RT-CT-2002-05080, University of Leeds, 2004
8. ROBOVOLC project, www.robovolc.dees.unict.it/

Interaction Space Analysis for CLAWAR WP5 Societal Needs

M. Armada and M. Prieto

Control Department, Industrial Automation Institute, Madrid, Spain
armada@iai.csic.es

Abstract. In the last years of CLAWAR Network an interesting concept has arisen concerning the use of the interaction space to explain the modularity principles. Following the success of this approach it became interesting to ask if the same could be applied to other, non-technical tasks within CLAWAR project. So, a concept of interaction space for WP5 Societal Needs came to light. This paper summarizes the work done on WP5 and presents an analysis of this WP by using interaction space showing the interconnections among the different tasks targeted by the WP5 both along the three years of the project and also, specifically, for year two, where Education and Training, Employment and Quality of Life tasks has been addressed.

1 Introduction

CLAWAR Network WP5 is aimed at carrying out analysis and formative work for robotic system requirements from the viewpoint of the needs of society. The specific needs, of different sectors of society, will be considered and the benefits and drawbacks identified. The impact to society will be established and guidelines and regulations needing to be adhered to also identified if they exist or steps initiated to formulate them if they do not exist. Two-three tasks are carried out per year; each one focusing on a particular societal need and the most promising applications areas identified. The tasks selected for the year 2 are:

Task 5.1: Education and training

Activities for the task will focus on encouraging young people to become involved in the area of robotics. The activities will include competitions, demonstrations, workshops, etc.

Task 5.5: Employment

The task will give specific attention to predicting the impact of globalization and how the global work force growth will govern the potential robotics market. Specific sectors where robot technologies are already well adopted

will be used to formulate the effect of robotics on employment, e.g., the automotive industry. Using the practical experience from these sectors the impact on a more global scale will be formulated after detailed discussions within the membership and with others.

Task 5.6: Quality of life

The task will investigate how the new robots could contribute to improve quality of life. The aspects of life considered will include activities at work, during leisure, at home, in industry and commerce.

The tasks that were undertaken in Year 1 and those remaining for Year 3 are the following:

- Year 1: Task 5.1: Education and training; Task 5.2: Working conditions and safety
- Year 3: Task 5.1: Education and training; Task 5.3: Environment;

Task 5.4: Health

There is a close relationship among WP5 and other CLAWAR workpackages. WP5 will be used with the results of WP2-4 and WP6 on formulating socio-economic commercially viable new robotic markets and new robotic components. The results will be used in the formulation of new R&D project proposals. In order to determine the societal needs with respect to these aspects, it has been decided to consider these societal needs under a general framework where the following questions can be asked:

- General or focused automation of human activities?
- What sectors?
- Which benefits?
- Which priorities?
- Up to what extent?
- Implications?

The general scope of Task 5 can be recalled in [1] and the results from Year 1 are summarized in [2] and in the corresponding year report.

2 WP5 Interaction Space

In the last years of CLAWAR an interesting concept has arisen concerning the use of the interaction space to try to explain the modularity principles. Following the success of Prof. Virk decomposition of the modularity through the interaction space [15], it became interesting to ask if the same could be applied to other, non-technical, tasks within CLAWAR project. So, a concept of interaction space for WP5 Societal Needs came to light. As it can be noticed from the next diagrams (Fig. 1) a decomposition of WP5 using interaction space has been done, in order to better understand the implications and the interconnections among the different tasks targeted by the WP5 along the

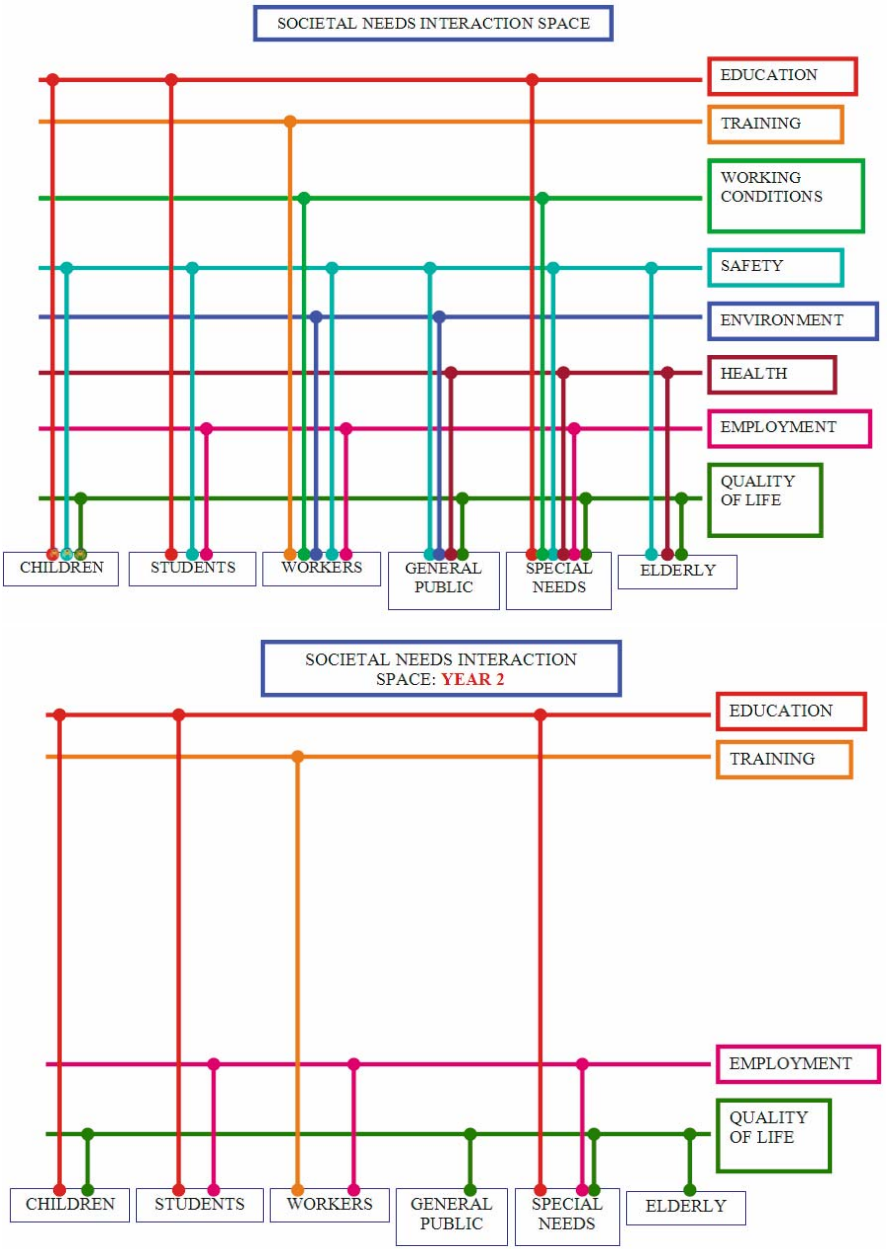


Fig. 1. Interaction space for WP5 Societal Needs

three years project. The first diagram shows the full WP5 interaction space, while the second diagram shows the interaction space addressed in the second year.

3 WP5 Second Year Activities

This paper is a summary the full report for the second year of WP5. In this introduction we can anticipate some of the results that are covered in greater detail in the next paragraphs. For the second year the activities have been carried out through several meetings, many contributions from partners, and three major events:

1. 3rd IARP International Workshop on Service, Assistive and Personal Robots – Technical Challenges and Real World Application Perspectives, held at the IAI-CSIC (Madrid), October 14–16, 2003.

Following the first IARP Workshop in Sydney (Australia, 1995) and the second in Genoa (Italy, 1997), the 3rd IARP International Workshop on Service, Assistive and Personal Robots – Technical Challenges and Real World Application Perspectives, was aiming to provide an overall picture of nowadays real significant robots in this domain. It was organised by the WP5 Co-ordinator, M. Armada, who is presently the Spain National Con tact Person for IARP. According to the general objective of the International Advanced Robotics Programme: "...encourage development of advanced robotic systems that can dispense with human work for difficult activities in harsh, demanding or dangerous environments, and to contribute to the revitalization and growth of the world economy", and taking into account the latest developments in service, assistive and personal robots that have been disseminated by the scientific community and are being anticipated by some commercial firms, it is noticeable a growing interest in this kind of advanced robots. The incorporation of new concepts and relevant innovations, the search for new applications and the use of state of the art supporting technologies will permit to predict an important step forward and a significant socio-economic impact of advanced robot technology in the forthcoming years. This was the main goal of the WS.

Apart from other very relevant contributions from the IARP community, there were important presentations coming from CLAWAR partners, and, in this way, it was opened an interesting exchange between IARP and CLAWAR. The CLAWAR contributions to this WS covered several aspects of WP5 tasks for the second year:

Keynote presentations:

- Robotics and automation in construction industry: from hard to soft robotics, by Carlos Balaguer (University Carlos III of Madrid, Spain).

Fig. 2. CLAWAR people participating with world renown scientists at IARP WS

- Modularity for service robots, by Gurvinder S. Virk (University of Leeds, UK).
  Regular presentations:

- Design of an active walking-aid for elderly people, by V. Pasqui, P. Mederic, P. Bidaud (LRP, France)
- RoboTab-2000: A Manipulator to Handle Plaster Panels in Construction, by P. Gonzalez de Santos, J. Estremera, E. Garcia, M. Armada (IAI-CSIC; Spain).
- The current state of robotics: post or pre-robotics?, by M.A. Salichs, C.
- Balaguer (University Carlos III of Madrid, Spain).
- Perturbed hamiltonian systems in robotics, by Juan I. Mulero Martinez, Juan López Coronado (Univ Politécnica Cartagena, Spain).

The WS Keynote Speakers (H. Asama, C. Balaguer, R. Chatila, R. Dubey, M. Hägele and G. S. Virk) seated at the end of the Workshop in a panel to further assess and discuss the domain directions and perspectives.

2. 13th International Symposium on Measurement and Control in Robotics (ISMCR'03), held at the IAI-CSIC (Madrid), December 14–16, 2003.

The 13th International Symposium on Measurement and Control in Robotics (ISMCR'03) was held last 11-12th December 2003 in the premises of the Industrial Automation Institute (IAI-CSIC), who co-organised the event with the International Measurement Confederation TC 17 (IMEKO TC 17). The Chairs of the ISMCR'03 were Dr. M. A. Armada (IAI-CSIC, Spain), Prof.

S. Tachi (Univ. of Tokio, Japan) and Dr. Zafar Taqvi (Communications and Tracking, International Space Station, USA).

ISMCR'03 was aiming to gather high quality original contributions in the robotics field and associated measurements with the final goal of assessing the most recent developments in this utmost domain of science and technology. The Symposium was organised in two parallel sessions and there were also four Plenary sessions with the following Keynote Speakers: Professor Susumu Tachi (The University of Tokyo, Japan), Professor Gurvinder Singh Virk (University of Leeds, UK), co-ordinator of CLAWAR Network, Professor Giovanni Muscato (Universita degli Studi di Catania, Italy) and Professor Jean-François Le Maitre (The University of Aix-Marseille III- LSIS-UMR CNRS, France).

The Symposium technical sessions covered a broad spectrum in this interesting are of robotics: Medical Robots, Sensors for Robots, Climbing and Walking Machines, Dynamic Control, Flexible Robots, Intelligent Vehicles, Virtual reality, Computer Vision, and Design. The contributions to the Symposium accounted for 53 high quality papers collected in a book. There were over 70 delegates attending the event.

CLAWAR network has provided outstanding contributions to ISMCR'03, as those of our two Keynote Speakers (Professors Virk and Mustato), long with several papers. CLAWAR partners like Bill Warren (Qinetiq) were of excellent help for the Symposium organisation. Another important contribution from CLAWAR was to the exhibition of advanced robots and related technologies, that was be set-up to illustrate the most recent results that are just leaving from important Research Institutes and Universities world-wide.

Fig. 3. CLAWAR partners attending ISMCR'03 and the robot exhibition

3. International Symposium on Robotics (ISR'04), held in Paris, organised by the IFR, where a Special Session and an exhibition stand on CLAWAR have been organised. Paris, March 2004. For the Special Session (TH32) there were 9 contributions from CLAWAR partners. These contributions covered

a wide spectrum, from bomb disposal and humanitarian demining, to tank inspection, passing through mobile robots for education and training, bio-robotic orthosis, or climbing robots for construction/inspection.

The exhibition was also a great success and several CLAWAR partners have contributed to this (HUT, Cybernetix, Robosoft). So, CLAWAR has been present in one of the major events of the robotic worldwide activity in the two main fronts: scientific contribution and technological demonstration.

4 WP5 Second Year Achievements

The second year achievements are summarised in what follows. We have divided the contribution received both from inside CLAWAR and from the other experts contributing following the three major events. Then we have performed an analysis of each contribution and placed a clear mark showing if the contribution is mostly related to:

- Education and Training
- Employment
- Quality of Life

Table 1 summarises this work. So, if reader is interested in a particular topic, it will be easy to select the concerned contributions.

Table 1. WP5 second year achievements classified by targets

Task	IARPWS	ISMCR'03	ISR'04
Education and Training	37	10-11-12-13	43
Quality of Life	19-20-21-22-23-24-25-26-27-29-30-31-32-36	3-4-9	40-41-42-44
Employment – Quality of Life	16-17-18-27-28-33-34-35-38-39	14	45-46-47-48
Education and Training Quality of Life	-	5-6-7-8	-

We have performed a small statistical work over the contributions, with the main idea of investigating what is the emphasis that the authors are placing when preparing their Research. So, as it can be noticed from the above graphic (Fig. 4), most contributions present a clear tendency to contribute to "quality of life" (near four times in respect to education and training, and more than two times in respect to employment). For this reason it can became "clear", with the natural precautions, that the trends in developing/applying new robotic technology are oriented to improve quality of life. This is something to

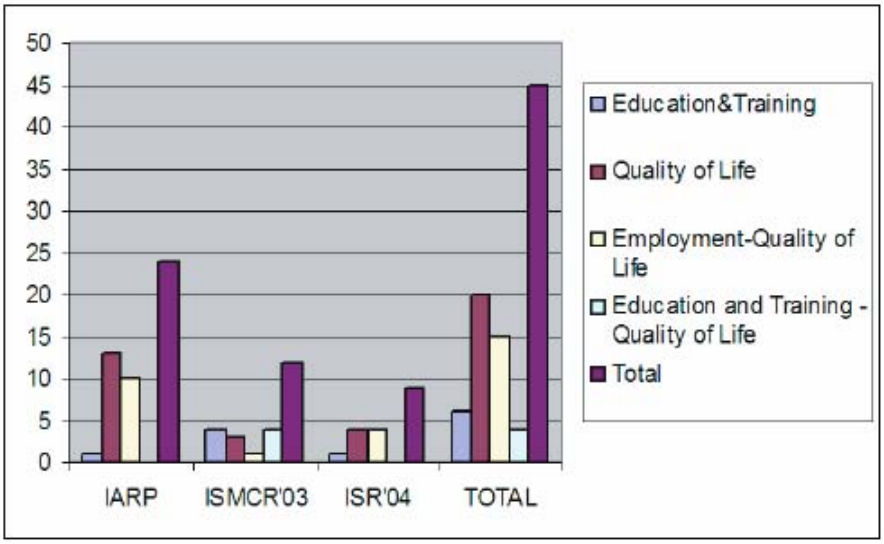


Fig. 4. WP5 second year achievements classified by number of inputs

be thinking out in our next activities in more depth, and also it confirms what we are observing in the market of new robots: home, medical, entertainment, etc.

5 Conclusions

The scope of WP5 for the second year has been addressed and positive analysis and formative work for robotic system requirements from the viewpoint of the needs of society was carried out. The two mainstreams: education & training and employment & quality of life of the foreseen activity were covered, and many experts have provided their points of view and their experience on the area under discussion. Some light and some structuring effect has been drawn over these, somehow difficult to understand (or at least difficult to translate to mathematical equations), matters. The concept of interaction space has been successfully translated both to help understanding WP5 and to organise it in a better way. Nevertheless, this second year work can be considered only as a step towards a full understanding on how the new robot technology will influence to society, and so, it will remain a lot of work to be done at the next CLAWAR meetings and workshops.

Acknowledgements

This WP5 year 2 paper has been mainly possible by the contributions provided by the WP5 members. To all of them our sincere thanks. Our acknowledgement is to be extended to other CLAWAR members, observers and external reviewers and experts that, with no obligation, were very kind in helping us. Special thanks are due to Georges Giralt, Susumu Tachi, Zafar Taqvi, and Jean-Guy Fontaine.

References

1. M. Armada, *WP5 – Societal Needs: Objectives*. CLAWAR Workshop: The role of CLAWAR in education, training, training, working conditions and safety,Madrid, November 2002.
2. M. Armada, *"The Role of CLAWAR in Education, Training, Working Conditions and Safety"*, Proc. of CLAWAR'03, September 2003, Catania, Italy.
3. R. Morales, A. González, P. Pintado, V. Feliu *A New Staircase Climbing Wheelchair*, Measurement and Control in Robotics, 2003, PGM, pp. 127–132.
4. P. J. Wright, G. S. Virk, S. C. Gharooni, S. A. Smith, R. I. Tylor, S. Bradshaw, M. O. Tokhi, F. Jamil, I. D. Swain, P. H. Chappell, R. Allen *Powering and Actuation Technology for a Bio-Robotic Walking Orthosis*, Measurement and Control in Robotics, 2003, PGM, pp. 133–138.
5. S. Val, C. Campos, M. Maza, S. Baselga *A New Methodology to Quantify the Effects of VR Systems on Users' Physiological Condition*, Measurement and Control in Robotics, 2003, PGM, pp. 61–66.
6. M. Maza, S. Baselga, J. G. Fontaine *Generation of Motion Sensation in VR Systems: The Driving Simulator SIMUSYS*, Measurement and Control in Robotics, 2003, PGM, pp. 33–40.
7. S. Val, C. Campos, M. Maza, S. Baselga *Influence of Driving Simulators on Users Physical Condition*, Measurement and Control in Robotics, 2003, PGM, pp. 67–74.
8. A. Lobov, E. López, J. L. Martínez, R. Tuokko *Communication components for Internet Robotics: ChessRobot.net Communication Chain*, Measurement and Control in Robotics, 2003, PGM, pp. 49–53.
9. H. Kajimoto, N. Kawakami, S. Tachi *Psychophysical Evaluation of Receptor Selectivity in Electro-Tactile Display*, Measurement and Control in Robotics, 2003, PGM, pp. 83–86.
10. M. Maza, S. Baselga, J. G. Fontaine *TELEDRIVE: Vehicle Teleoperation System for Application to Entertainment, Research and Remote Interventions*, Measurement and Control in Robotics, 2003, PGM, pp. 25–32.
11. H. A. Warren *Telepresence, Augmented Reality and Safety – Issues when Operating Mobile Robotic Systems in Unstructured Environments*, Measurement and Control in Robotics, 2003, PGM, pp. 41–48.
12. M. Iglesias, A. Noguera, G. Fernández, J. C. Grieco, M. Armada, R. Ponticelli *Remote Robotic Lab for Educational Purposes*, Measurement and Control in Robotics, 2003, PGM, pp. 91–94.

13. Giovanni Muscato *Beware (Or Be Happy...): The Robots Are Coming!*, Measurement and Control in Robotics, 2003, PGM, pp. 297–301.
14. J. F. Sarria, M. Prieto, H. Montes, M. Armada *High Speed Object Manipulation Using Computer Vision* , Measurement and Control in Robotics, 2003, PGM, pp. 297–301.
15. G. S. Virk *Open Modularity for Service Robots*, Measurement and Control in Robotics, 2003, PGM, pp. 297–301.
16. C. Balaguer Robotics and Automation in Construction Industry: From Hard to Soft Robotics, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
17. R. Saltaren, R. Aracil, M. A. Scarano Climbing Parallel Robot for Construction: A Study of its Behaviour, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
18. P. González de Santos, J. Estremera, E. García, M. Armada RoboTab-2000: A Manipulator to Handle Plaster Panels in Construction, Proc. 3rd IARP International Workshop on Service, Assistive and PersonalRobots, October 2003, Madrid.
19. Alami, R. Chatila, T. Simeon Planning Interactive Human-Robot Manipulation Tasks, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
20. Z. Yuan, J. Wu, Z. Gong, Y. Kanou, D. Yang, J. Chen. A Registration Method Applicable to outdoor AR System, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
21. W. Hamel Intersections Between Telerobotics and Assistive Robotics, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
22. R. Chatila Humans and robots: trends and perspectives, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
23. C. Pradalier, J. Hermosillo, C. Koike, C. Braillon, P. Bessiere, C. Laugier Safe and Autonomous Navigation for a Car-Like Robot Among Pedestrian, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
24. F. Lamiraux, J. P. Laumond, C. VanGeem Nonholonomic Path Deformation: Application to Trailer-truck Trajectory Optimization, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
25. J. Diard, P. Bessière, E. Mazer Combining Probabilistic Models of Space for Mobile Robots: the Bayesian Map and the Superposition Operator, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
26. L. Jun, G. Zhenbang, S. Hong Tri-dimensional System for Firefighting and Rescuing Based on the Aerial vehicles and Ground Mobile Robots, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
27. M. A. Salichs, C. Balaguer The Current State of Robotics: Post or Pre-Robotics?, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
28. R. Dubey Maximizing Manipulation Capabilities of Persons with Disabilities through Sensor Assistance and Skill Learning in Telerobotics, Proc. 3rd IARP

International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
29. V. Pasqui, P. Mederic, P. Bidaud Design of an Active Walking-Aid for Elderly People, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
30. D. Guiraud, P. Fraisse, P. Poignet Automatic Control Theory Applied to the Restoration of the Movement of Paralysed Limbs with Functional Electro-Stimulation (FES), Proc. 3rd IARP International WorkINTERACTION SPACE ANALYSIS FOR CLAWAR WP5 SOCIETAL NEEDS shop on Service, Assistive and Personal Robots, October 2003, Madrid.
31. M. Maza, S. Baselga, J. G. Fontaine An Application of VR and Telepresence Techniques for Home Care and Telerehabilitation, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
32. Y. Matsuoka Robotic Rehabilitation Environment Using a Safe Haptic Device, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
33. K. Kosuge, T. Oomichi Open Robot Architectures Strategy for Service Robots, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
34. J. Amat, A. Casals, J. Fernández Service Robotics. Technological Aspects that Render its Progress Difficult, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
35. A. Ollero, J. Alcázar, F. Cuesta, F. López-Pichaco, C. Nogales Helicopter Teleoperation for Aerial Monitoring in the COMETS Multi-UAV System, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
36. J. I. Mulero, J. López Coronado Perturbed Hamiltonian Systems in Robotics, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
37. P. Kopacek Robotics for Edutainment – Robot Soccer, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
38. H. Asama Future Perspective for Service Robots in Ubiquitous Computing Environment, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
39. M. Hägele Key Technologies and Pioneering Applications for Service Robots, Proc. 3rd IARP International Workshop on Service, Assistive and Personal Robots, October 2003, Madrid.
40. *Sandro Costo, Rezia Molfino, Matteo Zoppi* Bomb disposal robots: state of the art and an innovative solution for airplane security, Proc. ISR'04, March 2004, Paris.
41. *K. Berns, C. Hillenbrand* A Climbing Robot based on under Pressure Adhesion for the Inspection of Concrete Walls, Proc. ISR'04, March 2004, Paris.
42. *P. Gonzalez de Santos, E. Garcia, T. Guardabrazo and J. A. Cubano* Using Walking Robots for Humanitarian De-mining Tasks, Proc. ISR'04, March 2004, Paris.
43. P. Dutkiewicz, T. Jedwabny, M. Kielczewski, M. Kowalski, M. Lawniczak, M. Michalek, K. Kozlowski, D. Pazderski Mini Tracker V3.0 – a mobile robot for educational and research purposes, Proc. ISR'04, March 2004, Paris.

44. P. J. Wright, G. S. Virk, Sc Gharooni, S. A. Smith, R. I. Tylor, S. Bradshaw, M. O. Tokhi, F. Jamil, I. D. Swain, P. H. Chappell, R. Allen, Powering and Actuation of a Bio-robotic Walking Orthosis, Proc. ISR'04, March 2004, Paris.
45. S. Nabulsi, M.A. Armada Climbing Strategies for Remote Maneuverability of ROBOCLIMBER, Proc. ISR'04, March 2004, Paris.
46. *Domenico Longo, Giovanni Muscato* Design of a single sliding suction cup robot for inspection of non porous vertical wall, Proc. ISR'04, March 2004, Paris.
47. *Mario P. Ribeiro* In service inspection robotized tool for tanks filled with hazardous liquids – Robtank Inspec, Proc. ISR'04, March 2004, Paris.
48. Michaël Van Damme, F. Daerden, D. Lefeber Design of a pneumatic manipulator in direct contact with an operator, Proc. ISR'04, March 2004, Paris.

CLAWAR WP3 Applications: Natural/Outdoor and Underwater Robots

D. Longo and G. Muscato

Dipartimento di Ingegneria Elettrica Elettronica e dei Sistemi
Università degli Studi di Catania. Catania, Italy
gmuscato@diees.unict.it
robotics.diees.unict.it

Abstract. This paper summarizes the activities performed in the Workpackage 3 Application sector of the European thematic network CLAWAR. This represents the result of the joint work performed by the WP3 members during network meetings, workshops and special sessions organised.

For the second year two task were selected : Task 3.3: Natural / Outdoor robot applications and Task 3.4: Underwater applications.

1 Introduction

The main aim of CLAWAR WP3 is to formulate the requirements for CLAWAR machines in specific sectors that have already been identified as having good potential for automation [1].

Two tasks are carried out per year focusing on applications sectors that are felt to offer good potential for exploiting the CLAWAR technology. In specifying the requirements, the concepts of modularity and modular components that have been formulated are. In Year 2 (2003/04) the tasks were concerning Natural/Outdoor applications and Underwater robotics. The activities of the task consisted in analysing some application sectors in dangerous environments, in evaluating the capabilities of innovative locomotion systems and in the study of methods for autonomous navigation and in the analysis of the projects involving the CLAWAR members in underwater robotics [2].

2 Task 3.3: Natural/Outdoor Robot Applications

The activities of this task had some strong link with some tasks explored in the four years of the CLAWAR 1 network. In particular previous tasks related are:

Task 9 Humanitarian demining
Task 12 Outdoor applications

For these reasons it was decided to focus the activities of this task on three sub tasks:

- Dangerous environments
- Next stage type locomotion systems
- Methods for autonomous navigation

In the following sections details on some sub tasks will be exposed with reference to the partners that have contributed

2.1 Robots for Dangerous Environments

The following sectors have been analysed in some detail and are briefly reported:

- Fire-fighting
- Exploration of volcanoes
- Demining
- Quarrying and mining

Fire Fighting (QinetiQ, U.K, Fraunhofer IFF, Germany)

Mobile robots are becoming an option of choice where there is a need to remove people from danger. An example of the new commercial thrust has been to develop from existing military technology a range of remotely operated robotic fire fighting vehicles for use in a variety of commercial operations. The financial and human cost of fire fighting is substantial. Robotic systems have the potential to reduce this figure quite dramatically. For natural disasters such as forest fires and 9/11 type disasters, it may not be possible to solve these issues but there will be opportunities where levels of assistance may be appropriate.

As a risk reducing measure, robotic machines are a valuable addition to any fire-fighting team. Robotic fire fighters are well equipped to handle extreme temperatures and hazardous atmospheres. As well as the ability to contain the blaze, the machines are fitted with a low cost thermal imager and other onboard sensor options to allow them to investigate operational risk and provide statistical monitoring by assessment.

Two of the most prominent vehicles are “Firespy” and “Carlos” developed by QinetiQ. As might be expected, given the company’s history, the systems developed by QinetiQ are not aimed at the fire service’s bread-and-butter deployments but are intended for use in emergency situations. The robots have been designed for robustness and reliability in a variety of hazardous situations such as large-scale petroleum fires; aircraft engine fires; hazardous

chemical leaks; and fires with the risk of building collapse. QinetiQ's two fire-fighting robots can therefore cope with a range of applications.

A detailed analysis of the requirements of a fire fighting robot has been also carried out by Ulrich Schmucker of the Fraunhofer IFF [3].

Exploration of Volcanoes (Univ. Catania, Italy)

Today there are 1500 volcanoes on the Earth potentially active, 500 of them have been active during last 100 years and about 70 are presently erupting. Ten percent of the world population live in areas directly threatened by volcanoes, without considering the effects of eruptions on climate or air-traffic for example. About 30.000 people have died from volcanic eruptions in the past 50 years, and billions of euros of damage have been incurred. As a consequence it is important to study volcanoes and develop the technologies to support the volcanologists in this process. In the last decade alone, due to both the unpredictable timing and to the magnitude of volcanic phenomena, several volcanologists have died surveying eruptions. So a major aim of a robotic system is to minimise the risk for volcanologists and technicians involved in working close to volcanic vents during eruptive phenomena. It should be noted that observations of, and measurement of the variables relating to volcanic activity are of greatest interest during paroxismal phases of eruptions, which unfortunately are also the time of greatest risk for humans. Among the projects for volcano exploration we can mention the Dante II walking robot developed by Carnegie Mellon University and NASA, the RMAX helicopter developed by YAMAHA and the recently EC funded ROBOVOLC project [4].

Demining (RMA, Belgium)

There exists a big difference between the military and the civilian mine clearance. The military demining operations accept low rates of Clearance Efficiency (CE). For these purposes it is often sufficient to punch a path through a minefield. For the humanitarian demining purposes, on the contrary, a high CE is required (a CE of 99.6% is required by UN). This can only be achieved through a "keen carding of the terrain and an accurate scanning of the infested areas" that implies the use of sensitive sensors and their slow systematic displacements, according to well-defined procedures or drill rules, on the minefields. The robots, carrying the mine-detectors, could play here an important role. Obviously, the automatization of an application such as the detection, the marking and removal of AntiPersonnel mines implies the use of autonomous or teleoperated mobile robots following a predefined path, sending the recorded data to their expert-system (in charge of processing the collected data), marking the ground when a mine is detected with a probability of predefined level and/or, possibly removing the detected mine. This complete automatization is actually unrealistic and may surely not be entrusted to a unique robot: the technologies allowing it exist, but the integration of those

technologies in mobile Robotic Systems moving in unpredictable outdoor environmental conditions is not yet mature.

Prof. Y. Baudoin of the RMA Belgium has proposed an Humanitarian Demining catalogue intended to (1) disseminate the results of research-activities related to the development of mobile roboticized carriers of detection sensors with the support of the GICHD (Geneva International Centre for Humanitarian Demining), (2) assist actual and future T&E activities of ITEP (International Test and Evaluation Programme), (3) help the research-centres focusing on this kind of solutions to refine and continuously adapt and update generic modules that may be transferred to useful Robotics Systems. This catalogue includes existing robotics systems but also current projects focusing on the possible use of roboticised solutions.

A detailed classification of the requirements for a robot for humanitarian demining has also been prepared subdivided into three levels:

– the system-level requirements
– the robot-level requirements
– the mobility-level requirements

A IARP-EURON-CLAWAR International Workshop on robotics and Mechanical assistance in Humanitarian Demining and similar risky interventions has been recently organised by RMA in Brussels-Leuven, Belgium, on 16–18 June 2004

Quarrying and Mining (Univ. Genova, D'Appolonia, SPACE AS)

Quarries are an important economic source for many countries. Working in quarries is a good job opportunity, but is very hard and dangerous. In fact, mineworkers work at open-air, subject to all weather conditions, in a dangerous environment. Quarries change aspect every day because as mining proceeds the ground and walls change shape. The infrastructures such as energy supply cables and access paths result temporary and unreliable. The problem of safety in quarries is urgent to be solved; especially for poorer countries where protection for workers is not foreseen as life value is low, although regulation to prevent accidents exist, but for richer countries as well, because, although protections are higher, accidents happen every day because satisfactory technology, able to guarantee mineworker safety, doesn't exist. Some recent project on quarrying and mining are the ROBOCLIMBER [5], the Microdrainage, and the applications developed by Caterpillar.

2.2 Next Stage Type Locomotion Systems

Innovative robots for outdoor robotic applications require new locomotion systems. The analysis performed within WP3 was concentrated on the following typologies:

- Hybrid locomotion
- Peristaltic locomotion
- Self-reconfigurable modular systems
- Aerial systems

Hybrid Locomotion (HUT, Finland and Univ. Catania Italy)

The design of a new mobile robot is process that entails satisfying several requirements with respect to the environment in which the robot will be asked to work. Most classical solutions adopt wheels as a locomotion technique. It is, however, clear that wheeled robots have some limitations in highly unstructured environments or in difficult structured environments (like stairs). On the other hand the current technology for walking robots is not yet so advanced as to represent a valid alternative. Most current walking robots are very slow, have a low payload capability, are difficult to control and, due to the high number of sensors and actuators, have low fault tolerance capabilities.

A way to find a compromise between these two solutions is to choose a hybrid system adopting wheels and legs together. Several examples of hybrid robots exist in the literature and belong to two main categories.

The first category includes articulated-wheeled robots, with the wheels mounted at the end of legs. This category comprises the WorkPartner robot [6–8] with its particular kind of locomotion called rolking, the Roller Walker [9], where the wheels are not actuated but the robot simply skates, the biped type leg-wheeled robot developed by Dr. Matsumoto of AIST [10, 11] able to climb up stairs, the Walk'n Roll with four legs with a wheel attached at the end of each leg [12] and the mini-rover prototype developed by LRP [13], just to name a few examples. Most of these systems can have three type of locomotion: wheel mode, where they act as a conventional wheeled mobile robot; step mode, where the wheels are locked and they act as a simple legged system and hybrid mode with cooperation of legs and wheels.

The second category consists of robots with wheels and legs separated, but acting always together to locomote the system. This category includes for example the Chariot II robot [14, 15] that has four articulated legs and two central wheels, the RoboTrac robot [16, 17] with two front legs and two rear articulated wheels, the hybrid wheelchair proposed by Krovi and Kumar [18] and the ALDURO, [19, 20]. This category of robot has usually only the hybrid type of locomotion but are from a mechanical point of view more simple and consequently with a more light weight.

Peristaltic Locomotion (University of Genova, Italy)

Peristalsis has always been deeply studied because of its diffusion in nature. This mechanism is often used when a fluid has to be pushed in contact with a natural surface, such as blood in veins and other biological fluids, and some low-order animals use it for their locomotion.

All these animals are invertebrates, but there are differences between them according to different anatomies and behaviours. The most studied from the point of view of locomotion are snails and earthworms. Main applications of these robots are for diagnostic and surgery purpose, but several prototypes have been designed and built also for disaster-relief.

Self Reconfigurable Modular Systems (LRP, France)

In the last years, many efforts have been made in research in modular robotics and especially in self-reconfigurable systems (SRS), i.e. systems which are able to change dynamically their topology. Such systems are made of sets of mechatronic building blocks which have the capacity to connect and disconnect together. By manipulating themselves their own building blocks, those systems can change their topology. The advantages of modular systems are several:

- Easy and fast to deploy: the robot does not need to be designed and constructed; the modules are already available and can be fast and easily manually assembled. The required topology for a task is automatically generated by software and the control algorithm is downloaded to the robot.
- Easy and fast maintain: a deficient module can be easy and fast replaced.
- High versatility and little stuff: a set of modules can be assembled in various topologies optimised for various tasks.
- Adaptability: it is possible to perform tasks which were not originally foreseen.
- Low cost: the genericity of the modules authorizes mass-production

Furthermore, a self-reconfigurable system has even more advantages:

- Autonomy: the robot can reconfigure and repair itself
- Increased versatility: the robot can adapt its topology during a task, and not only before.

The applications of SRS are numerous: i.e. industrial manipulations, locomotion on rough and difficult terrain, pipe inspection, space station construction, etc.

Because of their versatility and robustness self-reconfigurable systems are very interesting for space applications. A set of robotic modules could do a very wide range of tasks, which would else need numerous big and heavy non-modular robots, including tasks which are not originally foreseen. Moreover a low gravity context reduces tremendously the mechanical constraints and permits much more efficient SRS. The system can move more economically, carry heavier loads and form bigger structures. All candidate sites in space, like orbital stations, asteroids, moons, and non-gaseous planets, have less gravity than on Earth.

The mechatronic design of self-reconfigurable robotic modules is quite complex because it involves a high degree of technology for energy storage or

production, efficient and low energy consumption of the connexion mechanism, distributed control system, etc.

The optimisation of module characteristics (like connectivity, mobility, geometry) which gives the best versatility is a non-trivial problem because they are only components of a modular systems which should be able to form an undefined number of different topologies, depending of the task and the context, like the local shape of the terrain. Therefore the modules design must be optimised for some main and general functions which must be defined first.

Aerial Systems (BAESYSTEMS, U.K.)

Over the recent years the use of low cost Uninhibited Aerial Vehicle (UAV) for civilian applications has evolved from imagination to actual implementation. Systems have been designed for fire monitoring, search and rescue, agriculture and mining. In order to become successful the cost of these systems has to be affordable to the civilian market, and although the cost/benefit ratio is still high, there have been significant strides in reducing this, mainly in the form of platform and sensor cost. Many new research projects are currently developed with the aim to build totally autonomous systems, groups of cooperating UAVs and to study the cooperation between ground vehicles and UAVs.

3 Task 3.4 Underwater Robots

Nowadays there is a wide adoption of robots for underwater applications ranging from scientific explorations to maintenance and inspection of pipes and cable, to ship inspection and cleaning, just to name a few. Among all sectors mobile robotic seems to be one of the most promising area for the next future. For these reasons Task 3.4 of the CLAWAR network Year 2 has been devoted to Underwater Applications. In this task some of the CLAWAR partners presented their experiences and projects on underwater robots. Some of these projects are here briefly summarised.

3.1 Helsinky University of Technology

Three recent projects on underwater robots in which HUT was involved were SUBMAR, ROTIS and SWARM.

The SUBMAR systems acts following the concept of Robot Society where a group of autonomous individual robots operate collectively towards a common goal. This has strong analogy from nature – social animals and insect colonies (ants, bees, termites, etc.), seemingly chaotic local behavior and distributed intelligence eventually forms complex and efficient high-level structures. Novel type of robot platform have been designed and tested for mobile process instrumentation and 3D on-line monitoring. These small spherical

robots including inside all the sensors, control, communication systems and actuators for depth control. These robots are passively transported within the fluid they are monitoring. Several tests, conducted in laboratory, have been shown, demonstrating the possibilities of these systems. Potential future applications are the monitoring of flow-through processes in petrochemical industry, pulp mills, water purifying plants, the validation and control of batch processes in food processing, fermentation, or chemical and pharmaceutical industry and environmental surveillance, i.e. detection of pollutants, and various oceanographical studies, such as the mapping of sea currents.

The second project is ROTIS (Remotely Operated Tanker Inspection System) an EU-project, coordinated by Tecnomare in Italy, which aims at development of a prototype system for tanker ship inspection based on a smart ROV concept. The ROTIS project deals with the design, development and testing of a novel remote inspection system. It should be capable of navigating inside the ballast tanks of tankers and other cargo vessels, through the standard man-holes and openings and perform the close visual inspection and thickness gauging. Successful implementation of ROTIS is expected to lead to better, safer and cleaner maritime vessel inspections. In this project HUT developed intelligent sensor based tether deployment and management unit.

Last project presented is SWARM (Autonomous Underwater Multi-Probe System For Coastal Area/Shallow Water Monitoring) a CRAFT project which aims at the underwater monitoring for shallow water areas. The system consists of multiple, homogenous, small size, reasonably priced, robust and easy to use underwater robotic probes that can perform two-week missions autonomously. The probes control their buoyancy but move otherwise freely with the water flows. They communicate with each other and with the control station (acoustic modem/radio/GSM/satellite), and localize themselves. In this project a platform that can measure biological and physical variability at the scale relevant for single event (meter-kilometer and second-day scale) will be developed and tested in the Baltic Sea.

3.2 University of Genova

The SBC (Sub Bottom Cutting) project is an European research project funded under 5th FP, whose activities started in March 2001 with a duration of 2 years. It covers the development of SBC: a new-concept Sub-Bottom Cutting robotic system, enabling a dig-and-saw process, able of removing-and-convoying the subsurface soil sediments, with low turbulence and without supply of materials, resources and energy, enhancing negative impacts. The final system consisted of three parts:

The support platform whose main aim is the positioning, leveling and anchoring to the seabed soil the robotic system. Even if traditional tracked locomotion is foreseen for the final machine, at present under the platform are

located suction anchors for docking and horizontally setting purposes that are detachable for transportation and for emergency release.

The excavating arm: a 2dof arm made by a serial cinematic chain: first member, arm, is linked through a revolute joint to the platform and second member, forearm, can slide along the arm. It is made by twin tubes endowed, at their front ends, with excavating heads and with a dredging system, comprising the water and slurry piping inside the twin tubes.

The cutting end effector: it consists of a moving loop of diamond wire supported on pulleys. The pulleys and their power and control systems are mounted on a carriage fed down the forearm twin tubes.

Several results of simulations and videos of test of the system in real operations in the sea were shown and commented.

3.3 Budapest University of Technology

Budapest University of Technology presented a project on the design, development and realization of a multisensor system to detect unexploded ordnance and other type of undefined objects in the Lake Balaton. Lake Balaton is the largest lake of Hungary and also the largest freshwater lake of Central and Western Europe. Virtually the entire territory of Hungary was the scene of heavy fighting between the Soviet, German and Hungarian armies during the Second World War. Soviets were also supported by the British-American aviation. As a result, several millions of mines were installed and their exact numbers are unknown.

In particular the Balaton region was the theater of intensive military operations during the period of December 1, 1944 – March 30, 1945, which is also referred to as the Battle of Balaton. Around 200,000 mines were installed on the ice of the frozen lake.

Objectives of this project were then the detailed study of archives and other data sources in order to assess the nature and conditions of objects in the lake, the detailed study of satellite images (Iconos) and aerial photographs and the conception and realization of a (semi-)autonomous vessel integrating sensor and localization equipment.

In particular results of different shallow water remote sensing platforms were discussed, including Satellites (spectral, spatial, temporal & digital-radiometric resolution), Aircraft (low & high altitude), Buoys (tethered & untethered – surface & subsurface – autonomous, Shipboard (Surface & subsurface), AUV's (Autonomous Underwater Vehicles) and Fixed location (land, bottom, moorings).

3.4 University of Catania

University of Catania is involved in the development of some innovative control strategies for the attitude control of Underwater vehicles. This activity was done within the TECSIS project, a national project coordinated by ENEA

on diagnostic technologies and intelligent systems for the development of archaeological parks in south Italy. In particular the adoption of quaternionic neural networks to improve the behaviour of a PD attitude controller was proposed.

3.5 Bulgarian Academy of Sciences

The conflicting performance demands on underwater robots give rise to complicated design requirements. The mechanical subsystem should be as light and rigid as possible, the actuators should be as powerful and quick in response as possible. Moreover sensors and actuators should be optimally placed in the vehicle. The controllers should be time/energy efficient and optimally robust in the presence of arbitrary disturbances. From these considerations it was clear the need for quantitative relations between the design parameters and an overall design criterion that could lead to best possible dynamic performance of the UR.

As a design guideline was proposed to decompose the overall problem into a series of easy to solve design problems for the sub-systems.

Integrated structure-control design criteria and optimal trade-off design relations were proposed based on necessary and sufficient conditions for robust decentralised controllability. They enable us to decompose the overall design problem into several iterative design solutions for its subsystems: mechanical structure, actuators, and controllers. The design recommendations are not in conflict with the basic design requirement for strength/load capacity and the proposed design specifications can be used in developing new modularity concepts in designing mechanical, drive, and control subsystems of underwater robots.

3.6 CYBERNETIX

Cybernétix Offshore Division develops robotic systems for inspection, maintenance and repair of deep water facilities. Main applications are deepwater intervention, Inspection/Maintenance and Construction.

The presented projects were “ICARE” for Chain inspection and cleaning, “Octopus” for inspection, cleaning and painting of ship hulls and the Intervention AUV (I-AUV) “ALIVE”, financed by the European Commission in the frame of the the “GROWTH” Programme. The objective of the ALIVE project was to build an Intervention AUV able to execute telemanipulation tasks on sub-sea structures. The vehicle is controlled only by an acoustic communication link to the surface (no umbilical).

4 Conclusions

In this paper only some of the activities carried out within the WP3 of the CLAWAR network have been briefly summarised. For a more complete

description please refer to the Network Web site [1] or to the complete report [2].

This work has been prepared using some part of the contributions received by many CLAWAR members. Their help is gratefully recognized and acknowledged.

References

1. The CLAWAR Network, http://www.clawar.net.
2. Muscato G (2004) CLAWAR 2 WP3 Application sectors Year 2 Report. GIRT-CT-2002-05080, May 2004
3. Schmucker U (2003) Fire fighting robots requirements. Notes from CLAWAR meeting Catania 09/2003 and Karlsruhe 11/2003. See also [1]
4. The Robovolc project homepage http://www.robovolc.diees.unict.it
5. The Roboclimber web pages http://www.t4tech.com/roboclimber
6. Halme A, Leppanen I, Salmi S (1999) Development of Workpartner Robot – Design of actuating and motion control system- CLAWAR 99, Portsmouth
7. Halme A, Leppanen I, Salmi S, Ylonen S (2000) Hybrid locomotion of a wheel-legged machine. CLAWAR 2000 International Conference on Climbing and Walking Robots, Madrid, Spain, 2–4 October 2000
8. Leppanen I, Salmi S, Halme A (1998) Workpartner – HUT Automations new hybrid walking machine. CLAWAR 98, Brussels 1998
9. Endo G, Hirose S (2000) Study on Roller-Walker (Multi-mode steering Control and Self-contained Locomotion). ICRA 2000, International Conference on Robotics & Automation, San Francisco
10. Matsumoto O, Kajita S, Saigo M, Tani K (1998) Dynamic Trajectory Control of Passing Over Stairs by a Biped Leg-Wheeled Robot with Nominal Reference of Static Gait. Proc. of the 11th IEEE/RSJ International Conference on Intelligent Robot and Systems, pp. 406–412
11. Matsumoto O, Kajita S, Saigo M, Tani K (1999) Biped-type leg-wheeled robot. Advanced Robotics, Vol. 13, No. 3, pp. 235–236
12. Adachi H, Koyachi N, Arai T, Shimuzu A, Nogami Y (1999) Mechanism and Control of a Leg-Wheel Hybrid Mobile Robot. Proc. of the IEEE/RSJ International Conference on Robotics and Automation, pp. 1792–1797
13. Benamar F, Bidaud P, Plumet F, Andrade G (2000) A high mobility redundantly actuated mini-rover for self adaptation to terrain characteristics. CLAWAR 2000 International Conference on Climbing and Walking Robots, Madrid, Spain, 2–4 October 2000
14. Dai YJ, Nakano E, Takahashi T, Ookubo H (1996) Motion Control of Leg-Wheel Robot for an Unexplored Outdoor Environment. Proceedings of the 1996 IROS Intelligent Robots and systems, Nov. 4–8, 1996
15. Nakano E, Nagasaka S (1993) Leg-Wheel Robot: A Futuristic Mobile Platform for Forestry Industry. Proceedings of the 1993 IEEE/Tsukuba International Workshop on Advanced Robotics", Tsukuba, Japan, Nov. 8–9, 1993
16. Six K, Kecskeméthy A (1999) Steering Properties of a Combined Wheeled and Legged Striding Excavator. In: Proceedings of the Tenth World, Congress on the Theory of Machines and Mechanisms, Oulu, Finland, June 20-24, vol. 1, pp. 135–140. IFToMM

17. Schuster C, Krupp T, Kecskemèthy A (1997) A non-holonomic tyre model for non-adhesive traction and braking manouvres at low vehicle speed. In: Proceedings of the NATO Advanced Study Institute on Computational Methods in Mechanisms, Varna, Bulgaria, June 1997
18. Krovi V, Kumar V (1999) Modeling and Control of a Hybrid Locomotion System. In: ASME Journal of Mechanical Design, Vol. 121, No. 3, pp. 448–455, September 1999
19. Hiller M, Müller J, Roll U, Schneider M, Schröter D, Torlo M, Ward D (1999) Design and realization of the anthropomorphically legged and wheeled Duisburg robot ALDURO. In: Proc. of the 10th World Congr. of theory of Machines and Mechanisms. IFToMM, Oulu, Finnland, 1999
20. Müller J, Hiller M (1000) Design of an energy optimal hydraulic concept for the large-scale combined legged and wheeled vehicle ALDURO. In: Proceedings of the 10th World Congress on the Theory of Machines and Mechanisms, Oulu, Finland, June 1999

CLAWAR 2 Work Package 6 (WP6) – Assessment Year 2, May 2004. Economic Prospects for Mobile Robotic Systems – Exploitation and Risk

H.A. Warren and N.J. Heyes

QinetiQ Plc, Farnborough, UK

Abstract. The paper describes the topics and inputs from the members of CLAWAR2 in assessing the expansion of mobile robotic markets and how risk exploitation has been addressed. A workshop was also held and notes are presented. A questionnaire/survey being conducted is also described.

CLAWAR Contributors: Task Leader and preparation: H A Warren, QinetiQ.

Task Members and technical inputs:Robosoft, Robonetics, Shadow Robot Co, Roboscience, Lego Systems, Univ. of Leeds, Univ. of Catania, FZI, Fraunhofer-IPA, IAI-CSIC, RMA-FUB, Poznan Univ. of Technology, CEA, ISQ, Cybernetics, CRF, Caterpillar Europe, Politechnico di Torino, King College London.

Workshop Presenters: H Warren – QinetiQ, M Haegele – IFR & Fraunhofer IPA, F Cappello, Media IRC, K Althoefer – Kings College London. Web and questionnaire/survey: N Heyes – QinetiQ.

1 Introduction

This year's task focuses on the continuing need to identify new market sectors for mobile robots and systems. In year 1, a generic approach was adopted to explore new activities in sectors already exploiting robotic technologies such as the nuclear, construction and automotive industries. Tasks are now split into sub-sections with specific activities to assess applications and the risks involved in their exploitation. Levels of economic viability are investigated and recommendations made in order to focus on the future development of new products. We have aimed to identify end users at an early stage to maximise the benefits for users and stakeholders. This paper describes, exploitation of existing, new markets and risk types, seeking rules and guidelines and standards, considerations for health and safety issues, economic impacts of service robotics world wide, surveys of expert opinions in the robotics field, discussion at the robotics workshop and individuals inputs. There are conclusions and recommendations on the way forward for year 3 and beyond.

2 The Task Objective

WP 6.2, the topic to be identified and reported is EXPLOITATION and RISK ASSESSMENT: The task studies the aspects related to exploitation, health, safety and risk assessment of the markets and systems identified in Task 6.1. This involved gathering information on the current market status, identifying areas where rules and regulations are required and encouraging the dialogue between different organisations to assist in the development of standards. Stakeholders' viewpoints are required to remove existing generic concerns and speed up widespread acceptance of robotic systems. A survey aims to identify the economic prospects for mobile robotics.

In the first years work package WP6, we identified some new markets and assessed the economic prospects for mobile robotics by identifying future markets, predicting volumes and prioritising the importance to society. However, as year 1 evolved, all was not clear as to the way forward for the projected time scales, quantities and most of all the identification of generic but detailed new markets and End Users. This proved to be more complex and greater input was found necessary to provide a detailed statement. Data collection came from a selection of sources:- IFR World Robotics future projections for service robotics, the views from SMEs and large industrial organisations, from End Users and by conducting CLAWAR surveys from industrial manufacturers and from CLAWAR experts/members.

In this second year we have aimed at addressing the key missing ingredients for the absence of a mass market in mobile robotic systems. We needed ideas to be substantiated for a new market era and looked at the risk and exploitation issues for volume mobile robotics. CLAWAR have members gathering information on the current status of mobile robotics markets and attempted to identify areas (from their own expertise) where rules and types of regulations are required. We have encouraged dialogue between different bodies and organisations with an aim to help in the development of guidelines that are acceptable from all stakeholders' viewpoints. We have aimed at removing existing generic concerns and hope to speed up the widespread acceptance of mobile robotic systems. We have also looked at the IFR interpretation on the exploitation of mobile service robotics, taken from their world 2003 robotics [1] statistical report and analyses. Our questions in year 2 are; what are the risks, what are the markets and how are they to be exploited?

3 The Requirement Definition

This year 2003/2004, CLAWAR members were asked to contribute and agreed to provide exploitation and risk assessment information to reflect upon their personal experiences, knowledge and ideas and also where possible, their organisation's business planning ideals for future mobile robotic products. It was decided to not be too specific on what each member should provide in

order to take a type of Delphi study approach. (The Delphi Method is based on a structured process for collecting and distilling knowledge from a group of experts by means of a series of questionnaires interspersed with controlled opinion feedback). We needed a general picture from as many diverse angles as possible and as we have universities, research organisations, SMEs and Industry in the work package group a wide range of ideas should be possible. Individuals were asked to put themselves in Users shoes as well to look at the problems from a Users perspective. We also gained more information by organising a workshop held at FZI on 26th November 2003. Recommendations from this workshop are detailed later.

4 Member Information Request

An information request was sent out with the following definitions of information required from the team members and was discussed at regular stages of the report production.

4.1 Rules and Guidelines (Standards)

Think about what guidelines are likely to be required or to be accepted (if any). What you would be prepared to accept if you are a User, if you are a Manufacturer or if you are a Service operator.

What will guidelines cover? general guidelines to cover safety of operation, acceptance criteria, morality issues of use, gender issues, green issues including design concept to recycling of components. Detailed guidelines for manufactures such as bus protocols, mobility acceptability, size, operational requirements, – e.g. commercial/industrial/in-the-home/office or outside in multitude of environments. Assess how or where your statements relate to either exploitation or risk or both. Consider if there is a definitive link with WP2. Are the tasks duplicated or the same (perhaps only for those members also working on WP2).

4.2 Exploiting

How can we exploit, what can we exploit, how do we go about doing this, what are the areas and what and where are the likely volumes of mobile robotic products. Has anyone ideas on possible costs involved, both hidden and real? There is money to be made and this is what exploitation is all about. I believe that whilst exploitation is an essential ingredient to success we should consider the world wide social and personal benefits to mankind, not purely for justification of the product but as a genuine reason to make the world a better place.

4.3 Health & Safety Issues

Try to assess what the limit of H&S is required for safe operation of mobile robotic platforms (with manipulation). Provide at least one or more different case scenarios from your research and development work. Give examples of modes of operation and categories where equipment could be used. e.g in hospitals, care homes, restaurants, hotels, factories, farms etc. Put your hat on as a manufacturer and say what you believe you could "get away with" as the minimum level of H&S to sell a product. Who is liable if there are claims due to neglect or mal-function? Consider what the reliability issues are in the equation of sales and manufacturing success.

4.4 Risk and New Markets

We failed to identify in WP6 year 1 quantities of new markets. We did however identify that there are several areas that have possibilities. It is these that we need to consider the risk for the future markets. They are as described in the recommendations of the WP6 year 1 report.

- We need to address the domestic market first and make unit costs very low.
- Medical applications are favoured using exoskeletons for disabled people and consumer products to assist in home-health and monitoring. Dissemination in this field is required.
- Novel disaster recovery robots can be used to assist after the event. Demonstrators are required to actively tackle disasters. This way awareness profiling will be raised.
- Universal mobile assistants are required for complex tasks in the household. Expansion of the limited marketing already started will hasten the pace to volume production.
- Production quantities of existing design and novel edutainment toys are likely to increase. New application areas should be sought.
- Members were asked to give risk considerations for the above market and other possible areas.

4.5 Old Markets

This refers to existing markets – how and where we can exploit, expand or form new ones from them. Please get your thinking caps on and start by making some suggestions on which markets have the best opportunity of success through exploitation.

4.6 Questionnaire/Survey

Neil Heyes, QinetiQ has sent details on what is required for the survey from the members. Two studies will be conducted: A Delphi study for the technical

members and later in year 3 a survey will be sent to a wider audience – to assess social acceptance of future mobile robot usage and interaction with public use. A pilot questionnaire has been published. 20 sets of results have been returned. We have to date identified a need to re-structure the questions for year 3.

5 Abridged Summary of CLAWAR Member Inputs

H Warren CEng MIEE – Task Leader, QinetiQ, UK

Areas for exploitation: –information taken from UK Health and safety Commission. [2]. Workplace Safety is not so fine – UK H&S Executive – November 2003. New figures show how dangerous Britain's workplaces remain. The Health and Safety Commission announced that 226 people were killed in work-related accidents around Britain in 2002–03 – 25 less than in 2001–02. But there were 28,426 major injuries reported – 415 more than the previous year. Despite the increase in major injuries, the average fine for health and safety offences fell by 21 per cent last year – from £11,141 to £8,828 – compared to a sharp rise of 39 per cent in 2001–2002. This was partly because of fewer larger fines. The number of court actions taken by HSE was down on the previous year by 15 per cent, although it issued 13,263 enforcement notices, an increase of 20 per cent over 2001–2002. Last year 933 companies, organisations and individuals were convicted of health and safety offences. Agriculture and construction both received a far higher number of improvement and prohibition notices than in the past. In agriculture, improvement notices rose by 250 per cent, from 429 to 1,503 in 2002–03. The most common cause of major injury to employees was slipping and tripping, accounting for 37 per cent of cases reported. Being struck by a moving or falling object accounted for 14 per cent of injuries, falling from a height for 14 per cent, and handling, lifting or carrying objects for 12 per cent.

From this information are we able to come to any conclusions about safety in the working environment? If the answer is YES, then it would be prudent to attempt to exploit and provide safe operating modules for systems to reduce injuries and life threatening situations.

Professor G. S. Virk, University of Leeds, UK

A primary barrier for making robotic devices more widely acceptable is that no real standards exist other than the Machine Directive for the robotic systems that are needed to meet the needs of the services sector. These standards are designed to separate the machines from the humans because of the potential danger that they pose to human life. Safety is clearly very important but the blanket requirement that robots need to be 100% safe is unrealisable. We are in the difficult situation where robotic systems are not seen to be reliable nor safe enough for widespread adoption. No standards exist for manufacturers to use in their new products and developments occur

in a totally ad hoc manner. New systems are being developed first and then the question of what needs to happen if they are to be commercially exploited. Various groups are being asked to approve the systems but there is no real methodology or organised network to control the developments. We must have standards that allow (potentially) powerful systems to operate and occupy the safe spaces we live in whilst providing the services that are needed. What level of safety is needed to get this process underway? This is a difficult question but a consensus is needed within the community. International H&S bodies need to be tasked with this and they must work with the various stakeholders to produce the standards for safety that are acceptable to society and how these can be satisfied in a new and emerging area of technology. Coupled with the safety issue must be legal aspects and the responsibilities of the various people involved in the loop. Users must have a responsibility in the safe use of the devices, i.e. they must take appropriate training and ensure protection to other people in the vicinity of the operating system.

Dr Nicola Amati – Politechnico di Torino, Italy

The enormous success of Sony AIBO robot dog (more than 30,000 robots sold) demonstrates that the field of entertainment can be considered at present to be a most profitable area for exploitation where an increase in mobile robots may occur. It is reasonable to assume that domestic applications (instead of industrial) are potentially a safe source for profits. The current development of humanoid robot technology is not ready yet to perform industrial tasks. Simple interaction with children is what technology can offer for a reasonable cost today. Inter-reaction with a machine to perform accurate predefined tasks with high reliability is something which it not ready now. The biped robot Sony SDR-4X, has been programmed to interact and play with humans. This has gained more success than the biped Honda Asimo, which is operated by a real worker. The likely key to success for mobile robots is through the use of toys and could be the major attraction of the general public (possibly in large numbers). Several web sites are available on the internet where one can buy, assemble and program models of mobile robots. The market in this case is aimed at robotics enthusiasts. This means that we are producing in this instance, a niche market. Toys are the market of today and will be expanded further as one of the future ones for mobile robots. This is an example of the expansion of existing markets, so what is the market of the future for mobile robots?

Perhaps in nuclear applications, underwater exploration, high rise building's inspection, petrochemical plant operations, anti-terrorist devices (searching for bombs), space applications. Maybe for service robot for hotel assistance (transport of luggage to the rooms), personal robot for domestic fitness exercises, personal robot for home surveillance and personal robot for support in the garden or for other activities in agriculture.

Dr J Dai – Kings College, London UK

Academic partners – exploitation of the results of the network for academic partners will be in the use of the underlying design, robotic and modular techniques in further research projects and in case study ideas for teaching. The use of modularity and robotics techniques will enhance the research of academic partners in our field by providing a good industrial application in assembly and excavation and by solving a practical problem. This fits well with some universities' projects on stiffness modelling and on manipulation planning, benefits the research particularly in our current study of excavation, packaging and robotic manipulation. Another research area on re-configurable manufacturing can be greatly enhanced. Such technology enhancements will benefit the use of the techniques in future applications and projects, which are likely to relate also to methods of robotic climbing. On the teaching side, universities run undergraduate and postgraduate course units in the areas of design, mechatronics and manufacturing techniques and the results of the network are likely to provide examples of the application of these techniques and case studies for students. Industrial Partners – the basic aim of our project is to investigate techniques suitable to promote modularity and flexibility. This will be of interest to companies that have to design out problems in virtual prototyping and rapid prototyping. For a dedicated packaging system it is unlikely to be commercially worthwhile unless one offers flexibility (albeit at some loss of speed). The system then becomes more attractive to prospective purchasers/users. The flexibility with these types of assembly lines means that it is possible to reconfigure the system between product changes. This aspect can be exploited by providing a means whereby end-users can perform the reconfiguration for themselves (e.g. via software). An alternative is to provide a service for end-users in which the specification for reconfiguration is provided based on details of the packaging design.

Dr M Ribero –Instituto de Soldadura e Qualidade, Lisbon, Portugal

In Europe, issued directives are from initiation to the developed standardised object. There are many, but for the robotic industry it seems that the most important ones could be the Machine Directive and the Low Voltage Directive. Other directives could be applied according to the scope of application, for example in the case of applications for hazardous explosive atmospheres where Directive ATEX 94/9/EC applies. For toys there is the 88/378/EEC Directive. Of course health and safety are of high importance and are also covered by the Directive 89/655/CEE. Another example is the Directive for Active implant medical devices, 90/385/EEC.

Standards are important for the economic participants because they:

- increase the rationalization of production by mastering technical characteristics, validating manufacturing methods, increasing productivity and giving operators and installation technicians a feeling of security whilst satisfying the customers;

- by standardization, the existence of reference systems will enable us to make better assessments and reduce uncertainties, to aid in the definition of user needs, will help to optimize the supply chain with minimal additional testing and will clarify business transactions;
- allow the easier transfer of knowledge and innovation whilst participating in the development process of a standard. This will help the innovation and product development through cooperative technology pull-through;
- facilitates and accelerates the transferal of technologies in fields which are essential for both companies and individuals e.g. new materials, information systems, biotechnology, electronics, computer-integrated manufacturing;

Easy maintenance operations and simple component replacements that are widely available is very important as a general requirement for any tool in industry. Interchangeable components or integral parts of a system allowing the use of different modules for different tasks with the same main system assumes a high level of acceptance by industry. Manufacturers generally dislike standardisation as this reduces competitiveness and they are likely to exclude the choice of competitive tools. Customers will be the first to defend standardisation as it saves them money and will result in technical superiority. Reliability is another very important equipment aspect required by industry. Not only regarding the wear and potential damage, but also the capacity to easily recover the situation for optimal working conditions when a problem affects the normal or optimal performance of the tool or system. From the industrial point of view as a user of robotised systems, the more important standards cover 5 main areas, mechanical; electrical/electronic; safety; economical; control & computing.

Mr Y Measson, CEA, Paris, France

In the nuclear and military (army) fields, H&S are the most important issues to take into account when designing a mobile platform. Two applications we are involved with are Centaure, a platform capable of opening doors, and providing a video feedback of the installation in nuclear facilities, where humans can not enter, or when there is a risk where it is not safe for humans to operate and Dactarix, a robotic platform designed to manipulate bull noses used outdoors on firing ranges.

Safety remains the most important design criteria for these applications. Industrial production has to maintain this high safety level. Another robotic platform, Eros with the same safety level as Centaure is used in nuclear environments and is manufactured by Cybernetix. Liability – in respect of neglect are the responsibility of the operator in most of the cases. For malfunction of operation, the manufacturer is responsible.

Risk considerations for market areas are:

- Domestic market: the main risk is in producing unreliable or inappropriate robots. Domestic applications do require safety but not safety critical in terms of degraded modes. For the robot itself, low power motors ensure an

almost "unbreakable" robot where the main risk is in retaining the ease of use of the robot and its reliability.
- Medical applications: the safety and reliability aspects here are extremely important. The risks are not the same for a home help or health monitoring robot as for an exoskeleton type. For the first two applications, an extremely reliable robot is required and safety aspects can be provided.
- The universal mobile assistant robots and domestic area ones are very close. The risk here is that they are too similar.

Dr F Simons, Fraunhofer, IPA, Stuttgart, Germany

Technological risks – robotic floor cleaning devices already exist on the market (Elektrolux, iRobot, Kärcher). The market demand for these devices has been proven by the (unexpected) high sales figures in the last year. Thus for devices at the current level of functionality there is no risk from the technological point of view. However there are technological challenges for improving these devices in terms of, energy supply, navigation based on low-cost sensors (partially to developed), flexibility of cleaning systems covering both flat floors (parquet, flagstones, etc.) and carpets, improving cleaning speed and cleaning capacity. The successful development of technologies is connected with a corresponding risk related to the customer's behaviour. The prognosis of the UNECE and the IFR has shown that there is a growing market with high potentials for selling floor-cleaning robots. Thus the risk in engaging in this market seems to be quite low. Potentials for optimisation are increasing with cleaning speed and cleaning quality improvements whilst the product price is simultaneously decreasing. Customer acceptance is primarily dependent on the cleaning quality of the robotic device. Secondly the customer confidence of the wall climbing systems depends on the safety of the climbing features. Other aspects of customer confidence and concern are the design and man-machine-interfaces.

In the short term, existing markets are in the domestic cleaning tools area, like manual vacuum cleaners, dry and wet cleaning of clothes and ancillaries such as cleaning fluids, swabs etc. The sales channels established in this market can directly be used in the distribution chain for future mobile and autonomous cleaning devices. In the medium to long term, improving speeds of operation and cleaning quality of the robot cleaning devices to a semi-professional level is likely.

R Walker BSc, The Shadow Robot Company, London, UK

There are market areas to be addressed in – disability, healthcare – to be considered are entry barriers, volume, emergencies are these limited and small markets? In creating new markets we must consider production costs, customer needs and whether there is a viable technology available or within a defined time scale. The barriers to entry will be the cost of devices, their limited function and over-selling. The hardware will always need to go through an approval process, an optimisation process, a legal process, and others.

In any volume product, specialisation will prevail. Re-use of hardware will happen, but at a low level. Here, modularisation will start to play a huge part in exploitation success. Software is usually the stumbling block. Software development is much more expensive than hardware to produce, to test, to verify and to understand. If we can re-use software, we gain substantial leverage for the success of the product in terms of cost, acceptance and system interfacing. Each robot application will still generate it's own code base.

H&S issues for patient handling at present is performed with equipment that is highly regulated, in terms of quality of construction, regular inspection, training of users, and so on. People who have to lift a patient are specially trained to do so but there is still a high accident injury rate. Also safe lifting weight levels are gradually coming down. For the independently minded disabled person, the ability to get up from a fall is critical to them to being able to live alone. Hence if it is possible for a robot system to be designed for home use to lift the user off the ground and then place them back into the chair or bed, then it can make a major improvement to their quality-of-life. Liability and the current situation with respect to software liability is that the manufacturer is able to disclaim all liability for the software. Clearly this will tie the market down, hence there must be a method permitting people who want safe systems to use well-tested software, and those who want experimental systems to have an alternative option. The Debian stable/testing/unstable model fits well to this.

Mr K Herman, Caterpillar, Belgium SA

Manufacturing in new and existing markets – currently, the use of mobile robots in manufacturing is quite limited and applications involve the movement of materials rather than the manufacturing processes. Industrial robot and automation providers tend to focus their efforts on the automation industry due to the potential possible large volume of sales. Unfortunately for the mobile robotics industry, automobile manufactures are satisfied with hard automation because of the large volumes of the same products that they produce. The key to mobile robots breaking into the manufacturing industry will be a quantum leap in sensor technology. If these robots were better able to "find" and "see" in their environment, they would have the ability to carry out more accurate operations safely and make important decisions that they are not able to make today. This will have a direct impact on the quality and robustness of robotic operations. Today's manufacturers are striving for 6-sigma levels of quality and the current state of mobile robotics will not produce this level. Targetting the manufacturer is very important in the mobile robotic low to medium volume product sector who have a greater need for flexibility in their operations and are therefore more willing to spend time and money to reconfigure their equipment to meet the current needs of production. Modularity will play a key roll in economic success in this type of market.

One area that has not had much attention from the robot industry is farming. The farming industry is moving from the small family farm (with low capital resources) to larger scale industrial farms (more capital and incentive to reduce labour costs). There are many applications where robotics could be applied such as:- milking cows, (information on robotic milking can be found at http://www.roboticdairy.com - feeding livestock where the storage of animal feed and the eating area for the animals are often structured environments – cleaning manure from animal lots as a necessary and repetitive operation that occurs in a relatively structured environment. The cultivation of fields offers opportunities in wide-open low risk areas where today's GPS technology allows automatic vehicle tracking. Authors note: US wheat combine harvesting is already operating like this in some areas.

Dr J Albiez – FZI, Karlsruhe, Germany

Health & Safety issues – The topmost guideline for implementing CLAWAR recommendations for service type robots interacting with humans must be "Never ever harm a human". Robots that operate in a human interactive environment must have enough security protection software and hardware to prohibit them from harming humans. Consider this in more detail and you will quickly come to the conclusion that the robot will not be able to do anything because every action could lead to human injury when in close proximity to the operating robot. To have a safe robot that can execute a task can be categorised in 3 ways:- Environments without humans – robots working in such an environment (e.g. Pipelines, sewers and industrial plants) can not directly harm humans. Such a robot will have to conform to safety requirements of the specific working environment. Environments with some trained humans – such an environment could be an industrial plant. Humans working in this environment know how the robot behaves, and how its safety systems work. Therefore the robot doesn't need to have special safety measures. Environments with untrained humans. This type of robot must have enough safety measures incorporated such that a human being interacting with it does not need to know of the specific functions of the system and thereby will not be harmed. A typical environment and application would be a service robot operating in a museum or in a home.

Two examples from FZi are: a) – FZI is working on a project to develop an autonomous sewer inspection robot called Makro. The idea of this system is to have a multi-segmented, snake like sealed robot system that "lives" in the sewer, inspects it and regularly reports back damage, illegal chemicals etc. affecting operators. b) – FZI are developing the Telelift controls for an autonomous hospital goods transportation system to operate by staff in the cellars. There is a safety chain to link all safety devices together and a vehicle power cut-off bumper sensing from obstacle detectors. As hospital staff know about robot behaviours and use them regularly, many risks can be accepted.

Professor. K Berns, University of Kaiserslautern

Mobile manufacturing robots – future production is focusing on a very flexible manufacturing process, in which mobile manufacturing units or manipulators can be arbitrarily grouped together. It must also be considered that humans should be able to work directly together with machines. At the moment in Germany it is not permissible for a human worker to move inside the workspace area of an operating industrial robot arm. To maintain this condition most manufacturing cells are bounded by a safety fence. When a human moves inside the cell area, the manipulator automatically stops. Integrated work between robot and human worker is not allowed. A first step towards the acceptance of this type of operation, is a work cell without any fence in which a human can operate around a robot with 100% safety. Currently the University of Kaiserslautern is working on this problem together with a large established robot manufacturing company. A first solution could be that additional sensors are installed directly onto a robot (mobile platform and manipulator) and in the workspace area of the robot. E.g. several ultrasonic sensors or laser distance sensors are fixed on the robot. The human – robot workspace critical areas are calculated based on the manipulation task and the distance between the robot and the human. These areas can be put into 3 classes, absolutely safe, low risk for the safety of the human worker and high risk where the robot motion must be stopped directly. Considering these concepts, even integrated sensor systems will still find 100% safe solutions problematic. E.g. one can use a Laser scanner to define safety areas and stop objects immediately. The problem associated with this requirement is that several sensors must be fixed on the system in order to observe the whole workspace of the robot. Currently (January 2004) the University of Kaiserslautern together with other German research institutes has begun research based on the use of ultrasonic sensors with special modelled frequencies placed on the robot in such a way that overlaps of the areas will provide 100% coverage.

Dr. Manuel Armada, IAI-CSIC, Spain

Risk, New Markets, Old Markets and Exploitation I have been conducting an exercise to try to understand the barriers to exploitation in which CLAWAR partners are involved, why these barriers exists and how hard they are for manufacturers in industrial production and the subsequent marketing of the **advanced robotic devices.** I have done this by using my personal academic and engineering experience over the last 25 years whilst I was following closely the evolution of applied robotic research activities in Spain, Latin America and the EC (from my participation on many mixed committees between academia and industry). From my experiences concerning situations where there has been significant activity in many sectors, there have actually been very few projects and real ideas that were finally fully exploited. Of course, it is difficult to assess this with high accuracy, however, according

to my records, I have prepared the following table where I summarise my findings:

Sector	Activity	Trends
Nuclear	Inspection, maintenance	⇔⇑
Space	Planet exploration, in-orbit manipulation	⇔⇑
Maritime industry	Shipbuilding, repairing, inspection, cleaning	⇔⇑
Underwater	Inspection	⇑
Agriculture	Tractor automation, greenhouses,	⇑
Tunnelling, mining	Instrumentation, control	⇔⇑
Medical and healthcare	Surgical, robotised wheelchairs, rehabilitation	⇔⇑
Civil engineering and construction	Inspection, assisted manipulation	⇔⇑
Services	Building maintenance (cleaning, inspection)	⇑
Domestic applications and personal robotics	House keeping, robotised house	⇑ ⇗
Humanoid	Humanoid design, control, new actuators	⇗
Edutainment	Guiding in museums	⇑⇓⇗
Humanitarian demining	APL detection using walking robots	⇗

⇔: long tradition, steady exploitation (>15 years) ⇑ : increasing exploitation ⇓ : decreasing exploitation ⇑⇓ : oscillating exploitation ⇗ : future exploitation
Trends from Dr Armada's personal experiences during the last 25 years

Professor. G Muscato, University of Catania

Guidelines and standards will be important factors in many robotic sectors to stimulate new research and subsequent investments. As regards safety, guidelines should be general to cover safety of the operator with sufficient specific details for each different application sector. In this way, manufacturers will be also encouraged in the development of new products, without taking too much care about safety issues, they will just follow the guidelines. Guidelines should not be very detailed, but will clarify the responsibility of the manufacturer and that of the operator for the use of the equipment. The operator will need to know what he can do and what he cannot do with the robotic product and how the environment should be prepared for the given operations. Knives, guns and cars are sold to everyone and I think that a robot will not be more dangerous over any of the three items given in the example. It will be very important to instruct people on the right way to operate with them and on the possible dangers in their use. Standards on technical issues will be essential if there is a future expansion in robotic applications. It is not clear if the definition of standards will cause the expansion or if the beginning of the expansion will cause the birth of new robotic standards (as has occurred in many other sectors).

Industrial manufacturing: new industrial robots co-operating with humans will embody a good level of exploitation of an existing market for remote applications. New technologies and sensors allow the interaction of robots and human thus opening new opportunities from the old well established markets. Industrial maintenance: new robotic autonomous tools or semi-autonomous ones will be adopted for industrial maintenance. This will include pipe inspection,

tank inspection, wall inspection etc. Domestic maintenance: this will mainly be in the application area of cleaning inside hotels, factories, offices and personal habitats such as houses. Automotive – automation: here we have the opportunity to create much safer vehicles through autonomy. Reference [3] CLAWAR2003 conference paper on autonomous levels of vehicle control. Agriculture: in all the agricultural sectors will require more automation when the global market needs to compete with low cost manpower of the worlds developing countries. Our work in CLAWAR has focussed on hazardous applications; this included CLAWAR machines (Robovolc) for volcanic inspections. The H&S issues have demanded "man-in-the-loop" stability. Hence the mode of operation has been tele-operation although we have managed to include a few autonomous behaviours in some of our machines. Ultimately the humans have been in charge and full autonomy has been unnecessary. The general view is that the robotic system will definitely fail at some point in time and we must ensure that we can handle the failure. Robot risk assessment – Suggested readings from Prof G Muscato. There is a *Robot risk assessment tool version2* where one can customise applications. This is applied for end users and manufacturers in suggested fields such as robotics, automotive, construction, military, medical, construction, aviation, and others. Information on operations, tutorials, and upgrades can be found at www.robotics.org or www.designsafe.com. Further information about safety through design can be reviewed from the Institute for Safety through Design, www.nsc.org.

Year 2 Technical Workshop Title: Economic Prospects for Mobile Robots, Exploitation & Risk Assessment. This workshop was arranged and prepared by Bill Warren and Neil Heyes, QinetiQ. The workshop was held at FZI on 26/11/03 as part of the continuing CLAWAR 2 network where over 30 members attended. There was a series of presentations and a roundtable discussion followed the CLAWAR network series of meetings. Topics addressed exploitation and risk assessment issues for future mobile robotic systems and what decisions would affect factors to influence users and manufacturers on decisions for viable solutions for volume production and after sales service.

The Workshop Presentations: Bill Warren, QinetiQ, UK gave an introduction to risk & exploitation mechanisms for new robotic products, a resume of year 1's findings and presented the year 2 task requirements. This task is studying exploitation, health, safety and risk assessment of the markets using new data and data identified in year 1. This involves gathering information on the current status of mobile robotics and identifying areas where rules and regulations are required for safe operation. We are encouraging the dialogue between different bodies/organisations and assisting in the development of standards that are acceptable from all stakeholders' viewpoints. Finally, we aim to remove existing concerns on the use of mobile robots and are trying to speed up the widespread acceptance of robotic systems and time to market.

Martin Haegele, IFR & Fraunhofer IPA, Germany gave an excellent overview of many mobile robotic applications in the service industry. Martin

explained how world markets influenced the exploitation of robotics in the service sector. The World Robotics 2003 analysis published and co-authored jointly by the International Federation of Robotics and the United Nations economic commission for Europe have seen that world demands fell in 2001 & 2002 by 21% & 12% respectively. The first half of 2003 has seen a positive upturn in the markets. Time will tell if this is a continuing trend.

An update of these facts and predictions can be obtained from the World 2003 Robotics book on statistics, market analysis, forecasts, case studies and profitability of robot investment. [1].

Martin also showed some very interesting and technically inspiring videos of safety features where man and machine manipulation robotics systems operate together. 100% safety of operation was demonstrated and is essential for volume sales of products. The use of virtual and hardware back-up controls featured highly in a specific application where operators work with machines in an industrial assembly plant.

Francesco Cappello, Media IRC, Sicily, from one of the European Innovation Relay Centres (Catania) gave an interesting presentation on how links are provided and maintained between academia/industry. The presentation demonstrated how IPR exploitation is advanced (especially for SMEs) in technology sectors including our very own robotics industry. Francesco explained how the IRC centres operate and what benefits are available to aid technology exploitation to produce new and innovative products. The IRC network was set up by the EU in 1995 to support transitional technology co-operation in Europe with a range of specialist business support services having almost 250 offices to form an integrated European transitional technology network.

IRC's procedures involve setting up a 3 step plan for: Technology watch and implementation planning, follow-up research by searching for partners through technology offers, brokerage events and company missions (get-togethers), discussing and signing deals with confidentiality agreements, team negotiations and IPR and linking innovation funding.

For further links to IRC opportunities, please contact:

Dr Franceso Cappello, email: fcappello@mediainnovation.it or have a look at the IRC public website on http://irc.cordis.lu/

Kaspar Althoefer, Kings College London, UK

As an academic, Dr. Althoefer presented his and his colleagues viewpoints on preparing research bids and the implications for future production. The topics looked at the main difficulties concerning commercial exploitation in the academic environment, the advantages of industrial collaboration, obtaining funding from UK Universities, getting governmental support and conclusions of where this exploitation is leading in the robotics industry. Exploitation hindrance due to lack of contact, market knowledge and support is high, IPR agreement issues make development difficult and many innovations get no further than the research bed, hence "Blue sky to Product" can be a very tortuous route. The advantages of involving an industrial partner and the types of agreements were discussed, e.g. Licences, royalty payments, IPRs and

collaborative research negotiations. UK exploitation schemes were presented such as, research councils (EPSRC), department of trade and industry (DTI), knowledge transfer partnerships (TTI), EU schemes (CRAFT & STREP), company secondments and the use of knowledge consults.

Workshop Roundtable Discussion

The discussion for WP6 was held at the CLAWAR network meeting on the 28th November 2003.

The main topics discussed were:-

Exploitation – There are several areas affecting this – cost, type of product, belief in the innovation, value of prestige, possible market size – e.g. is there a gap and what can the market sustain? Questions were asked as to why certain products are successful and others are not. There was an important discussion about costs coming down as market size goes up for robot and manufacturers to be successful. It has been suggested that these two factors need to meet or at least merge.

Risk – What types are there? Definitions are: financial, for user and financier, product, problems when robots are interacting with humans. Safety, health and reliability are all big issues and the resultant liability cases. *Safety* – there is a high priority need to address safe operation to ensure the success of products as these are essential to our industry. Risk concept templates are needed and categorised as High/Medium/Low Risk. How can we reliably assess risk when there is no tried and tested market? Reliability is all about property, 3rd party safety and what might happen to the machines. Operational safety is a therefore an interactive essential.

Technology – Large use of current technologies is needed for latest design opportunities whilst introducing new innovations as they come to Market. High reliability is essential for acceptance of materials especially those approaching nano-level size. There is a need to transfer both military and commercial technology to maximise the success equation. Agreement at the workshop was reached that this pull-through should be maximised from basic research to major levels of production.

Rules & regulations – Due to limitations and cost implications there should be a focus on specific set of standards & guidelines linking WP6 with WP2. These should include liability, traceable quality procedures, interchangeability standards for modules e.g. modularity of blocks, components, connectors etc. Certification of products will become an output from these procedures.

Dialogue – An increase in dialogue between academia and industry is much needed and is starting to be fulfilled, e.g. the IRC brokerage events. Demonstrations of products are essential for users and to lure financial backing – a very sound business case is required for venture capital or bank financing. The CLAWAR survey, was discussed and agreement was reached that we need to extend the survey from CLAWAR members to the newsletter technical associates. Ref: www.clawar.com

Abridged Conclusions and Recommendations by the Members

Following the member inputs and workshop discussions, here is a summary of the pertinent points that were decided.

Conclusions Summary

Rules and Guidelines (Standards).

- Decide on standards which are guidelines and not mandatory.
- Do we recommend any directives – e.g. Machine Directive and the Low Voltage Directive.
- Safety is clearly very important but the blanket requirement that robots need to be 100% safe is unrealisable, hence loose guidelines to clarify the responsibility of the manufacturer and operator.
- Get designers, manufacturers, equipment sellers and users involved with all stages and at module level.
- Secure verification for the user – levels of ID before a command is given.

Exploitation

- The main fields for exploitation are service robots used in human centred environments.
- Increased automation in industrial plant inspection involving hazards to humans.
- Safe domestic applications (instead of industrial) are potentially a safe source for profits.
- Lists are required e.g. nuclear industry, underwater exploration, building's inspection, petrochemical etc.
- Simple interaction toys for children. What can technology offer for a reasonable cost?
- Farming is an area for exploitation – milking, feeding livestock, cleaning and crop cultivation.
- Exploitation of the results of academic partner networks is desirable.
- The use of modularity and robotics techniques will enhance research flexibility.
- Local reconfiguration by software, by electronics or remotely via communication links (web/radio).

Health & Safety issues

- Safety remains the most important design criteria for most applications.
- Safety of the human element, the equipment and the operational consequences must be considered.
- H&S issues have demanded "man-in-the-loop" stability for specific operations.

- The use of robotics to inspect areas inaccessible to humans will result in new safety case requirements.
- To ensure safety, we need reliability and the capacity to repeat precisely the same task, operation or movement.
- Industrial production has to maintain a high safety level and deal with any hazardous or dangerous materials.
- There should be a method of ensuring that well-tested and validated software is incorporated in systems.

Old Markets

- A problem is that traditional players have established market positions with no desire to change.
- Revitalisation in manufacturing where hazardous conditions prevail is necessary.
  Edutainment markets are likely to increase, machines are small and do not pose an H&S problem and many EU children own or use a computer thus providing huge opportunities for controlling mobile robotics.
- An example, floor-cleaning systems require technological challenges for design improvements.
- CLAWAR concludes that there is a huge global market "out there" and we are but at the beginning of a lengthy and healthy road to robotic usage and prosperity in both the fixed and mobile robotic sectors.

Risk and New Markets

- We must consider production costs, customer's needs, technologies and time scales.
- The barriers to entry will be the cost of devices, their limited function and perhaps over-selling.
- To reduce the risk in new markets, it is important to bring reality to the general public especially in the home.
- Mass demonstration events and less hype is needed.
- Market sizes vary, so there is a huge need for open modularity to bring in many innovative players.
- Modularity reduces installation and component costs as they are reused many times.
- Some technology is available but there needs to be focused research on HMI psychology.
- The risks with domestic markets is in producing unreliable, unsafe or unacceptable robots for the public.
- *Edutainment* robotics is a growing market and is lower risk as it is fairly safe and technically available.
- With maintenance systems such as w*all cleaning systems,* customer acceptance primarily is a function of quality and confidence of the climbing safety and the robot's cleaning ability

Recommendations Summary

Rules and Guidelines (Standards).

- An industry-wide standard is needed that is internationally recognised and developed by consensus among trading partners to serve as the authoritative language for mobile robotics.
- Robotics standardization must provide defining terminology, modularity and accumulating databases of quantitative information to develop the global standards.
- Standards should cover at least the 5 main areas of, Mechanical, Electrical/Electronic, Safety, Economical and Control/Computing.
- H&S bodies must work with the stakeholders to produce society safety acceptable standards.
- Appropriate governing bodies must make users responsible for appropriate training and ensuring the protection of others whilst in the vicinity of the operating system footprint.
- Robots must have an audit trail facility that is separate from the operational system of the robot.
- It is necessary to ensure that the burden of certification is not so high as to stifle innovation.

Exploiting

- We must address the likely key to success for mobile robots (through toys) as a major attraction to the general public. Expand today's toy further as a future area for mobile robots.
- Define through market research how user needs will produce successful products.
- Produce a list of exploitable products. The list could be extensive, service robots for hotel assistance, personal robots for domestic fitness exercises, for home surveillance, for agricultural support etc. (Year 3 work).
- Continue network organizations to promote the cause for mobile robotics (if we believe this is so).

Health & Safety issues

- The topmost guideline design for implementing CLAWAR recommendations for service type robots interacting with humans must be **"Never ever harm a human"** – so education in the media is essential.
- Develop safe operating modules to eliminate injuries and life threatening situations – using specific information from accident and death statistics.
- Because of the nature of the intended uses, the robot must be able to be managed safely by a variety of operator types.
- Define how ultimately humans are in charge and where autonomy levels should be controlled.
- Design the systems to "fail in a safe mode".

- Concept design should consider important points for H&S criteria for service robots, entertainment and toys.
- When objects are moving inside defined safe areas, when in a danger situation sensors must detect and immediately stop equipment movement.

Old Markets

- Adopt an open modular approach to make designs cheaper and quicker to produce. Hazardous applications could be a major growth area.
- H&S issues must be overcome for successful expansion of healthcare and medical device developments.
- The sales channels established in existing markets should be directly used in the distribution chain for future mobile robotic markets.

Risk and New Markets

- Use the work of CLAWAR to encourage open modularity to allow organisations to "make the wires open and the boxes closed" thus keeping IPR secure in the module.
- Focus initially by addressing the primary barriers such as minimising risk to human life.
- Use low power motors to ensure "unbreakable" risk reduced robots and design electronics and mechanics to provide ultra-safe operation for users – e.g, safe control of speed and torque.
- For medical applications including disability and healthcare we must consider entry barriers, volumes and emergency categories.
- In creating new markets we must consider production costs, customer needs and in what time scale viable technologies become available.
- There must be a new market approval process for software, middle-ware and hardware, including re-use of modules.
- The sale of a service robot should come with a service agreement or an acceptable warranty guarantee period including the contact number of the manufacture's hotline.

Acknowledgements

Many thanks are due to the members for their contributions, to the workshop presenters and to the host organisers of the venues.

References

1. International Federation of Robotics World 2003 robotics statistical report and analysis, United Nations publication, ISBN 92-1-101059-4, ISSN 100-1076
2. Health and Safety Executive Statistics of Fatal Injuries 2002/03 report, web site http://www.hse.gov.uk
3. CLAWAR2003 conference paper, Warren/Fig. 4, Warren's steps to autonomous levels of vehicle control. ISBN 1-86058-409-8

Index

Abbate N 503–509
Aguayo E 869–878
Akinfiev T 519–527, 625–631, 677–688
Akira E 163–170
Al-Kharusi S 495–501
Alba D 701–712
Albarral L J 135–141
Albiez J 55–69, 349–364
Alexander R 47–54
Allen A R 801–811
Almeida de T A 419–426, 989–996, 1097–1105
Amar F B 825–833, 879–888
Amati N 329–340
Amato P 1041–1051
Arbulú M 633–641, 643–653
Armada M 219–227, 229–236, 467–476, 519–527, 625–631, 677–699, 701–712, 943–952, 1147–1158
Asada M 757–764
Asama H 113–121
Azzi D 577–584

Böhme T 1033–1040
Babkirov S 1041–1051
Bade R 971–979
Bag K S 801–811
Bagheri S 919–934, 1071–1080
Bajbouj M 399–409
Balaguer C 569–575, 633–641, 643–653, 655–663
Baselga S 437–445, 447–456
Basile A 503–509
Becerra M 367–375
Berner N 879–888
Berns K C 357–364, 899–908, 981–988
Besseron G 825–833
Bidaud Ph. 825–833, 879–888, 1127–1135
Bock G H 725–735
Boemo E 869–878
Bona B 329–340
Bradshaw S 801–811
Braun T 981–988
Brockmann W 935–942
Bruneau O 543–559

Caballero R 689–699
Cabas L 633–641, 643–653
Carabelli S 329–340
Cardona S 295–303
Ceccarelli M 561–568
Celaya E 135–141
Chapple H P 801–811
Chellali R 377–385
Chiaberge M 329–340
Chochlidakis I 265–274, 285–294
Chocron O 879–888
Christaller T 835–842
Chugo D 113–121
Clark E J 859–867, 961–970
Cobano A J 1119–1126
Cruse H 143–151
Cruz C A 1019–1031
Cutkosky R M 859–867, 961–970

David A 551–559
Deutscher M 399–409

Dillmann R 55–69, 349–356, 953–960
Dutkiewicz P 237–245, 427–435

Elkmann N 1033–1040
Estremera J 1119–1126

Farkas F 835–842
Feliu V 249–264
Felsch T 1033–1040
Fernandez R 519–527, 677–688
Figliolini G 889–896
Fischer J 97–102
Fleischer C 153–161
Fontaine J 377–385, 543–559
Fujiki N 305–312

Gaßmann B 55–69, 953–960
Garcia E 457–465, 1083–1090
Gatsoulis Y 265–274, 285–294
Geisel T 765–773
Geisler B 961–970
Genta G 329–340
Gerth W 601–609
Gharooni C S 171–179, 487–493, 801–811
Golubev F Yu. 909–917
González A 249–264
Gonzalez de Santos P 457–465, 511–518, 1083–1090, 1119–1126
Gonzalez F 275–284
Gonzalez-Gomez J 869–878
Gradetsky V 1041–1051
Grand Ch. 825–833
Guardabrazo A T 511–518
Guastella C 503–509
Gutiérrez D 633–641

Haß J 765–773
Ham V 713–722
Han W M 201–209
Harvey D 577–584
Harvey R I 813–823
Heikkilä M 1091–1096
Heng J 1053–1060
Herms A 971–979
Heyes J N 1171–1190
Hillenbrand C 899–908, 981–988
Hiller M 411–417
Hirokawa V 85–96
Hobbelen D G E 787–800
Hofschulte J 601–609
Holmes J P 585–592
Hosoda K 757–764
Howard D 495–501
Hsu Z 835–842
Hurst W J 123–133

Ichikawa A 479–486
Ihme T 971–979
Inoue K 305–312
Ishihara H 341–348

Jamil F 801–811

Kadar E E 1107–1117
Kaetsu H 113–121
Katz M 399–409
Kawabata K 113–121
Kerscher T 55–69
Khashan S T 387–397
Khraief N 611–623
Kiełczewski M 237–245, 427–435
Kim S 859–867
KM'Sirdi N 611–623
Knyazkov M 1041–1051
Kondak K 153–161
Kopacek P M 201–209
Korianov V V 909–917
Kowalczyk W 191–199
Kozlowski K 191–199
Krüger S 399–409
Krueger T 1033–1040
Kunst D 1033–1040

López-Coronado J 211–217
Larin B V 531–541
Lefeber D 665–676, 713–722
Licer O 611–623
Linder R C 313–320
Liu R 1053–1069
Lo Presti M 503–509
Longman W R 725–735
Longo D 1005–1015, 1159–1170
Lope de J 593–600
Lopes M A 73–83
Lucke M 1033–1040
Luksch T 981–988
Luo W Z 103–111

Lytridis C 1107–1117

M. Nogués M 295–303
Mösch F 935–942
Maaoui C 377–385
Macina G 503–509
Mae T 305–312
Manoonpong P 97–102
Maravall D 593–600
Marques L 419–426, 989–996
Martinez G 367–375
Martins A 989–996
Mayer M N 835–842
Maza M 437–445, 447–456
Mederic P 1127–1135
Mishima T 113–121
Miyake T 341–348
Molfino R 997–1003
Molina-Vilaplana J 211–217
Mombaur D K 725–735
Montes H 219–227, 229–236, 625–631, 943–952
Moral D E 163–170
Morales R 249–264
Morgado de Gois A J 411–417
Moronti M 997–1003
Muscato G 181–190, 1005–1015, 1159–1170

Nabulsi S 219–227, 229–236, 943–952
Nassiraei F A A 835–842
Naudet J 713–722

Odashima T 103–111
Ogino M 757–764
Okada T 85–96
Ortiz J 437–445, 447–456
Otatviano E 561–568

Padovani M 329–340
Palis F 321–328
Palivtseev A S 387–397
Paolo D E 813–823
Pardos M J 569–575
Pasemann F 97–102, 737–744
Pasqui V 1127–1135
Pavlovsky E V 387–397
Pedraza L 467–476, 625–631
Pedreño-Molina L J 211–217
Pintado P 249–264
Plumet F 825–833, 1127–1135
Ponticelli R 701–712
Prieto C 655–663
Prieto I 633–641, 655–663
Prieto M 1147–1158
Provancher R W 961–970
Putz B 201–209

Quinn D R 849–857

Ramirez A 457–465
Reinicke C 153–161
Reyes C 275–284
Ribeiro S M 1019–1031
Ripa V 889–896
Ritzmann E R 849–857
Rizzi A A 123–133
Rizzoto G G 1041–1051
Roca J 295–303
Rusin V 321–328

Sabourin C 543–550
Saki T 85–96
Salinas C 467–476
Sanguineti M 997–1003
Schlöder P J 725–735
Schmitz J 143–151
Schneider A 143–151, 321–328
Schumucker U 321–328
Schwab L A 745–756
Seebode M 601–609
Seipel E J 585–592, 843–848
Shibuya K 85–96
Silva F M 73–83
Solovtov V 1041–1051
Spampinato G 181–190
Spong W M 775–786
Sshierer E 201–209
Stürze T 1033–1040
Staroverov P 633–641
Suzuki M 479–486
Swain D I 801–811

Takuma T 757–764
Tavakoli M 919–934, 1071–1080
Tavoloeri C 561–568
Tenreiro A J 73–83
Tokhi O M 171–179, 487–493, 801–811

Torre de S 643–653
Tylor I R 801–811

Vanderborght B 665–676, 713–722
Vaughan D E 813–823
Velkenko M 1041–1051
Vermeulen J 665–676, 713–722
Verrelst B 665–676, 713–722
Virekoski P 1091–1096
Virk S G 171–179, 265–274, 285–294, 487–493, 577–584, 801–811, 1107–1117, 1139–1145
Volpe R 329–340
Vossoughi R G 919–934, 1071–1080

Wang W 1061–1069
Warren H A 1171–1190
Wei E T 849–857
Wettach J C 899–908
Winter D A 39–46
Wischmann S 737–744
Wisse M 745–756, 787–800
Würzl M 201–209

Xie M 3–38

Ylönen S 1091–1096

Zöllner M J 953–960
Zakerzadeh R M 919–934, 1071–1080
Zavgorodniy Y 321–328
Zhang H 1061–1069
Zhang J 1061–1069
Zong G 1053–1069
Zoppi M 997–1003