



springer tracts in advanced robotics 6

Raymond A. Jarvis
Alexander Zelinsky (Eds.)

Robotics Research

The Tenth International Symposium



Springer

Springer Tracts in Advanced Robotics

Volume 6

Editors: Bruno Siciliano · Oussama Khatib · Frans Groen

Springer

*Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo*

Engineering  **ONLINE LIBRARY**

<http://www.springer.de/engine/>

Raymond A. Jarvis · Alexander Zelinsky (Eds.)

Robotics Research

The Tenth International Symposium

With 305 Figures



Springer

Professor Bruno Siciliano, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy, email: siciliano@unina.it

Professor Oussama Khatib, Robotics Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305-9010, USA, email: khatib@cs.stanford.edu

Professor Frans Groen, Department of Computer Science, Universiteit van Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands, email: groen@science.uva.nl

STAR (Springer Tracts in Advanced Robotics) has been promoted under the auspices of EURON (European Robotics Research Network)

Editors

Prof. Raymond Austin Jarvis
Monash University
Dept. of Electrical and Computer Systems Engineering
Clayton, VIC 3800, Australia
ray.jarvis@eng.monash.edu.au

Prof. Alexander Zelinsky
The Australian National University
Dept. of Systems Engineering
Canberra, ACT 0200, Australia
alex.zelinsky@anu.edu.au

ISSN 1610-7438

ISBN 3-540-00550-1 Springer-Verlag Berlin Heidelberg New York

Cataloging-in-Publication Data applied for
Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <<http://dnb.ddb.de>>.

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 other ways, 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-Verlag. Violations are liable for prosecution act under German Copyright Law.

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science + Business Media GmbH

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2003
Printed in Germany

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: Digital data supplied by author. Data-conversion by PTP-Berlin e.k., Stefan Sossna
Cover-Design: design & production GmbH, Heidelberg

Printed on acid-free paper 62/3020Rw - 5 4 3 2 1 0

Editorial Advisory Board

EUROPE

Herman Bruyninckx, KU Leuven, Belgium

Raja Chatila, LAAS, France

Henrik Christensen, KTH, Sweden

Paolo Dario, Scuola Superiore Sant'Anna Pisa, Italy

Rüdiger Dillmann, Universität Karlsruhe, Germany

AMERICA

Ken Goldberg, UC Berkeley, USA

John Hollerbach, University of Utah, USA

Lydia Kavraki, Rice University, USA

Tim Salcudean, University of British Columbia, Canada

Sebastian Thrun, Carnegie Mellon University, USA

ASIA/OCEANIA

Peter Corke, CSIRO, Australia

Makoto Kaneko, Hiroshima University, Japan

Sukhan Lee, Samsung Advanced Institute of Technology, Korea

Yangsheng Xu, Chinese University of Hong Kong, PRC

Shin'ichi Yuta, Tsukuba University, Japan

Foreword

At the dawn of the new millennium, robotics is undergoing a major transformation in scope and dimension. From a largely dominant industrial focus, robotics is rapidly expanding into the challenges of unstructured environments. Interacting with, assisting, serving, and exploring with humans, the emerging robots will increasingly touch people and their lives.

The goal of this new series of Springer Tracts in Advanced Robotics is to bring, in a timely fashion, the latest advances and developments in robotics on the basis of their significance and quality. It is our hope that the greater dissemination of research developments will stimulate more exchanges and collaborations among the research community and contribute to further advancement of this rapidly growing field.

As one of robotics pioneering symposia, ISRR, the "International Symposium on Robotics Research," has established over the past two decades some of the field's most fundamental and lasting contributions. With the launching of STAR, this and other thematic symposia devoted to excellence in robotics find an important platform for closer links and extended reach within the research community.

The Tenth edition of "Robotics Research" edited by Raymond Jarvis and Alex Zelinsky offers in its 11-part volume a collection of a broad range of topics in robotics. The content of these contributions provides a wide coverage of the current state of robotics research: the advances and challenges in its theoretical foundation and technology basis, and the developments in its traditional and new areas of applications.

Remarkably, the focus of a sizable portion of this edition is on advances in robotic technologies and applications. The diversity, novelty, and span of the work unfolding in these areas reveal the field's increased maturity and expanded scope. The Tenth edition of ISRR culminates with this important reference on the current developments and new directions in the field of robotics - a true tribute to its contributors and organizers!

Stanford, January 2003

Oussama Khatib

Preface

The 10th International Symposium on Robotics Research (ISRR 2001) was held from November 9–12 2001, at Lorne, Victoria. The ISRR series of conferences began in 1983, and is sponsored by the International Foundation of Robotics Research (IFRR), an independent organization comprised of top researchers around the world. The goal of the ISRR is to bring together active, leading robotics researchers from academia, government, and industry, to define the state of the art of robotics and its future direction. Papers are generally more reflective and authoritative than those at other conferences, and over the years the ISRR has developed a high reputation. The symposium is typically held in a pleasant setting with a limited number of participants in order to maximize interaction.

This proceedings comprises 40 papers selected for ISRR 2001. The process of paper selection proceeded primarily through an open Call for Papers; these papers were reviewed by the Symposium co-chairs and the IFRR. In addition, the three regional delegations of the IFRR (North America, Europe, and Asia/Australia) invited a total of 18 papers.

IFRR (at the time of paper selection and hence the formal Program Committee for the Symposium)

Hirochika Inoue, <i>President</i>	Jan-Olof Eklundh
George Giralt, <i>Secretary</i>	Shigeo Hirose
Takeo Kanade, <i>Treasurer</i>	Gerd Hirzinger
Ruzena Bajcsy	Ray Jarvis
Robert Bolles	Dan Koditschek
Rod Brooks	Bernie Roth
Raja Chatila	Tomomasa Sato
Paolo Dario	Yoshiaki Shirai
Joris De Schutter	Tsuneo Yoshikawa

Papers were presented in a single track during the four day symposium. In addition, there were a number of information presentations by participants, a video session, and evening group discussions. A session on Field Robotics was organised by Chuck Thorpe and Hugh Durrant-Whyte. Alex Zelinsky organised a session on commercialisation of robotics research results. In keeping with the spirit of past ISRR's, a number of breaks and activities were scheduled to allow for greater participant interaction. These included walks, and a tour of the scenic areas around Lorne.

The topics can be loosely placed into 8 categories: (1) Dynamics and control; (2) Planning and modelling; (3) Sensing technologies; (4) Vision based robotics; (5) Mobile robot localisation and mapping; (6) Humanoid robotics; (7) Human-centred robots; and (8) Applications. They represent progress in traditional areas of robotics, in areas of more recent expansion or emphasis, and in more speculative directions

for robotics research and development. ISRR 2001 was an opportune time to reflect on the successes of robotics, on the expansion of topics, which are now encompassed by the field, and on the challenges for future commercial, technical, and intellectual success. The papers in this volume provide ample substance for this reflection.

The ISRR 2001 co-chairs/editors would like to thank Sarina Kennedy and Amber McKinley, secretarial staff at Monash University who greatly contributed to the smooth handling of research manuscript collection, transmission to reviewers and final assemblage for pre-print production, and Rosemary Shepherd, Pei Yean Lee and James Ashton of the Australian National University for helping to put the book together.

Canberra, Australia,
November 2002

*Raymond A. Jarvis
Alex Zelinsky*

Contents

Part 1. Simultaneous Localization And Mapping

Session Summary	3
<i>Raja Chatila</i>	
Towards Robust Data Association and Feature Modeling for Concurrent Mapping and Localization	7
<i>John J. Leonard, Paul M. Newman, Richard J. Rikoski, José Neira, Juan D. Tardós</i>	
1 Introduction	7
2 “Explore and return” using Laser	10
3 Sonar Perceptual Grouping Using the Hough Transform	12
4 Delayed Stochastic Mapping	13
5 Conclusion	19
A Real-time Algorithm for Acquiring Multi-Planar Volumetric Models with Mobile Robots	21
<i>Sebastian Thrun, Wolfram Burgard, Deepayan Chakrabarti, Rosemary Emery, Yufeng Liu, Christian Martin</i>	
1 Introduction	21
2 Multi-Surface Models	23
3 Acquisition of Compact 3D Models in Real-Time	25
4 Experimental Results	28
5 Discussion	28
Implementation of Simultaneous Navigation and Mapping in Large Out- door Environments	37
<i>Jose Guivant, Eduardo Nebot</i>	
1 Introduction	37
2 Simultaneous Localization and Mapping	38
3 Sub-Optimal Decorrelation Algorithm	40
4 Environment Description and Feature Detection	43
5 Experimental Results	46
6 Conclusions	46
A Bayesian Algorithm for Simultaneous Localisation and Map Building ..	49
<i>Hugh Durrant-Whyte, Somajyoti Majumder, Sebastian Thrun, Marc de Battista, Steve Scheding</i>	
1 Introduction	49
2 Bayesian Formulation of the SLAM problem	50
3 Solving the SLAM problem	52
4 Implementation of the SOG Method in Sub-Sea SLAM	56
5 Discussion and Conclusion	59

Part 2. Humanoid Robotics I

Session Summary	63
<i>Hirochika Inoue</i>	
ETL-Humanoid: A Research Vehicle for Open-Ended Action Imitation ..	67
<i>Yasuo Kuniyoshi, Gordon Cheng, Akihiko Nagakubo</i>	
1 Introduction	67
2 Essential Issues in Imitation	68
3 ETL-Humanoid: Our Research Vehicle	70
4 Multi-modal Perception and Action	76
5 Proto-Imitation Experiments	79
6 Summary and Conclusions	80
Low-level Autonomy of the Humanoid Robots H6 & H7	83
<i>Satoshi Kagami, Koichi Nishiwaki, James Kuffner, Kei Okada, Yasuo Kuniyoshi, Masayuki Inaba, Hirochika Inoue</i>	
1 Introduction	83
2 Low-level Autonomy of Humanoid Robots	84
3 Humanoid Robots H6 & H7	85
4 Autonomy in Bipedal Locomotion	87
5 Autonomy in Object Manipulation	91
6 Autonomy in Human Interaction	93
7 Conclusion	95
OpenHRP: Open Architecture Humanoid Robotics Platform	99
<i>Hirohisa Hirukawa, Fumio Kanehiro, Shuuji Kajita</i>	
1 Introduction	99
2 Overview of OpenHRP	100
3 Unification of the controllers	102
4 Determining the repulsive force	104
5 Sharing software between the simulator and the controllers	107
6 Simulations and experiments	108
7 Conclusions	111
Building Spined Muscle-Tendon Humanoid	113
<i>Masayuki Inaba, Ikuo Mizuuchi, Ryosuke Tajima, Tomoaki Yoshikai, Daisuke Sato, Koichi Nagashima, Hirochika Inoue</i>	
1 Introduction	113
2 Design of spined muscle-tendon humanoid “Kenta”	114
3 Design of the Software System Structure	121
4 First steps to manage the body	122
5 Summary and Concluding Remarks	126

Part 3. Dynamics and Control

Session Summary	131
<i>Bernard Roth</i>		
Stride Period Adaptation for a Biomimetic Running Hexapod	133
<i>Jonathan K. Karpick, Jorge G. Cham, Jonathan E. Clark, Mark R. Cutkosky</i>		
1 Introduction	133
2 Simplified Model for Open-Loop Locomotion and Adaptation	135
3 Stride Period Adaptation	141
4 Conclusions and Future Work	144
Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Using a Neural System Model		
<i>Hiroshi Kimura, Yasuhiro Fukuoka, Yoshiro Hada, Kunikatsu Takase</i>		
1 Introduction	147
2 Adaptive Dynamic Walking Using a Neural System Model	148
3 Planar Walking	152
4 Three-dimensional Walking	155
5 Conclusion	159
Design and Applications of Parallel Robots		
<i>Karol Miller</i>		
1 Introduction	161
2 Optimal Design of Parallel Robots	161
3 Medical Applications – MRI Compatible Parallel Robot	170
4 Conclusions and Discussion	172

Part 4. Humanoid Robotics II

Session Summary	177
<i>Tomomasa Sato</i>	
Development of an Interactive Humanoid Robot “Robovie” – An interdisciplinary approach	179
<i>Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, Takayuki Kanda</i>	
1 Introduction	179
2 Robovie: An Interactive Humanoid Robot	179
3 Two Cognitive Experiments	181
4 A Robot Architecture for Generating Episode Chains	186
5 Conclusion	189
The German Collaborative Research Centre on Humanoid Robots	193
<i>Rüdiger Dillmann</i>	
1 Introduction	193
2 The PbD Process	198
3 System Structure	200
4 PbD Phases	201
5 Experiments	203
6 Future Work	204
A New Generation of Compliance Controlled Manipulators with Human Arm Like Properties	207
<i>Ralf Koeppe, Alin Albu-Schäffer, Carsten Preusche, Günter Schreiber, Gerd Hirzinger</i>	
1 Introduction	207
2 The Human Arm System	208
3 The Advanced Robot: Design and Control Principles	210
4 Conclusion	217

Part 5. Human Centred Robotics

Session Summary	221
<i>James Trevelyan</i>	
Uniting Haptic Exploration and Display	225
<i>Allison M. Okamura</i>	
1 Introduction	225
2 The Correspondence Between Exploration and Display	227
3 Exploration, Modeling, and Display of Surface Features	229
4 Conclusions and Future Work	235
Human-Centred Robotics and Interactive Haptic Simulation	239
<i>O. Khatib, O. Brock, K.C. Chang, D. Ruspini, L. Sentis, S. Viji</i>	
1 Introduction	239
2 Whole-Robot Control: Task and Posture	241
3 Interactive Haptic Simulation	246
4 Task-Consistent Elastic Plans	247
5 Conclusion	251
Collaboration, Dialogue, and Human-Robot Interaction	255
<i>Terrence Fong, Charles Thorpe, Charles Baur</i>	
1 Introduction	255
2 Dialogue	257
3 System Design	258
4 Results	261
5 Discussion	264
6 Related Work	265

Part 6. Applications

Session Summary	269
<i>Paolo Dario</i>	
Vertebrate-type Perception and Gaze Control for Road Vehicles	271
<i>M. Pellkofer, M. Lützeler, E. D. Dickmanns</i>	
1 Introduction	271
2 Motivation	272
3 Camera Configuration	273
4 Gaze Control Unit	274
5 Requesting Attention	276
6 Active Vision for Road Recognition	277
7 Optimization of Gaze Behavior	280
8 Experimental results	284
9 Conclusion	286
Towards a Realistic Medical Simulator using Virtual Environments and Haptic Interaction	289
<i>Christian Laugier, César Mendoza, Kenneth Sundaraj</i>	
1 Introduction	289
2 Physical Models for Soft Tissue	291
3 Numerical Resolution	296
4 Real-Time Interactions	298
5 Conclusion	304
Spoken Language Interface of the Jijo-2 Office Robot	307
<i>Toshihiro Matsui, Hideki Asoh, Futoshi Asano, John Fry, Isao Hara, Yoichi Motomura and Katsunobu Itoh</i>	
1 Introduction	307
2 Robot Architecture	308
3 Navigation and Map	309
4 Spoken Language Dialogue System	310
5 Task Execution	313
6 Example Dialogue and Behavior	314
7 Conclusion	316
Intelligent Home Appliances	319
<i>Henrik I. Christensen</i>	
1 Introduction	319
2 Example Systems	320
3 Analysis/Synthesis	324
4 Summary	326

Part 7. Field Robots

- Session Summary** 331
Chuck Thorpe, Hugh Durrant-Whyte

Part 8. Vision-Based Navigation

Session Summary	343
<i>Yoshiaki Shirai</i>	
Vision-based Navigation, Environmental Representations and Imaging Geometries	347
<i>José Santos-Victor, Alexandre Bernardino</i>	
1 Introduction	347
2 Imaging geometries	348
3 Environmental representations	350
4 Examples of Navigation and Vision based Control	353
5 Conclusions	359
Mobile Robot Navigation As A Planar Visual Servoing Problem	361
<i>Peter Corke</i>	
1 Introduction	361
2 Visual Servoing Revisited	362
3 A Planar Formulation For Navigation	364
4 What Can We Learn From The Visual Servoing Literature?	369
5 Conclusions	370
Landing Strategies in Honeybees, and Applications to UAVs	373
<i>M. V. Srinivasan, S. W. Zhang, J. S. Chahl, M. A. Garryatt</i>	
1 Introduction	373
2 How Bees Perform Smooth Landings	373
3 Tests on a Robotic Gantry	377
4 Implementation in flying vehicles	381
5 Conclusions	382
Visual navigation in a plane using the conformal point	385
<i>Richard Hartley, Chanop Silpa-Anan</i>	
1 Motion in a plane	385
2 Proof of the angle construction	386
3 Determining the position of the conformal point	387
4 Homography from conformal point	388
5 Camera moving in a plane	391
6 Motion estimation in planar motion	392
7 Examples	393
8 Conclusion	396

Part 9. Planning and Modeling

Session Summary	401
<i>Bob Bolles</i>		
A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking	403
<i>Gildardo Sánchez, Jean-Claude Latombe</i>		
1 Introduction	403
2 Definitions and Notations	405
3 Experimental Foundations	406
4 Description of SBL	407
5 Experimental Results	412
6 Conclusion	416
Geometrical Interpretation of the CCT Stiffness Mapping for Serial Manipulators	419
<i>Chintien Huang, Imin Kao</i>		
1 Introduction	419
2 Screw-Based Jacobian Matrices of Serial Manipulators	420
3 The Conservative Congruence Transformation	422
4 The Interpretation of K_g	423
5 Example of the Planar 3R Manipulator	424
6 Discussion	428
7 Conclusion	430
A Dynamic Model of Contact Between a Robot and an Environment with Unknown Dynamics	433
<i>Roy Featherstone</i>		
1 Introduction	433
2 Background	434
3 Dual Vector Spaces	435
4 Dynamic Model of Contact	439
5 Analysis	441
6 Control	443
7 Conclusion	445

Part 10. Robot Technology I

Session Summary	449
<i>John Hollerbach, Jan-Olof Eklundh</i>	
AURORA – Minimalist Design for Tracked Locomotion	453
<i>Dr. Hagen Schempf</i>	
1 Introduction	453
2 Background	454
3 Performance Requirement	455
4 System Description	455
5 Prototype Testing	463
6 Summary & Conclusion	463
7 Future Work	464
Analysis of Robotic Locomotion Devices for the Gastrointestinal Tract ..	467
<i>L. Phee, A. Menciassi, D. Accoto, C. Stefanini, P. Dario</i>	
1 Introduction	467
2 Inchworm Type Locomotion	468
3 Types of Actuators and Mechanisms	474
4 First Prototype for Navigation in the Colon: Experimental Results	476
5 Modified Clampers Disposition: A Second Prototype	479
6 Conclusions	481
Advanced Sonar Sensing	485
<i>Lindsay Kleeman</i>	
1 Introduction	485
2 Sensor Design	486
3 Range and Bearing Measurements	489
4 Natural Selection of Landmarks	492
5 Discrimination	492
6 Interference Rejection	493
7 Classification	495
8 Some Underwater Results	496
9 Conclusions	497
Mechano-Media that Transmit Kinesthetic Knowledge from a Human to Other Humans	499
<i>Yasuyoshi Yokokohji, Yoshihiko Sugawara, Junji Kinoshita, Tsuneo Yoshikawa</i>	
1 Introduction	499
2 WYSIWYF display	501
3 Accurate Image Overlay on HMDs by Hybrid Vision/Inertial Tracking ..	504
4 Encountered-type Haptic Device That Renderes Multiple Virtual Objects ..	506
5 Mechano-Media	510
6 Conclusion	511

Part 11. Robot Technology II

Session Summary	515
<i>Raymond Austin Jarvis, Alex Zelinsky</i>	
The Evolution of a Robot Soccer Team	517
<i>Claude Sammut, Bernhard Hengst</i>	
1 Introduction	517
2 The Sony Legged Robot League	518
3 History	518
4 Vision	520
5 Localisation	523
6 Locomotion	525
7 Communication	527
8 Game Play Behaviours	528
9 Conclusion	529
A New Class of Hybrid Motion Simulation Using a Very Fast Parallel Robot	531
<i>Masaru Uchiyama, Susumu Tarao, Hiroshi Kawabe</i>	
1 Introduction	531
2 A New Class of Hybrid Simulation	532
3 Design of a Fast Parallel Robot	533
4 The Second Prototype Robot	535
5 A High-Speed Hybrid Motion Simulator	536
6 Experiments on the Simulator	538
7 Conclusions	542
General Solution for Linearized Error Propagation in Vehicle Odometry	545
<i>Alonzo Kelly</i>	
1 Introduction	545
2 Linearized Error Dynamics	548
3 Solution to Linearized Systems	549
4 Application To Odometry	553
5 Validation	555
6 Conclusions	557
Probabilistic Adaptive Agent Based System for Dynamic State Estimation using Multiple Visual Cues	559
<i>Alvaro Soto, Pradeep Khosla</i>	
1 Introduction	559
2 Probabilistic Adaptive Agent Based System	561
3 Related Work	566
4 Implementation	566
5 Results	569
6 Conclusions	570

Part 1

Simultaneous Localization And Mapping

Session Summary

Raja Chatila

Chair

LAAS-CNRS, 7 Ave. du Colonel Roche 31077 Toulouse cedex, France

Autonomous navigation is the central task of a mobile robot. It can be decomposed into four interleaved steps that the robot has to solve: building adequate spatial representations, robot localization, motion decision and planning, and motion execution. These four steps may be based on different paradigms and methods.

It appeared very clearly in the mid eighties that the issue of uncertainties was a central one in this process. Sensors are indeed always imperfect: their data are incomplete and inaccurate. Environment representations should therefore explicitly take into account uncertainties (which should also be the case for motion planning and execution, but this is not the issue here). Another phenomenon also immediately aroused. The robot discovers its environment gradually and new perceptions must be integrated to the previously built model. For this, it is necessary to know – or estimate – the transform between the new observation and the model already built, which is nothing else than localizing the robot (the observation point) in the model. Incremental environment modelling and robot localization are therefore two *intimately related processes*. The problem to be solved is *simultaneous* or *concurrent* localization and mapping (SLAM or CML). Solving this problem requires to identify that an environment element perceived from different positions is actually the same, and this is complicated by uncertainties. Hence the *data association* problem. The SLAM process is depicted in figure 1.

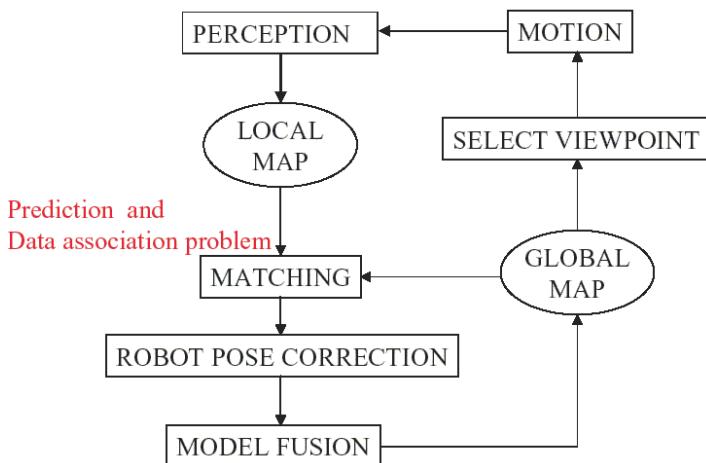


Fig. 1. The Simultaneous Localization and Mapping process

These issues are actually not new. When we look at old maps of coastlines and continents (e.g., from the Sixteenth Century), we can understand why they are inaccurate and twisted: the explorers and cartographers had to solve exactly a SLAM problem with inaccurate sensors. And in everyday life, everyone encountered a similar problem when arriving, say, to a new office building where all corridors and offices look alike.

Addressing this problem required a solid mathematical formalism to represent and handle uncertainties, and this was the probabilistic framework. Methods for computing uncertain transforms were developed and it soon appeared that a basic tool for representing and tracking uncertain representations by using uncertain observations was the Kalman filter, extended for the non-linear case. It was also established that there were correlations between observations and between observations and state, and that they imperatively should be considered in the filter formalization.

Several applications were investigated, mostly in indoors environments and for rather structured (e.g., lines or planes) or discrete (e.g., grids) representations – including generalization to multi-robot mapping – which enabled to understand the problems more thoroughly.

In the second half of the nineties, new ideas emerged for addressing SLAM and the localization problem, using a collection of techniques such as EM or particle filtering, which showed to be very fertile and were an essential step towards solving the problem.

Some of the important open issues are: complexity (i.e, how to deal with a large – and growing – number of environment features); data association (which stems partly from the poor representations used: it is difficult indeed to associate similar and close data when they are points or edges, but this would be much easier for more structured representation which have intrinsically recognisable features); more complex environments (i.e., outdoors, non structured); and uncertainty representation.

The four talks of this session deal with some of these issues and provide an excellent sample of the latest results on SLAM.

The paper by Leonard, Newman, Rokoski, Neira and Tardos presents recent results on feature-based representations and deals with the data association problem by introducing a novel scheme: delayed decision for combining data. The application is modeling large building environments. The authors also show how the Hough transform which is adapted to detect linear features can be used within this framework.

Thrun, Burgard, Chakrabarti, Emery and Liu address volumetric models for environment reconstruction using vision and laser by Expectation Maximisation. 3D representations pose a complexity problem which is solved here by fitting (probabilistically) flat surfaces to the data. The algorithm fuses nearby surfaces and a post-processing smoothes the surfaces to obtain more accurate representations.

Guivant and Nebot deal with Landmark-based localization. In order to reduce complexity, two main related directions were proposed in the past: hierarchical representations and decorrelation. They investigate the latter – which is suboptimal – and use a compressed Extended Kalman Filter that enables to update a local map

only, instead of the global representation at each step. The application is outdoor environments and the landmarks are trees.

Durrant-Whyte, Majumder, Battista and Scheding present a general Bayesian framework for solving SLAM in real time (Kalman Filter is such a Bayesian technique). They address uncertainty representation and propose an elegant means, the sum of gaussians, to approximate arbitrary probability distributions. The application is subsea terrain mapping by sonar and visual sensing.

Towards Robust Data Association and Feature Modeling for Concurrent Mapping and Localization

John J. Leonard¹, Paul M. Newman¹, Richard J. Rikoski¹, José Neira², and Juan D. Tardós²

¹ Massachusetts Institute of Technology, Cambridge MA 02139, USA

² Universidad de Zaragoza, 50015 Zaragoza, Spain

Abstract. One of the most challenging aspects of concurrent mapping and localization (CML) is the problem of data association. Because of uncertainty in the origins of sensor measurements, it is difficult to determine the correspondence between measured data and features of the scene or object being observed, while rejecting spurious measurements. This paper reviews several new approaches to data association and feature modeling for CML that share the common theme of combining information from multiple uncertain vantage points while rejecting spurious data. Our results include: (1) feature-based mapping from laser data using robust segmentation, (2) map-building with sonar data using a novel application of the Hough transform for perception grouping, and (3) a new stochastic framework for making delayed decisions for combination of data from multiple uncertain vantage points. Experimental results are shown for CML using laser and sonar data from a B21 mobile robot.

1 Introduction

The problem of concurrent mapping and localization (CML) for an autonomous mobile robot is stated as follows: starting from a initial position, a mobile robot travels through a sequence of positions and obtains a set of sensor measurements at each position. The goal is for the mobile robot to process the sensor data to produce an estimate of its position while concurrently building a map of the environment. While the problem of CML is deceptively easy to state, it presents many theoretical challenges. The problem is also of great practical importance; if a robust, general-purpose solution to CML can be found, then many new applications of mobile robotics will become possible.

CML, also referred to as SLAM (simultaneous localization and map building), has been a recurring theme at the series of ISRR Symposia over the years (Brooks, 1984; Chatila, 1985; Moutarlier, Chatila, 1989; Smith, Cheeseman, 1987). For example, in his paper for the second ISRR symposium, Brooks (Brooks, 1984) was among the first to suggest that a probabilistic approach was necessary to develop robust algorithms for mapping and navigation:

“Mobile robots sense their environment and receive error laden readings. They try to move a certain distance and direction, only to do so approximately. Rather than try to engineer these problems away it may be possible, and may be necessary, to develop map mapping and navigation algorithms which explicitly represent these uncertainties, but still provide robust information (Brooks, 1984).”

The key technical difficulty in performing CML is coping with uncertainty. Three distinct forms of uncertainty – data association uncertainty, navigation error, and sensor noise – work together to present a challenging data interpretation problem. For example, Figures 1 and 2 show the laser and sonar data, respectively, collected by a B21 mobile robot during several back-and-forth traverses of a corridor a few tens of meters long. Figure 3 shows the accumulation of dead-reckoning error during a longer duration traverse of about 500 meters in the MIT “infinite corridor”.

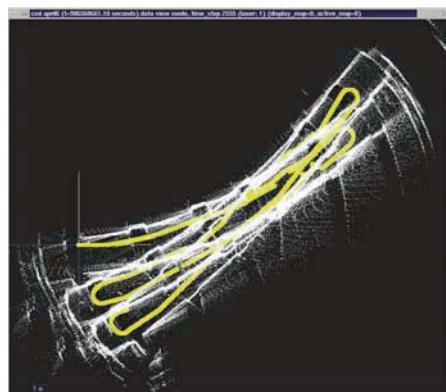


Fig. 1. Laser data for a short corridor experiment, referenced to the dead-reckoning position estimate

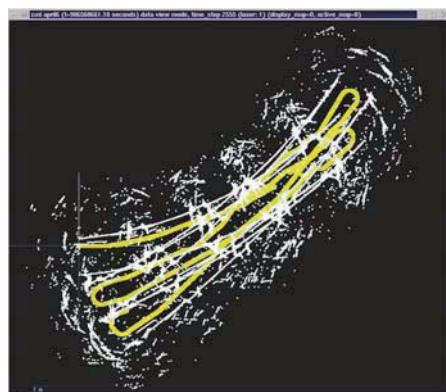


Fig. 2. Sonar data for a short corridor experiment, referenced to the dead-reckoning position estimate

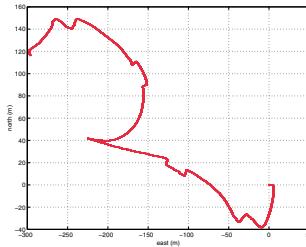


Fig. 3. Accumulation of position error relying only on dead-reckoning for a long distance traverse of the B21 mobile robot. The actual vehicle path went down approximately 40 meters, to the left approximately 225 meters, and then back to the origin.

Most successful recent implementations of CML have either been performed with SICK laser scanner data (Gutmann, Konolige, 1999; Thrun, 2001) or in environments that consist of isolated “point” objects (Castellanos, Tardos, 2000; Dissanayake, Newman, Durrant-Whyte, Clark, Csorba, 2001). However, there are many important applications of mobile robots where maps need to be built of complex environments, consisting of composite features, from noisy sensor data. The goal of our work is to enable autonomous underwater vehicles to navigate autonomously using sonar. Current methods for data association in feature-based CML are unable to cope with sonar because of its sparse and ambiguous nature.

Gutmann, Konolige (1999); Thrun (2001) have developed implementations of CML using laser data that are capable of closing moderately sized loops in real-time. In their work, the representation consists of “raw” sensor data referenced back to a complete trajectory of the vehicle. With this representation, they are able to greatly simplify the data association problem. CML algorithms that use a feature-based representation must explicitly solve the data association problem for each sensor measurement. Given a new sensor measurement, does it correspond to a previously mapped feature, a new feature that should be mapped, or is it spurious and should be ignored?

A key benefit of the SICK laser scanner is that the data from one position can be directly correlated with data taken from a nearby position, to compute the offset in robot position between the two positions. With sonar, the raw data is usually too noisy and ambiguous for this type of approach to work.

Recent work in feature-based CML has shown the importance of maintaining spatial correlations to achieve consistent error bounds (Castellanos, Tardos, 2000; Dissanayake et al., 2001). The representation of spatial correlations results in an $\mathcal{O}(n^2)$ growth in computational cost (Moutarlier, Chatila, 1989), motivating techniques to address the map scaling problem through spatial and temporal partitioning (Davison, 1998; Guivant, Nebot, 2001; Leonard, Feder, 2000). Almost all implementations of feature-based CML to-date have used fairly simple nearest-neighbor gating techniques. A more powerful technique that tests the Joint Compatibility of

multiple sensor measurements, using a branch and bound algorithm, has been developed (Neira, Tardós, 2001).

In this paper, we present results from several different new implementations of CML using either sonar or laser data. The results demonstrate feature classification and mapping from multiple uncertain vantage points. Section 2 presents results from a real-time implementation of CML with laser data that uses techniques from robust statistics for line segment extraction. Section 3 presents map-building results with sonar using a novel application of the Hough transform for perception grouping. Experimental results for sonar map-building and laser map-building of the same scene are compared. Section 4 summarizes a new stochastic framework for making delayed decisions to enable combination of data from multiple uncertain vantage points. Sonar data processing results are presented. Finally, Section 5 draws some conclusions and discusses challenges for future research.

2 “Explore and return” using Laser

This section presents results from use a new, generic, real-time implementation of feature-based CML. Novel characteristics of this implementation include: (1) a hierarchical representation of uncertain geometric relationships that extends the SPMMap framework (Castellanos, Tardos, 2000), (2) use of robust statistics to perform extraction of line segments from laser data in real-time, and (3) the integration of CML with a “roadmap” path planning method for autonomous trajectory execution. These innovations are combined to demonstrate the ability for a mobile robot to autonomously return back to its starting position within a few centimeters of precision, despite the presence of numerous people walking through the environment.

The sensors used were a SICK laser scanner and wheel encoders mounted on the B21 vehicle. The floor surface was a combination of sandstone tiles and carpet mats providing alternatively high and low wheel slippage. The exploration stage was manually controlled although it should be emphasized that this was done *without* visual contact with the vehicle. The output of the system was rendered in 3D and used as a real-time visualization tool of the robots workspace. This enabled the remote operator to “visit” previously un-explored areas while simultaneously building an accurate geometric representation of the environment. This in itself is a useful application of CML; nevertheless, future experiments will implement an autonomous explore function as well as the existing autonomous return.

To illustrate the accuracy of the CML algorithm the starting position of the robot was marked with four ten-cent coins; the robot then explored its environment and when commanded used the resulting map to return to its initial position and park itself on top of the coins with less than 2cm of error. The duration of the experiment was a little over 20 minutes long with just over 6MB of data processed. The total distance traveled was well in excess of 100m. Videos of various stages of the experiment can be found in various formats at <http://oe.mit.edu/~pnewman>.

Figure 6 shows the environment in which the experiment occurred. The main entrance hall to the MIT campus was undergoing renovation during which large

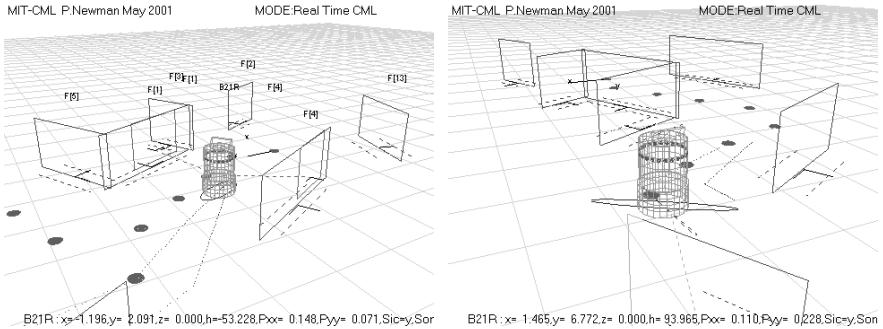


Fig. 4. Re-observing an existing feature

Fig. 5. Creating a new feature in the foreground following a rotation

wood-clad pillars had been erected throughout the hallway yielding an interesting, landmark rich and densely populated area.

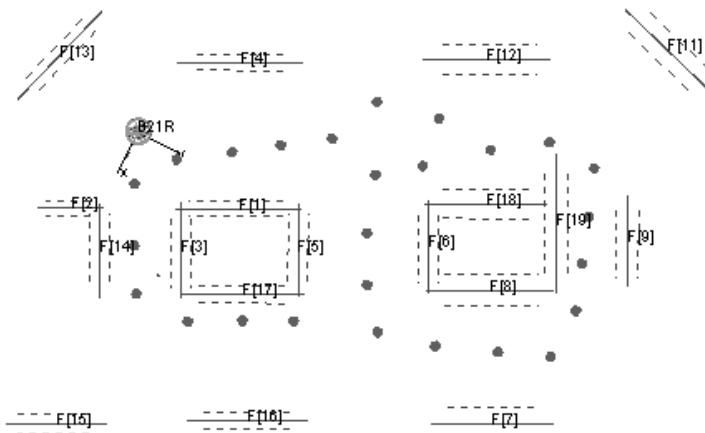


Fig. 6. The experiment scene

Figures 4 and 5 show rendered views of the estimated map during the exploration phase of the experiment. In Figure 4 the robot can be seen to be applying a line segment observation of an existing feature. In contrast Figure 5 shows an observation initializing a new feature just after the robot has turned a corner. The dotted lines parallel to the walls are representations of the uncertainty of lateral uncertainty in that wall feature. The vehicle was started with an initial uncertainty of 0.35 m and as shown in Dissanayake et al. (2001) all features will inherit this uncertainty as a limiting lower bound in their own uncertainty. The 1σ uncertainty of the vehicle location is shown as a dotted ellipse around the base of the vehicle.

MIT-CML P.Newman May 2001

MODE:Homing



B21R : x= -0.012,y= -0.002,z= 0.000,h= 3.266,Pxx= 0.055,Pyy= 0.063,Sic=y,Son:

Fig. 7. A plan view of the CML map at the end of the experiment. The approximate size of the environment was a 20m by 15m rectangle.

Figure 7 shows an OpenGL view of the estimated map towards the end of the experiment when the robot is executing its homing algorithm. The circles on the ground mark the free space markers that were dropped during the exploration phase of the experiment. The homing command was given when the robot was at the far corner of the hallway. Using the output of the CML algorithm, the robot set the goal marker to be the closest way point. When the algorithm deduces that the vehicle is within an acceptable tolerance ϵ of the present goal marker it sets the goal way-point to be the closest marker that has score less than the present goal marker. This then proceeds until the goal marker is the origin or initial robot position. At this point the goal seeking tolerance ϵ is reduced to 1cm. The program spent about thirty seconds commanding small adjustments to the location and pose of the robot before declaring that the vehicle had indeed arrived back at its starting location. Figure 8 and 9 show the starting and finishing positions with respect to the coin markers. As can be seen in these figures the vehicle returned to within an inch of the starting location. Readers are invited to view videos of this experiment and others including navigation in a populated museum at <http://oe.mit.edu/~pnewman>.

3 Sonar Perceptual Grouping Using the Hough Transform

The data from a standard ring of Polaroid sonar sensors can be notoriously difficult to interpret. This leads many researchers away from a geometric approach to sonar

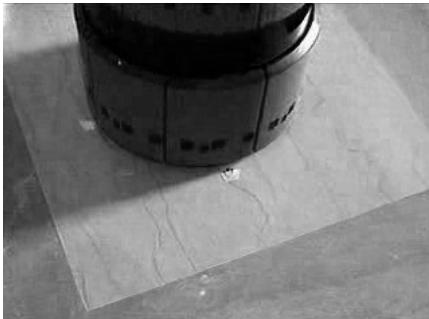


Fig. 8. The starting position



Fig. 9. The robot position after the completion of the homing leg of the mission

mapping. However, using a physics-based sensor model, the geometric constraints provided by an individual sonar return can be formulated (Leonard, Durrant-Whyte, 1992). Each return could originate from various types of features (point, plane, etc.) or could be spurious. For each type of feature, there is a limited range of locations for a potential feature that are possible. Given these constraints, the Hough transform (Ballard, Brown, 1982) can be used as a voting scheme identify point and planar features. More detail on this technique can be found in Tardós, Neira, Newman, Leonard (2002). A related technique called triangulation-based fusion has been developed in Wijk, Christensen (2000) for point objects only. Figure 10 through 12 provide an illustrative result for this approach. The Hough transform is applied to small batches of sonar data (22 positions each) as a pre-filter to look for potential new features in the sonar data. These groupings are then fed into an implementation of CML that uses the SPMMap as the state estimation framework (Castellanos, Tardos, 2000), Joint Compatibility for data association (Neira, Tardós, 2001), and a new technique called Sequential Map Joining (Tardós et al., 2002). Figure 13 shows a map of the same environment built from laser data. One can see that sonar map is almost as good as the laser map.

4 Delayed Stochastic Mapping

Stochastic mapping is a feature-based concurrent mapping and localization algorithm that was first published in Moutarlier, Chatila (1989); Smith, Self, Cheeseman (1990). The method assumes that there are n features in the environment, and that they are static. The true state at time k is designated by $\mathbf{x}[k] = [\mathbf{x}_r[k]^T \mathbf{x}_f[k]^T]^T$, where $\mathbf{x}_r[k]$ represents the location of the robot, and $\mathbf{x}_f[k]^T = [\mathbf{x}_{f_1}[k]^T \dots \mathbf{x}_{f_n}[k]^T]^T$ represent the locations of the environmental features. Let $\mathbf{z}[k]$ designate the sensor measurements obtained at time k , and Z^k designate the set of all measurements obtained from time 0 through time k . The extended Kalman filter to compute recursively a state estimate $\hat{\mathbf{x}}[k|k] = [\hat{\mathbf{x}}_r[k|k]^T \hat{\mathbf{x}}_f[k|k]^T]^T$ at each

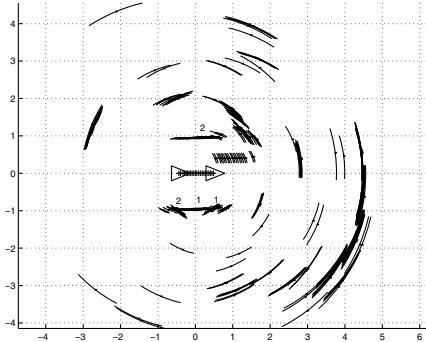


Fig. 10. Example of Hough processing to extract point and line features. Sonar returns are processed in a group of twenty-two positions. A voting scheme is performed to find clusters of measurements that hypothesize the existence of point and plane features. For this example, two planes and two points have been found.

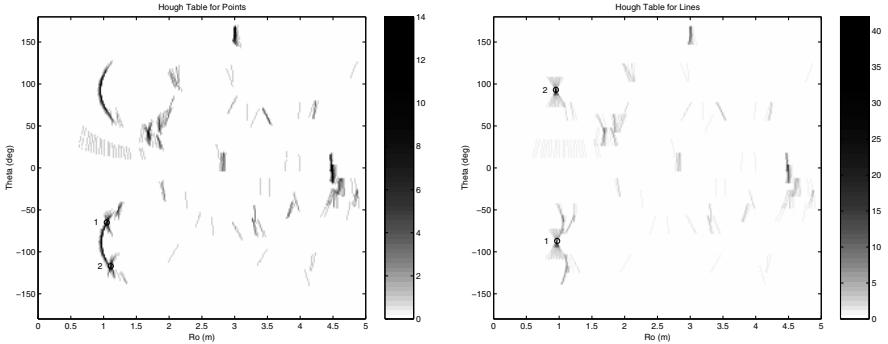


Fig. 11. Hough voting table for point (left) and line (right) features.

discrete time step k , where $\hat{x}_r[k|k]^T$ and $\hat{x}_f[k]^T = [\hat{x}_{f_1}[k]^T \dots \hat{x}_{f_n}[k]^T]^T$ are the robot and feature state estimates, respectively. The stochastic mapping equations are not repeated here, for more detail, see Feder, Leonard, Smith (1999); Smith et al. (1990).

Data association decisions must be made for each new measurement to determine if (1) it originates from one of the features currently in the map, (2) it originates from a new feature, or (3) it is spurious. In general, the data association problem is exponentially complex (Bar-Shalom, Fortmann, 1988), and no general solution that can run in real-time has been published. The motivation for delayed stochastic mapping is to be able to consider various hypothesis for the origins of measurements in a computationally efficient manner.

An assumption commonly employed in previous work is that the state of the new feature, $\hat{x}_{f_{n+1}}[k]$ can be computed using the measurement data available from

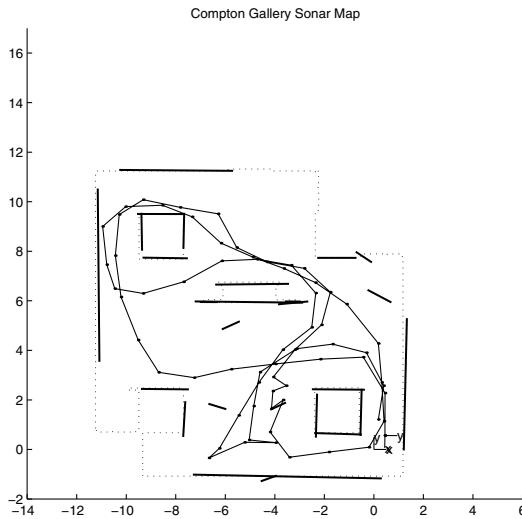


Fig. 12. Complete map for the MIT Compton Gallery built from sonar using Hough grouping, Map Joining, and Joint Compatibility.

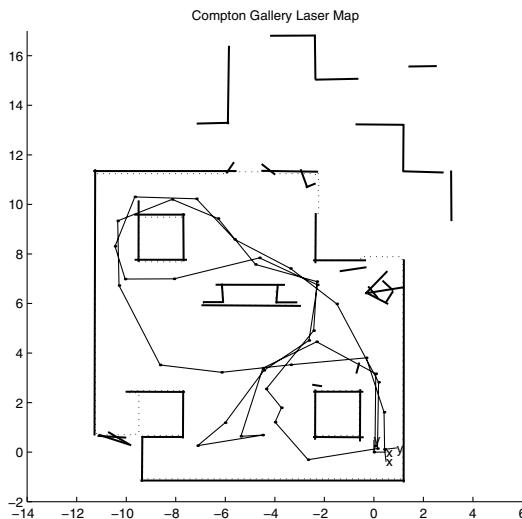


Fig. 13. Complete map for the MIT Compton Gallery built from laser data using Robust Statistics, Map Joining, and Joint Compatibility.

a single vehicle position, using a feature initialization function $\mathbf{g}(\cdot)$:

$$\hat{\mathbf{x}}_{f_{n+1}}[k] = \mathbf{g}(\hat{\mathbf{x}}[k|k], z_j[k]). \quad (1)$$

To be able to perform feature initialization from multiple vantage points, the representation is expanded to add a number of previous vehicle locations to the state vector. We refer to these states as trajectory states. Each time the vehicle moves, the previous vehicle location is added to the state vector. We use the notation $\hat{\mathbf{x}}_{t_i}[k]$ to refer to the estimate of the state (position) of the robot at time i given all information up to time k . The complete trajectory of the robot for time step 0 through time step $k - 1$ is given by the vector $\hat{\mathbf{x}}_t[k] = [\hat{\mathbf{x}}_{t_0}[k]^T \ \hat{\mathbf{x}}_{t_1}[k]^T \ \hat{\mathbf{x}}_{t_2}[k]^T \ \dots \ \hat{\mathbf{x}}_{t_{k-1}}[k]]^T$. The complete state vector is:

$$\hat{\mathbf{x}}[k|k] = \begin{bmatrix} \hat{\mathbf{x}}_r[k|k] \\ \hat{\mathbf{x}}_t[k] \\ \hat{\mathbf{x}}_f[k] \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_r[k|k] \\ \hat{\mathbf{x}}_{t_0}[k] \\ \hat{\mathbf{x}}_{t_1}[k] \\ \hat{\mathbf{x}}_{t_2}[k] \\ \vdots \\ \hat{\mathbf{x}}_{t_{k-1}}[k] \\ \hat{\mathbf{x}}_{f_1}[k] \\ \hat{\mathbf{x}}_{f_2}[k] \\ \hat{\mathbf{x}}_{f_3}[k] \\ \vdots \\ \hat{\mathbf{x}}_{f_{n-1}}[k] \\ \hat{\mathbf{x}}_{f_n}[k] \end{bmatrix}. \quad (2)$$

The associated covariance matrix is:

$$\mathbf{P}[k|k] = \begin{bmatrix} \mathbf{P}_{rr}[k|k] & \mathbf{P}_{rt}[k|k] & \mathbf{P}_{rf}[k|k] \\ \mathbf{P}_{tr}[k|k] & \mathbf{P}_{tt}[k|k] & \mathbf{P}_{tf}[k|k] \\ \mathbf{P}_{fr}[k|k] & \mathbf{P}_{ft}[k|k] & \mathbf{P}_{ff}[k|k] \end{bmatrix}. \quad (3)$$

New trajectory states are added to the state vector each time step by defining a new trajectory state $\hat{\mathbf{x}}_{t_k}[k] = \hat{\mathbf{x}}_r[k|k]$ and adding this to the state vector:

$$\hat{\mathbf{x}}[k|k] \leftarrow \begin{bmatrix} \hat{\mathbf{x}}_r[k|k] \\ \hat{\mathbf{x}}_{t_0}[k] \\ \vdots \\ \hat{\mathbf{x}}_{t_{k-1}}[k] \\ \hat{\mathbf{x}}_{t_k}[k] \\ \hat{\mathbf{x}}_f[k] \end{bmatrix}. \quad (4)$$

The state covariance is expanded as follows:

$$\mathbf{P}[k|k] \leftarrow \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{rt_0} & \dots & \mathbf{P}_{rt_k} & \mathbf{P}_{rf} \\ \mathbf{P}_{t_0r} & \mathbf{P}_{t_0t_0} & \dots & \mathbf{P}_{t_0t_k} & \mathbf{P}_{t_0f} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{P}_{t_{k-1}r} & \mathbf{P}_{t_{k-1}t_0} & \dots & \mathbf{P}_{t_{k-1}t_k} & \mathbf{P}_{t_{k-1}f} \\ \mathbf{P}_{t_kr} & \mathbf{P}_{t_kt_0} & \dots & \mathbf{P}_{t_kt_k} & \mathbf{P}_{t_kf} \\ \mathbf{P}_{fr} & \mathbf{P}_{ft_0} & \dots & \mathbf{P}_{ft_k} & \mathbf{P}_{ff} \end{bmatrix}, \quad (5)$$

where $\mathbf{P}_{t_k t_i} = \mathbf{P}_{r t_i}$, $\mathbf{P}_{t_k f} = \mathbf{P}_{r f}$, and $\mathbf{P}_{t_k t_k} = \mathbf{P}_{r r}$. The growth of the state vector in this manner increases the computational burden, however it is straightforward to delete old vehicle trajectory states and associated terms in the covariance, once all the measurements from a given time step have been either processed or discarded.

This process of adding past states is similar to a fixed-lag Kalman smoother (Anderson, Moore, 1979). In a fixed-lag smoother, states exceeding a certain age are automatically removed. In our approach, states are added and removed based on the data processing requirements of the stochastic mapping process. Unlike the fixed-lag smoother, states are not necessarily removed in the order in which they are added.

With the addition of prior vehicle states to the state vector, it now becomes possible to initialize new features using measurements from multiple time steps. For example, consider the initialization of a new feature using two measurements, $\mathbf{z}[k_1]$ and $\mathbf{z}[k_2]$, taken at time steps k_1 and k_2 . The state of the new feature can be computed using a feature initialization function involving data from multiple time steps:

$$\hat{\mathbf{x}}_{f_{n+1}} = g(\hat{\mathbf{x}}_{t_{k_1}}[k], \hat{\mathbf{x}}_{t_{k_2}}[k], [\mathbf{z}[k_1]^T \ \mathbf{z}[k_2]^T]^T). \quad (6)$$

For example, in two-dimensions if each measurement is a range-only sonar measurement, then the function $g(\cdot)$ represents a solution for the intersection of two circles. The procedure is the same if the feature initialization function $g(\cdot)$ is a function of measurements from more than two time steps.

Once a new feature is initialized, the map can be updated using all other previously obtained measurements that can be associated with the new feature. We call this procedure a “batch update”. It allows the maximum amount of information to be extracted from all past measurements. It also provides a means to incrementally build up composite models of more complex objects (Leonard, Rikoski, 2001). The method has been implemented as part of an integrated framework for real-time CML, which incorporates delayed state management, perceptual grouping, multiple vantage point initialization, batch updating, and feature fusion. Some illustrative results for this approach are presented in Figures 14 to 17, which show the results for processing of data in an MIT corridor. Further details can be found in Leonard, Rikoski, Newman, Bosse (2002). These experiments used the Hough Transform for sonar perceptual grouping as described above in Section 3.

This methodology provides a new generic framework for improved feature modeling and classification. The ability to perform a batch update using many previous

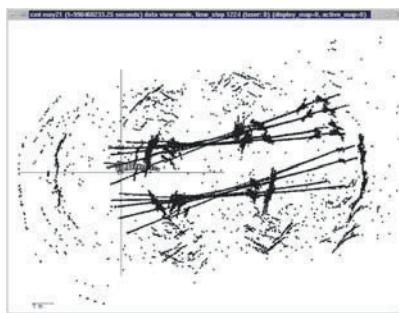


Fig. 14. Raw sonar data for corridor experiment, referenced to odometry.

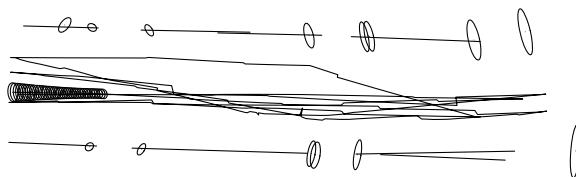


Fig. 15. CML estimated trajectory for corridor scene and estimated map consisting of points and line segments. Three-sigma error bounds are shown for the location of points.

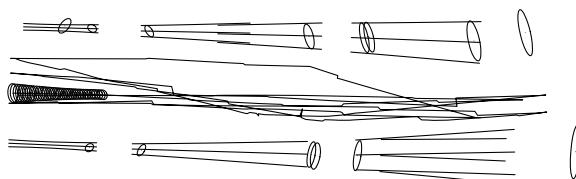


Fig. 16. Same plot as in Fig. 15 but with three-sigma error bounds for lines added.

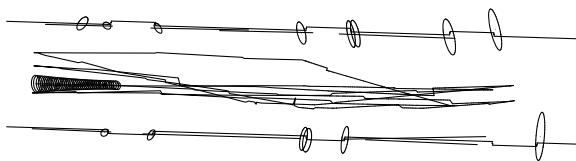


Fig. 17. Same plot as in Fig. 15 but with hand-measured model overlaid.

measurements provides a facility for making delayed data association decisions. If there is ambiguity about the correspondence between measurements and features, decisions can be postponed until additional information becomes available. Feature extraction is also simplified. The initialization of complex features in situations with high ambiguity can be greatly simplified by considering a batch of data obtained at multiple time steps. Efficient, non-stochastic perceptual grouping methods such as the Hough technique described above in Section 3 can be used to screen the data and make preliminary association decisions that can later be confirmed with delayed stochastic gating, and then applied via batch updating.

5 Conclusion

This paper has considered the development of improved data association and feature modeling techniques for CML. Experimental results have been shown for both Polaroid sonar and SICK laser scanner data from a B21 robot, operating in the corridors of MIT, using several new data association and feature modeling techniques. The ultimate goal of our research is to create a robust, consistent, convergent, and computationally efficient real-time algorithm for CML for large-scale environments.

Acknowledgments

This research has been funded in part by the Henry L. and Grace Doherty Assistant Professorship in Ocean Utilization, NSF Career Award BES-9733040, the MIT Sea Grant College Program under grant NA86RG0074, Ministerio de Educación y Cultura of Spain, grant PR2000-0104, and the Spain-US Commission for Educational and Scientific Exchange (Fulbright), grant 20079.

References

- Anderson BDO, Moore JB (1979) Optimal filtering. Prentice-Hall, Englewood Cliffs, N.J.
- Ballard D, Brown C (1982) Computer vision. Prentice-Hall.
- Bar-Shalom Y, Fortmann TE (1988) Tracking and data association. Academic Press.
- Brooks RA (1984) Aspects of mobile robot visual map making. In Second Int. Symp. Robotics Research, pp 287–293, Tokyo, Japan. MIT Press.
- Castellanos JA, Tardos JD (2000) Mobile robot localization and map building: A multisensor fusion approach. Kluwer Academic Publishers, Boston.
- Chatila R (1985) Mobile robot navigation space modeling and decisional processes. In Third Int. Symposium Robotics Research. MIT Press.
- Davison AJ (1998) Mobile robot navigation using active vision. PhD thesis, University of Oxford.
- Dissanayake MWMG, Newman P, Durrant-Whyte HF, Clark S, Csorba M (2001) A solution to the simultaneous localization and map building (slam) problem. IEEE Transactions on Robotic and Automation, 17(3):229–241.
- Feder HJS, Leonard JJ, Smith CM (1999) Adaptive mobile robot navigation and mapping. Int. J. Robotics Research, 18(7):650–668.

- Guivant J, Nebot E (2001) Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotic and Automation*, 17(3):242–257.
- Gutmann JS, Konolige K (1999) Incremental mapping of large cyclic environments. In *International Symposium on Computational Intelligence in Robotics and Automation*.
- Leonard JJ, Durrant-Whyte HF (1992) Directed sonar sensing for mobile robot navigation. Kluwer Academic Publishers, Boston.
- Leonard JJ, Feder HJS (2000) A computationally efficient method for large-scale concurrent mapping and localization. In Koditschek D, Hollerbach J (eds) *Robotics Research: The Ninth International Symposium*, pp 169–176, Snowbird, Utah. Springer Verlag.
- Leonard JJ, Rikoski R (2001) Incorporation of delayed decision making into stochastic mapping. In Rus D, Singh S (eds) *Experimental Robotics VII*, Lecture Notes in Control and Information Sciences. Springer-Verlag.
- Leonard JJ, Rikoski RJ, Newman PM, Bosse M (2002) Mapping partially observable features from multiple uncertain vantage points. *Int. J. Robotics Research*. To Appear.
- Moutarlier P, Chatila R (1989) Stochastic multi-sensory data fusion for mobile robot location and environment modeling. In *5th Int. Symposium on Robotics Research*, pp 207–216.
- Neira J, Tardós J (2001) Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. on Robotics and Automation*, 17(6):890–897.
- Smith R, Cheeseman P (1987) On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56.
- Smith R, Self M, Cheeseman P (1990) Estimating uncertain spatial relationships in robotics. In Cox I, Wilfong G (eds) *Autonomous Robot Vehicles*, pp 167–193. Springer-Verlag.
- Tardós J, Neira J, Newman PM, Leonard JJ (2002) Robust mapping and localization in indoor environments using sonar data. *Int. J. Robotics Research*. To Appear.
- Thrun S (2001) An online mapping algorithm for teams of mobile robots. *Int. J. Robotics Research*, 20(5):335–363.
- Wijk O, Christensen H (2000) Triangulation based fusion of sonar data with application in robot pose tracking. *IEEE Trans. Robotics and Automation*, 16(6):740–752.

A Real-time Algorithm for Acquiring Multi-Planar Volumetric Models with Mobile Robots

Sebastian Thrun, Wolfram Burgard, Deepayan Chakrabarti, Rosemary Emery,
Yufeng Liu, and Christian Martin

Carnegie Mellon University, Pittsburgh PA 15213, USA

Abstract. This paper summarizes recent research on developing autonomous robot systems than can acquire volumetric 3D maps with mobile robots in real-time. The core of our system is a real-time version of the popular expectation algorithm, developed for extracting scalar surfaces from sets of range scans (Martin, Thrun, 2002). Maps generated by this algorithm consists of a small number of planar rectangular surfaces, which are augmented by fine-grained polygons for non-flat environmental features. Experimental results obtained in a corridor-type environment illustrate that compact and accurate maps can be acquired in real-time from range and camera data.

1 Introduction

In the past decades, the topic of robotic mapping has received considerable attention. By many, the map acquisition problem is perceived as one of the most challenging problems in mobile robotics (Thorpe, Durrant-Whyte, 2001). A flurry of recent work has produced a range of algorithms that work both offline and online – see (Thrun, 2002) for a survey. The vast majority of successful robot mapping algorithms represent maps by grids (Elfes, 1987), raw point measurements (Gutmann, Konolige, 2000; Lu, Milios, 1997), or collection of fine-grained polygons (Liu, Emery, Chakrabarti, Burgard, Thrun, 2001). For example, the popular *occupancy grid map* paradigm (Elfes, 1987) represents the result of mapping in a fine-grained grid. While such a methodology appears to be sufficient for navigating robots, it suffers from two major disadvantages. First, there are intrinsic scaling limitations. For example, while two-dimensional occupancy maps require relatively little memory, the memory requirements of the same approach in 3D are prohibitive (despite the noteworthy by Moravec and Martin(Moravec, Martin, 1994)). Second, and more importantly, such maps fail to capture the true nature of indoor environments. Indoor environments are often composed out of walls, furniture, doors, windows etc, many of which have specific geometric structure (e.g., are flat). An understanding of such objects and their geometric properties (e.g., flatness) will inevitably lead to new, more powerful mapping algorithms that can generate more accurate maps.

Our work is an attempt to take a step in the direction of building more compact maps that exploit some of the known structure of indoor environments. We have developed an algorithm that identifies flat rectangular surfaces form the range data collected by a mobile robot. The resulting models combine fine-grained polygonal representation of non-planar objects with larger, monolithic representations for

planar surfaces in the environment. The mathematical approach for finding planar objects is the expectation maximization (EM) algorithm (Dempster, Laird, Rubin, 1977). The EM approach combines a phase of searching for a compact map with planar rectangular surfaces with one that associates measurements with individual surfaces. By doing so, it can generate accurate surface maps even at the boundary of different surfaces. In addition, our approach allows for objects that are not part of any rectangular surface, which are then represented using fine-grained polygons. The localization necessary to build these maps is obtained through a recently developed 2D mapping and localization algorithm, capable of determining the robot pose in real-time (Thrun, 2001) to estimate pose.

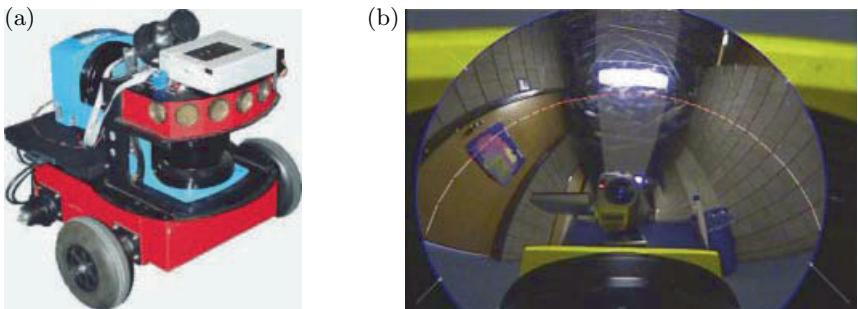


Fig. 1. (a) Mobile robot, equipped with two 2D laser range finders and a panoramic camera. (c) The panoramic camera is only a few centimeters away from the optical axis of the laser range finder.

The material presented here is previously been described in two separate papers, one focusing on the basic EM methodology (Liu et al., 2001) and one on our real-time implementation (Martin, Thrun, 2002). However, we report previously unpublished result form an implemented system in which all functions are computed strictly in real-time. In recent experiments, our approach has been applied to building multi-planar maps of an indoor environment in real-time. We present results of mapping multiple indoor environments using the robot shown in Fig. 1. This robot is equipped with a forward-pointed laser range finder for localization during mapping, and an upward-pointed laser range finder and panoramic camera for measuring the structure and texture of the environment in 3D. We also report results obtained by mapping texture onto those maps, obtained using an omni-directional camera mounted on the robot.

As noted above, robotic mapping is a highly active research area. The majority of robot mapping research has focused on building maps in 2D. Our approach is reminiscent of an early paper by Chatila and Laumond (Chatila, Laumond, 1985), who proposed to reconstruct low-dimensional line models in 2D from sensor measurements but provided no practical algorithm. Our work is also related to work on

line extraction from laser range scans (Lu, Milius, 1998). However, these methods address the two-dimensional case, where lines can be extracted from a single scan. Our approach is also related to (Iocchi, Konolige, Bajracharya, 2000), which reconstructs planar models of indoor environments using stereo vision, but does not generate such maps in real-time. Comparable approaches in the computer vision literature typically require human input for map generation, and often also require accurate camera pose information (Allen, Stamos, 2000; Bajcsy, Kamberova, Novcera, 2000; Debevec, Taylor, Malik, 1996; Shum, Han, Szeliski, 1998). A related algorithm for acquiring 3D maps in the presence of people has recently been described in (Hähnel, Schulz, Burgard, 2002).

2 Multi-Surface Models

As the core of our modeling algorithm is a probabilistic model of multi-planar environments and their effects on range sensors.

A *multi-surface map* – which models all planar regions of the environment – is a finite collection of rectangular flat surfaces. In indoor environments, these surfaces may represent doors, walls, ceilings, etc. They are denoted by θ_j , with

$$\theta = \{\theta_1, \dots, \theta_J\} \quad \text{with} \quad \theta_j = \langle \alpha_j, \beta_j, \gamma_j \rangle \quad (1)$$

Here J is the total number of rectangular surfaces. Each surface θ_j is described by 9 parameters, arranged in three groups. The vector α_j is the three-dimensional surface normal of the rectangular surface, β_j is the one-dimensional offset between the surface and the origin of the coordinate system, and γ_j are five parameters specifying the size and orientation of the rectangular area within the (infinite) planar surface represented by α_j and β_j . The surface normal α_j is a vector perpendicular to the surface with unit length.

Our surface model makes it straightforward to calculate the Euclidean distance of any coordinate z in 3D space to any surface θ_j . This distance function will be denoted $d(z, \theta_j)$. It is easily implemented by combining the standard point-to-surface, point-to-line, and point-to-point distance functions.

As stated above, the robot is localized during mapping using the real-time technique described in (Thrun, 2001). An example of the 2D mapping results obtained using this algorithm is shown in Fig. 2. The ability to perform localization in real-time enables us to map range measurements of both lasers into coordinates in 3D. These 3D coordinates are denoted z_i , and the set of all measurements is given by $Z = \{z_1, z_2, \dots\}$. We assume that measurements are generated according to the following probabilistic law, also known as *measurement model*:

$$p(z_i | \theta_j) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{d^2(z_i, \theta_j)}{2\sigma^2} \right\} \quad (2)$$

This is, we assume Gaussian measurement noise. In particular, let j be the index of the surface nearest to the measurement z_i . Then the error distribution is given by a

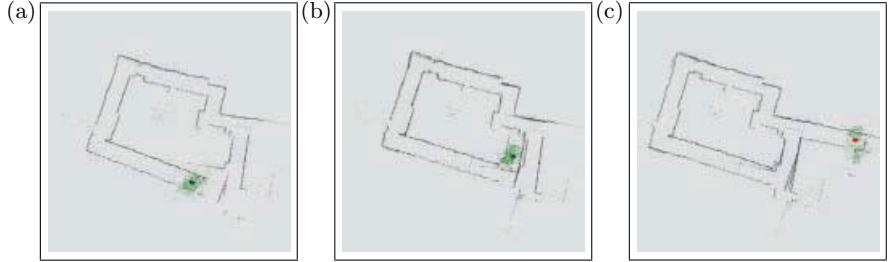


Fig. 2. Simultaneous localization and 2D mapping, using the real-time algorithm described in (Thrun, 2001). Shown here is a particularly challenging case, where the use of particle filters ensures that the cyclic environment can be mapped correctly despite considerable odometric error. See (Thrun, 2001) for more example and a description of the algorithm.

Gaussian with variance parameter σ . In addition, we explicitly model the case if a range finder fails to detect an object altogether (or the object detected by a range finder does not correspond to any of the flat surfaces in the model θ). Such events are modeled by a uniform distribution over the entire measurement range:

$$p(z_i | \theta_*) := \begin{cases} 1/z_{\max} & \text{if } 0 \leq z_i \leq z_{\max} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Here θ_* denotes a ‘phantom’ component of the map θ , which accounts for all measurements not caused by any of the surfaces in θ . The interval $[0; z_{\max}]$ denotes the measurement range of the range finder.

The definition of the sensor model enables us to define a likelihood function, which measures the probability of all measurements z as a function of the map θ . To define this likelihood function, we have to introduce the so-called *correspondence*, denoted c_{ij} and c_{i*} . Each correspondence variable is a binary random variable. The variable c_{ij} is 1 if and only if the i -th measurement z_i corresponds to the j -th surface in the map, θ_j . Likewise, the correspondence c_{i*} is 1 if and only if the i -th measurement was not caused by any of the surfaces in the map θ . The correspondence vector of the i -th measurement is given by

$$C_i = \{c_{i*}, c_{i1}, c_{i2}, \dots, c_{iJ}\} \quad (4)$$

Using these variables, the measurement model can be rewritten as follows (see (Liu et al., 2001) for details):

$$p(z_i | C_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2} \left[c_{i*} \ln \frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2} \right] \right\} \quad (5)$$

The *joint probability* of a measurement z_i along with its correspondence variables C_i is thus given by

$$p(z_i, C_i | \theta) = \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2} \left[c_{i*} \ln \frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2} \right] \right\} \quad (6)$$

This is because in the absence of any data, all correspondences are equally likely. Assuming independence in measurement noise, the logarithm of the likelihood of all measurements Z and their correspondences $C := \{C_i\}$ is given by

$$\ln p(Z, C | \theta) = \sum_i \ln \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} - \frac{1}{2} c_{i*} \ln \frac{z_{\max}^2}{2\pi\sigma^2} \frac{1}{2} \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2} \quad (7)$$

The log-likelihood is more convenient for optimization, since it contains a sum where the likelihood contains a product. Maximizing the log-likelihood is equivalent to maximizing the likelihood, since the logarithm is strictly monotonic. Finally, we notice that the correspondences are only interesting to the extent that they determine the most likely map θ . Therefore, the goal of estimation is to maximize the *expectation* of the log likelihood (7), where the expectation is taken over all correspondences C :

$$\begin{aligned} & E_C [\ln p(Z, C | \theta) | \theta, Z] \\ &= E_C \left[\sum_i \ln \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} - \frac{1}{2} c_{i*} \ln \frac{z_{\max}^2}{2\pi\sigma^2} - \frac{1}{2} \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2} \middle| \theta, Z \right] \\ &= \text{const.} - \frac{1}{2} \sum_i E[c_{i*} | \theta, z_i] \ln \frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j E[c_{ij} | \theta, z_i] \frac{d^2(z_i, \theta_j)}{\sigma^2} \end{aligned} \quad (8)$$

The later transformation is valid because of the linearity of the mathematical expectation.

3 Acquisition of Compact 3D Models in Real-Time

Our approach acquires multi-planar maps by maximizing the expected log likelihood (8). This is achieved via a modified version of the EM algorithm, a popular method for likelihood maximization with latent correspondence variables (Dempster et al., 1977).

3.1 Determining The Surface Parameters

Our approach generates a sequence of maps, $\theta^{[0]}, \theta^{[1]}, \theta^{[2]}, \dots$. Starting with a random map $\theta^{[0]}$, each update improves the log-likelihood of the data over the previous map until convergence. Each update is implemented in two steps: an E-step, where the expectations of the unknown correspondences $E[c_{ij}]$ and $E[c_{i*}]$ are calculated for the n -th map $\theta^{[n]}$, and an M-step, where a new maximum likelihood map $\theta^{[n+1]}$ is computed under these expectations. An important result is that for our planar map with uniform noise, both of these steps can be solved in closed form.

The E-Step: Here we are given a map $\theta^{[n]}$ for which we seek to determine the expectations $E[c_{ij}]$ and $E[c_{i*}]$ for all i, j . Bayes rule, applied to the sensor model, gives us a way to calculate the desired expectations (assuming a uniform prior over correspondences):

$$\begin{aligned} e_{ij}^{[n]} &= E[c_{ij} \mid \theta^{[n]}, z_i] = p(c_{ij} \mid \theta^{[n]}, z_i) \propto \exp \left\{ -\frac{1}{2} \frac{d^2(z_i, \theta_j)}{\sigma^2} \right\} \\ e_{i*}^{[n]} &= E[c_{i*} \mid \theta^{[n]}, z_i] = p(c_{i*} \mid \theta^{[n]}, z_i) = \exp \left\{ -\frac{1}{2} \ln \frac{z_{\max}^2}{2\pi\sigma^2} \right\} \end{aligned} \quad (9)$$

Here all expectations have to be normalized for each measurement z_i .

The M-Step: Here we are given the expectations $e_{ij}^{[n]}$ and $e_{i*}^{[n]}$ and seek to calculate a new model $\theta^{[n+1]}$ that maximizes the expected log-likelihood of the measurements, as given by Equation (8). Obviously, many of the terms in (8) do not depend on the map parameters θ . This allows us to simplify this expression and instead minimize

$$\theta^{[n+1]} = \operatorname{argmin}_{\theta} \sum_i \sum_j e_{ij}^{[n]} d^2(z_i, \theta_j) \quad (10)$$

Notice that this calculation can be carried out independently for each surface θ_j . This decomposition is a key advantage of the EM formulation of the mapping problem.

Our actual optimization algorithm proceeds in two steps. First, the parameters α_j and β_j are computed that specify the principal orientation and location of the rectangular surface, without considerations of the surface boundaries. This problem can be solved in closed form, which is essential for a fast online implementation. Second, our approach calculates the boundary parameters γ . It does this in a way that all measurements are included inside the rectangular bounding box of a surface whose most likely correspondence is this very surface. This operation is also easily performed in closed form and hence very fast. See (Liu et al., 2001; Martin, Thrun, 2002) for details.

3.2 Determining The Number Of Surfaces

Finally, the number of surfaces J has to be determined in the optimization. Our approach involves a simple Bayesian prior that penalizes complex maps using an exponential prior:

$$p(J) \propto \exp \{-\kappa J\} \quad (11)$$

Here κ is a constant factor (e.g., 0.02). The final map estimator is, thus, a maximum posterior probability estimator (MAP), which combines the complexity-penalizing prior with the data likelihood calculated by EM. In log form, we have as a log posterior:

$$p(\theta \mid Z) = \text{const.} + p(Z \mid \theta) - \kappa J \quad (12)$$

In practice, this approach implies that surfaces that are not supported by sufficiently many data measurements weighted by their expectation, are discarded. This approach makes it possible to choose the number of map components J (rectangular surfaces) dynamically, during the execution of EM. Our specific choice of starting and terminating surfaces will be described in the following section.

3.3 Real-Time Implementation

The EM algorithm, as stated so far, is off-line, since at any point in time constructing an accurate map requires multiple paths through the entire data set. Real-time performance is achieved through several modifications of the basic EM algorithm::

- The map generated at time $t - 1$ is used as initial map at time t .
- In the E-step, the expectation is only calculated for a small number of measurements, in a way that the total number of such calculations does not depend on the amount of available data. Expectations are re-calculated for new measurements, and for older measurements which meet several conditions: They lie at the boundary of two surfaces (judging from their maximum likelihood assignment) or are entirely unexplained by any existing surface, and they have been considered no more than 10 times in previous E-steps. The latter condition assures that measurements cannot be considered indefinitely in the E-step, hence bounds the number of measurements that can be considered in any E-step. The number of surfaces considered in the E-step is also bounded, and typically includes all nearby surfaces.
- In the M-step, only surfaces are re-estimated whose maximum likelihood assignments were changed in the E-step. This naturally guarantees that only a small number of surfaces is re-calculated in the M-step, with the total number being independent of the total number of surfaces J . However, each such surface might still be supported by an unbounded number of measurements, hence the calculation of surface parameters may take time that grows with the total number of measurements. To avoid the latter, the number of measurements involved in re-computing surface parameters in the M-step is subsampled so as to not to exceed 1,000.
- Finally, our approach implements Bayesian model selection in real-time. New surfaces may be introduced based on new measurements which are not “explained” (in a ML fashion) by any of the existing surfaces in the map. Surfaces are removed from the map if after 10 or more iterations of the basic algorithms, they are not supported by sufficiently many measurements in accordance with the map complexity penalty factor κ [see (12)]. In our implementation, this corresponds to a total of 500 measurements (in expectation).

As a result of these selective calculations, our implementation is strictly incremental, enabling us to find maps in real-time. At the same time, our approach retains the principle advantage of the EM approach, in that past correspondences are continuously re-evaluated as new data arrives. This is important in practice, since the correspondence of a measurements can often only be clarified in retrospect, as more data arrives.

4 Experimental Results

Our online EM algorithm for 3D multi-planar mapping was successfully tested in several corridor-style environments. All results shown in this paper were generated in real-time, in some cases using an autonomously exploring robot, in others using a manually controlled robot.

Figure 3 illustrates the online creation of a compact map. Shown in the left column are maps generated directly from the raw measurement data by creating polygons for any set of nearby measurements. Views of a map with full texture are shown in Fig. 4. This map, including the texture information, was entirely built in real-time. Fig. 5 shows the structure of a non-planar object (a trash bin) in one of the final maps. This example illustrates that the ability to merge flat surfaces with fine-grained polygons makes it possible to build complete maps of building interiors while exploiting the building’s planarity. Sequences of fully textured 3D maps, acquired in real-time, are shown in Figs. 6 and 7.

In an attempt to quantitatively evaluate our approach, we mapped three different corridor environments in different buildings. The complexity of those environments was comparable to the maps shown here. The number of initial polygons was between $3.5 \cdot 10^4$ and $6.5 \cdot 10^4$. The final maps contained on average 0.60% as many polygons (0.69%, 0.80%, and 0.32%), which corresponds to an average compression ratio of 1 : 192.

The identification of rectangular surfaces in the environment has a positive effect of the visual acuity of the map. Figure 8 shows inside projections of a 3D map built by EM and compress it to a map built without EM, using fine-grained polygonal maps. Obviously, the visual accuracy of texture projected onto flat surfaces is much higher than the renderings obtained from the fine-grained polygonal map. This illustrates that the resulting map is not only more compact, but also provides more visual detail than the map created without our approach.

5 Discussion

We have successfully developed real-time system for building compact maps of building interiors with mobile robots. This approach utilizes the expectation maximization (EM) algorithm for finding rectangular surface patches in 3D data, acquired by a moving robot equipped with laser range finders and a panoramic camera. While EM is traditionally an offline algorithm, a modified version of EM was developed capable of generating such maps online, while the robot is in motion. Extensive experimental results illustrate that this approach enables mobile robots to acquire compact maps of corridor-style environments.

Although stated here in the context of finding rectangular flat surfaces, the EM algorithm is more general in that it can easily handle a much richer variety of geometric shapes. The extension of our approach to richer classes of objects is subject to future research.

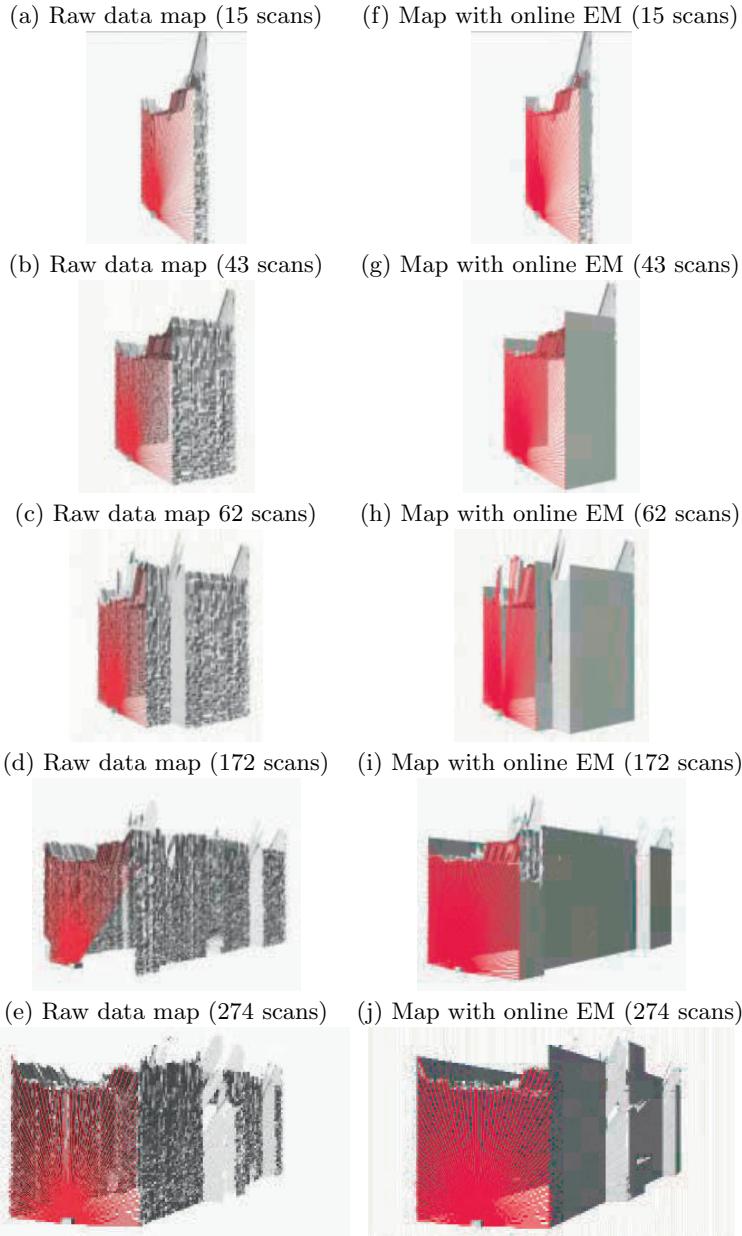


Fig. 3. Raw data (left column) and maps generated using online EM (right column). Some of the intermediate maps exhibit suboptimal structures, which are resolved in later iterations of EM. Despite this backward correction of past estimates, the algorithm presented in this paper still runs in real-time, due to careful selection of measurements that considered in the EM estimation.



Fig. 4. Views of a compact 3D texture map built in real time with an autonomously exploring robot.

Acknowledgment

This research is sponsored by DARPA's MARS Program (Contract number N66001-01-C-6018) and the National Science Foundation (CAREER grant number IIS-9876136 and regular grant number IIS-9877033), all of which is gratefully acknowledged. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the United States Government or any of the sponsoring institutions.



Fig. 5. View of a trash bin in the final map, with a large planar rectangular surface patch in the background. Our algorithm recognizes that this object cannot be explained by planar surfaces with sufficient likelihood, hence it retains the polygonal representation. The ability to merge polygonal surfaces with fine-grained polygons make it possible to build complete maps of building interiors, while exploiting planarity to the extent possible.

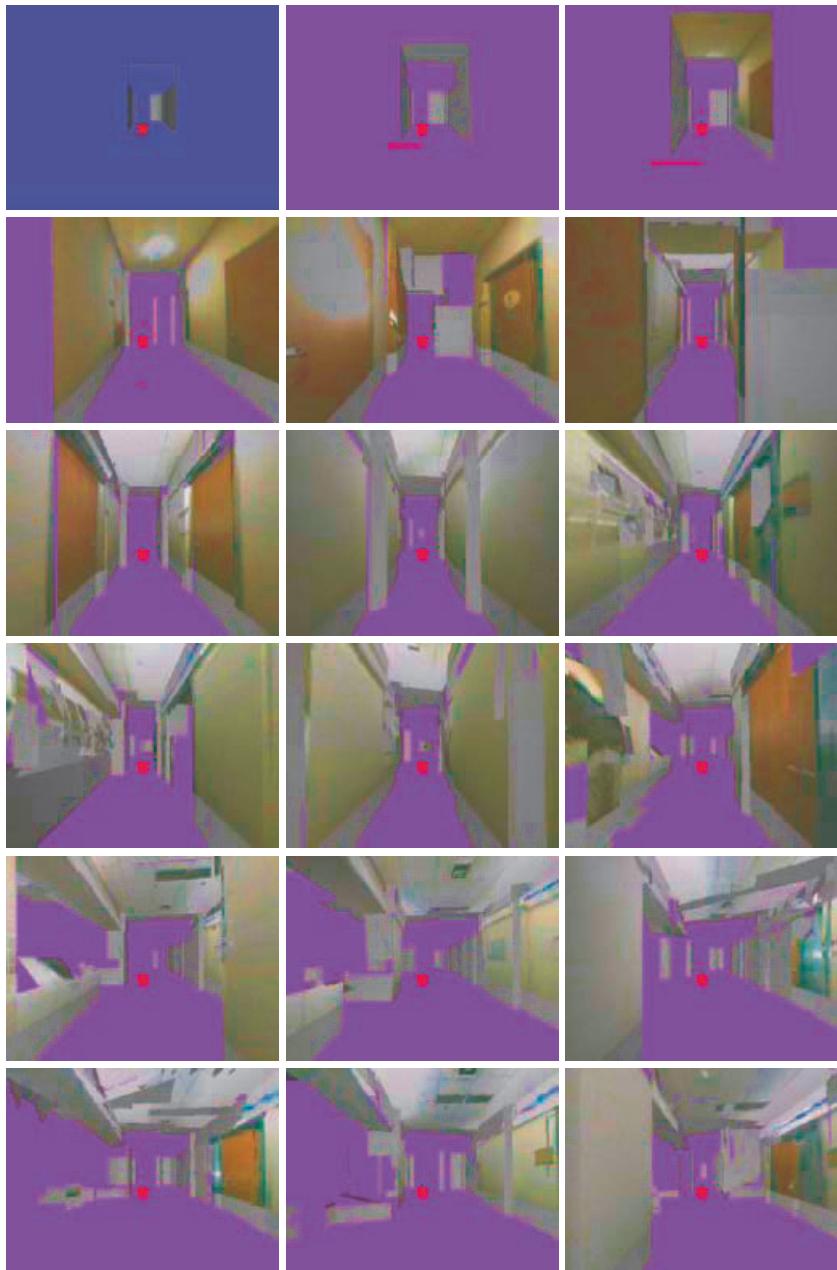


Fig. 6. Sequence of maps built during a real-time exploration demonstration in Stanford’s Robotics Laboratory, Gates Hall. All software components, including the texture visualization and robot navigation software, was executed in real-time with less than 0.5 seconds latency.

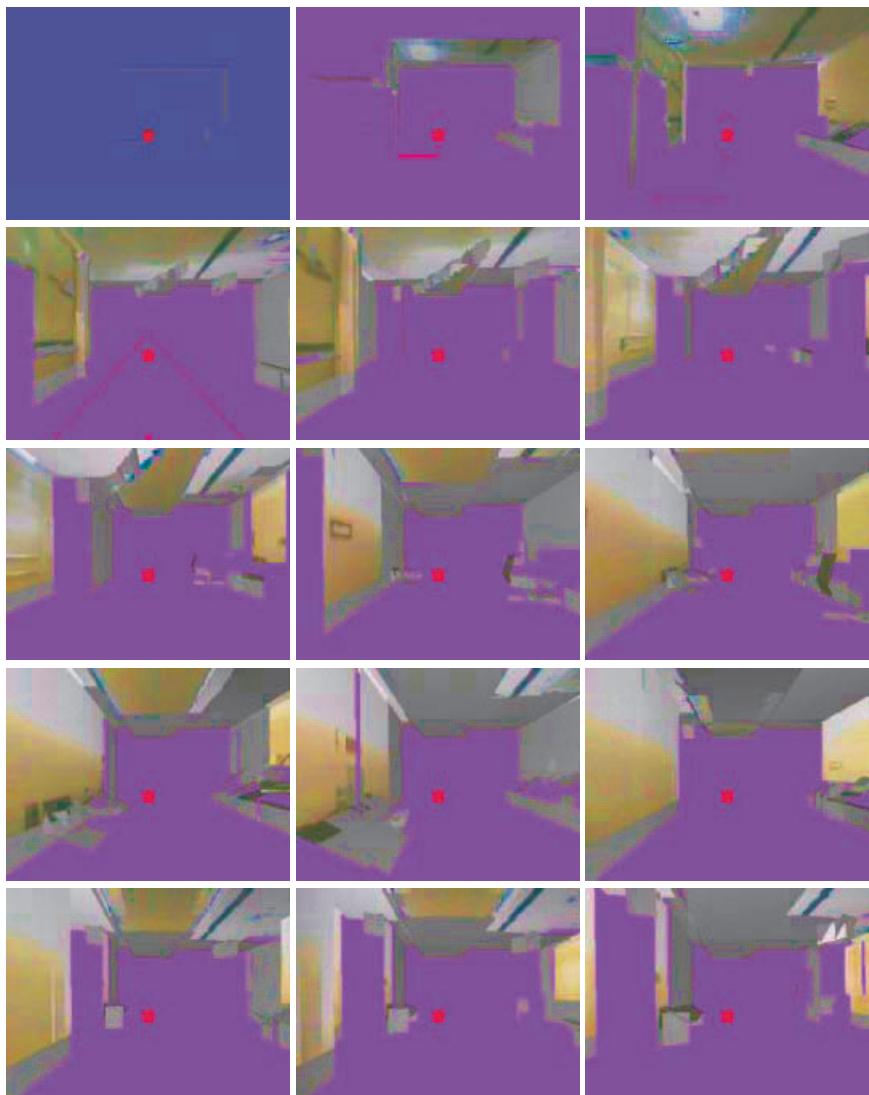


Fig. 7. Sequence of maps built *live* during a demonstration to the DARPA-MARS Program, in a meeting facility in Los Angeles. All software components, including the texture visualization and robot navigation software, was executed in real-time with less than 0.5 seconds latency.

(a) Polygonal maps generated from raw data



(b) Low-complexity multi-surface map

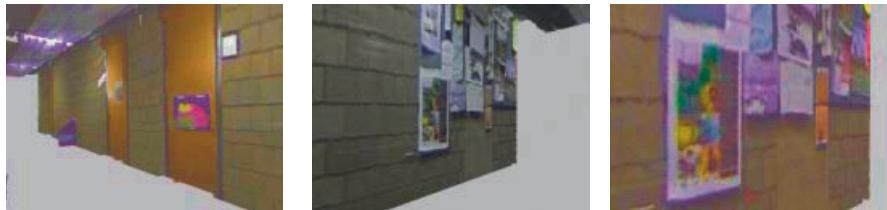


Fig. 8. 3D map generated (a) from raw sensor data, and (b) using EM, in which 94.6% of all measurements are explained by 7 surfaces. Notice that the map in (b) is much smoother and appears more accurate.

References

- Allen PK, Stamos I (2000) Integration of range and image sensing for photorealistic 3D modeling. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp 1435–1440.
- Bajcsy R, Kamberova G, Nocera L (2000) 3D reconstruction of environments for virtual reconstruction. In Proc. of the 4th IEEE Workshop on Applications of Computer Vision.
- Chatila R, Laumond JP (1985) Position referencing and consistent world modeling for mobile robots. In Proceedings of the 1985 IEEE International Conference on Robotics and Automation.
- Debevec PE, Taylor CJ, Malik J (1996) Modeling and rendering architecture from photographs. In Proc. of the 23rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH).
- Dempster AP, Laird AN, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.
- Elfes A (1987) Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265.
- Gutmann JS, Konolige K (2000) Incremental mapping of large cyclic environments. In Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA).
- Hähnel D, Schulz D, Burgard W (2002) Map building with mobile robots in populated environments. In Proceedings of the Conference on Intelligent Robots and Systems (IROS), Lausanne, Switzerland.
- Iocchi L, Konolige K, Bajracharya M (2000) Visually realistic mapping of a planar environment with stereo. In Proceedings of the 2000 International Symposium on Experimental Robotics, Waikiki, Hawaii.
- Li Y, Emery R, Chakrabarti D, Burgard W, Thrun S (2001) Using EM to learn 3D models with mobile robots. In Proceedings of the International Conference on Machine Learning (ICML).
- Lu F, Milios E (1997) Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349.
- Lu F, Milios E (1998) Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275.
- Martin C, Thrun S (2002) Online acquisition of compact volumetric maps with mobile robots. In IEEE International Conference on Robotics and Automation (ICRA), Washington, DC.
- Moravec HP, Martin MC (1994) Robot navigation by 3D spatial evidence grids. Mobile Robot Laboratory, Robotics Institute, Carnegie Mellon University.
- Shum H, Han M, Szeliski R (1998) Interactive construction of 3D models from panoramic mosaics. In Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR).
- Thorpe C, Durrant-Whyte H (2001) Field robots. In Proceedings of the 10th International Symposium of Robotics Research (ISRR'01), Lorne, Australia.
- Thrun S (2001) A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363.
- Thrun S (2002) Robotic mapping: A survey. In Lakemeyer G, Neberl B (eds) *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann. To appear.

Implementation of Simultaneous Navigation and Mapping in Large Outdoor Environments

Jose Guivant and Eduardo Nebot

Australian Centre for Field Robotics, University of Sydney, Australia

Abstract. This work addresses the real time implementation of Simultaneous Localization and Mapping (SLAM). It presents the integration if the Compressed Extended Kalman Filter (CEKF) and a new decorrelation algorithm to reduce the computational and memory requirements of SLAM to $\sim O(N * N_a)$, being N and N_a proportional to the total number of landmark in the global map and local area respectively. It also presents the problematic of outdoors navigation using natural feature based localization methods. The aspect of feature detection and validation is investigated to reliable detect the predominant features in the environment. Experimental results obtained in outdoor environments are presented.

1 Introduction

The problem of localization given a map of the environment or estimating the map knowing the vehicle position had been addressed and solved by using different approaches (Elfes, 1989; Guivant, Nebot, Baiker, 2000). A complete different problem is when both, the map and the vehicle position, are not known. It is well known that the two problems are correlated and can not be solved independently (Castellanos, Tardos, 1999). This problem is usually known as simultaneous localization and mapping (SLAM) and was originally introduced in (Smith, Self, Cheeseman, 1987). During the past few years significant progress has been made towards the solution of the SLAM problem. A number of different approaches have been presented to address this problem. In (Thrun, Fox, Bugard, 1998) a probabilistic approach is presented to solve the localization problem or the map building problem when the map or position of the vehicle respectively is known. This approach is based on the approximation of the probability density functions with samples, also called particles. This idea was originally introduced in (Gordon, Salmond, Simth, 1993) as a bootstrap filter but has been more commonly known as the particle filter. This approach has been applied very successfully to a number indoor navigation application, in particular in (Burgard, Cremers, Fox, Ahnel, Lakemeyer, Schulz, Steiner, Thrun, 1999). Due to the high computation requirements this method has not been used for real time SLAM yet, although work is in progress to overcome this limitation (Durrant-Whyte, Majumder, de Battista, Scheding, 2000). One of the most appealing approaches to solve the localization problem in real time is by modelling the environment and sensors and assuming errors with Gaussian distributions. Then very efficient algorithms, such as Kalman filters, can be used to solve this problem in a very compact and elegant manner (Maybeck, 1979). These methods can also be extended to perform SLAM. One of the main problems of the SLAM algorithms has

been the computational memory requirements. It is well known that the complexity of the SLAM algorithms can be reduced to the order of N^2 , N being proportional to the number of landmarks in the map. For long duration missions the number of landmarks will increase and eventually the on-board computer resources will not be sufficient to update the map in real time. This problem arises because the full map is correlated. The correlation appears since the observation of a new landmark is obtained with a sensor mounted on the mobile robot and obviously the error will be correlated with the uncertainty in the vehicle location and the other landmarks of the map. This correlation is of fundamental importance for the long-term convergence of the algorithm (Castellanos, Tardos, 1999), and needs to be maintained for the full duration of the mission. Leonard et. al. (Leonard, Feder, 1999), addressed the computational issues splitting the global map into a number of sub-maps, each with their own vehicle track. They present an approximation technique to address the update of the covariance in the transition between maps.

This paper makes use of the CEKF algorithm (Guivant, Nebot, 2001) to store and maintain all the information gathered in a local area and presents an algorithm to perform the full update with cost proportional to the number of landmarks N . Finally the issue of landmark detection in outdoor environments is addressed.

2 Simultaneous Localization and Mapping

The SLAM algorithm addresses the problem of a vehicle with known kinematics, starting at an unknown position and moving through an unknown environment populated with some type of features. The algorithm uses dead reckoning and relative observations to features in order to estimate the position of the vehicle and to build and maintain a navigation map as shown in Fig. 1. With appropriate planning the vehicle will be able to simultaneously navigate and build a relative map of the environment. If the initial position is known with respect to a global reference frame or if absolute position information is obtained during the navigation task then the map can be registered to the global frame. If not the vehicle can still navigate in the local area while exploring and incorporating new areas to the map.

Under the SLAM framework the size of the state vector for 2D environments is equal to the number of the vehicle states plus twice the number of landmarks, that is $2 * N + 3 = M$. In most SLAM applications the number of the vehicle states will be insignificant with respect to the number of landmarks. The number of landmarks will grow with the area of operation making the standard filter computation not suitable for on-line applications. In (Guivant, Nebot, 2001) it is shown that it is not necessary to perform a full SLAM update when working in a local area. With this approach the computational requirement of the SLAM becomes of the order of the number of features in the vicinity of the vehicle and independently of the size of the global map. This is strictly true when working in a local area. A common scenario is having the mobile robot moving in an area and observing features within this area. This may be due to the vehicle moving slowly or due to an external sensor with high frequency capabilities. Although high frequency external sensors are



Fig. 1. Navigation using SLAM. The vehicle builds a relative local map and localizes within this map using dead reckoning information and relative observations of features in the environment.

highly desired to reduce the position error growth, they will also introduce a high computational cost to the SLAM algorithm to process this information. For example a laser sensor can be used at frequencies of 4 to 30 Hz. Working with full SLAM will require to update M states at 30 Hz. While working in a local area observing a number of features the CEKF algorithm allows to preserve all the information processing a SLAM of the order of the number of features in the area. When this area is abandoned then the information acquired in this area can be transferred to the rest of the states without loss of information. This will allow incorporating high frequency external information with very low computational cost. It can be proved that the computational cost for each 'compressed' update will have a cost proportional to N_a^2 , with N_a the number of states representing the local landmarks. For more details on the derivation of the algorithm the reader is referred to (Guivant, Nebot, 2001). When the vehicle makes the transition to a new area a full update is required to transfer the information acquired in the previous area. The cost of the complete covariance matrix evaluation is approximately $N_a * N_b^2$. Provided that the vehicle remains for a period of time in a certain area, the computational saving will be considerable.

The local areas can be selected dividing the global map into rectangular regions with sides lengths at least equal to the range of the external sensor. When the car navigates in the region r the compressed filter considers that the active states group, X_A includes all the states related to landmarks that belong to region r and its neighbouring regions, as shown in Fig. 2. This implies that the local states belong to 9 regions, each of size of the range of the external sensor. The vehicle will be able to navigate inside the region using the CEKF and a full update will only be required when the vehicle leaves the central region r . The method will be extremely efficient

in applications with high frequency external sensors or where the vehicle navigates for long periods of time within local areas.

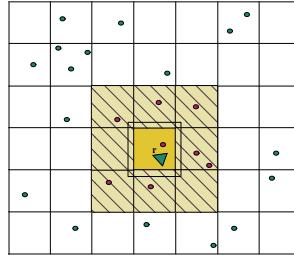


Fig. 2. Local area for the compressed Extended Kalman Filter (CEKF) algorithm.

3 Sub-Optimal Decorrelation Algorithm

The maintenance of the complete covariance matrix is important in cases where the cross-correlation between states is strong or at least not negligible. Any attempt to conservatively de-correlate a subset of states implies an increase in the value of some diagonal sub-matrices. If subsets of weakly correlated states are present then de-correlation of these states can be done with a small loss in the predicted estimated quality. Unfortunately when a map represents the landmarks in absolute form with respect to a single global frame, all the states are or tend to be strongly correlated. This correlation is of major relevance when a conservative de-correlation procedure is desired. Most of the state's high correlation coefficients are due to the map representation and not due to map estimation problem itself. An appropriate map representation that avoids this problem is the Relative Landmark Representation (RLR). This representation divides the map into sub-regions where the landmarks are defined respect to local coordinate frames. For the 2-D case each local frame is defined based on two local landmarks represented in global coordinate. The high correlation characteristic persists but only between the frame base landmarks and the vehicle states. These landmarks represent a small subset of the total landmark population. An attractive aspect of the RLR is that the cross-correlation between relative landmarks that belong to different frames (or constellations) tends to be extremely low, especially when these constellations are distant. Then a strategy that cancel the weakly cross-correlation terms and maintain the strong cross-correlation terms can be implemented. This sub-optimal simplification will generate results very close to optimal when the RLR representation is used. The algorithm to cancel the weakly cross-correlation terms in a consistent manner is now presented. Given a symmetric nonnegative definite matrix $P \geq 0, P \subset R^2$ it is possible to obtain a de-correlated

(diagonal) matrix $D \geq P$ according to

$$\begin{aligned} P &= \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} p_{11} + \kappa \cdot |p_{12}| & 0 \\ 0 & p_{22} + \frac{|p_{12}|}{\kappa} \end{bmatrix} - \begin{bmatrix} \kappa \cdot |p_{12}| & -p_{12} \\ -p_{12} & \frac{|p_{12}|}{\kappa} \end{bmatrix} = \\ &= D + \tau \leq \begin{bmatrix} p_{11} + \kappa \cdot |p_{12}| & 0 \\ 0 & p_{22} + \frac{|p_{12}|}{\kappa} \end{bmatrix} = D \quad \forall \quad \kappa > 0 \end{aligned} \quad (1)$$

This is true since τ is a nonnegative definite matrix:

$$\tau = \begin{bmatrix} \kappa \cdot |p_{12}| & -p_{12} \\ -p_{12} & \frac{1}{\kappa} \cdot |p_{12}| \end{bmatrix} \geq 0 \quad \forall \quad \kappa > 0 \quad (2)$$

In the general case it is possible to de-correlate the covariance matrices corresponding to two groups of states A and B using a similar technique:

$$P = \begin{bmatrix} \alpha & C \\ C^T & \beta \end{bmatrix} \quad \begin{array}{l} \alpha \in R^{n \times n} \\ \beta \in R^{m \times m} \\ C \in R^{n \times m} \end{array} \quad C = \begin{bmatrix} c_{11} & \dots & c_{1m} \\ \dots & \dots & \dots \\ c_{n1} & \dots & c_{nm} \end{bmatrix} \quad (3)$$

A conservatively de-correlated block diagonal matrix bound can be obtained for this matrix:

$$\begin{aligned} P &= \begin{bmatrix} \alpha & C \\ C^T & \beta \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} - \begin{bmatrix} 0 & -C \\ -C^T & 0 \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} - \begin{bmatrix} \tilde{\alpha} & -C \\ -C^T & \tilde{\beta} \end{bmatrix} + \\ &+ \begin{bmatrix} \tilde{\alpha} & 0 \\ 0 & \tilde{\beta} \end{bmatrix} = \begin{bmatrix} \alpha + \tilde{\alpha} & 0 \\ 0 & \beta + \tilde{\beta} \end{bmatrix} - \begin{bmatrix} \tilde{\alpha} & -C \\ -C^T & \tilde{\beta} \end{bmatrix} \leq \begin{bmatrix} \alpha + \tilde{\alpha} & 0 \\ 0 & \beta + \tilde{\beta} \end{bmatrix} \end{aligned} \quad (4)$$

$$\tilde{\alpha}, \tilde{\beta} \quad / \quad \tau = \begin{bmatrix} \tilde{\alpha} & -C \\ -C^T & \tilde{\beta} \end{bmatrix} \geq 0$$

The matrix τ will be nonnegative definite if the matrices α and β are formed with the following expressions:

$$\begin{aligned} \tilde{\alpha} \quad / \quad \tilde{\alpha}_{i,j} &= \begin{cases} \sum_{k=1}^m \kappa_{i,k} \cdot |c_{i,k}| & , \quad i = j \\ 0 & , \quad i \neq j \end{cases} \\ \tilde{\beta}/\tilde{\beta}_{i,j} &= \begin{cases} \sum_{k=1}^n \tilde{\kappa}_{i,k} \cdot |\tilde{c}_{i,k}| & = \sum_{k=1}^n \frac{1}{\kappa_{k,i}} \cdot |c_{k,i}| \\ 0 & , \quad i \neq j \end{cases} \\ \kappa_{i,k} &> 0 \quad \forall \quad i, k \end{aligned} \quad (5)$$

Equation 5 guarantees that the matrix τ will be, at least, nonnegative definite. Selecting $k_{i,k} = \tilde{k}_{i,k} = 1$ the coefficients α and β becomes:

$$\tilde{\alpha}_{i,i} = \sum_{k=1}^m |c_{i,k}|, \quad \tilde{\beta}_{j,j} = \sum_{k=1}^n |c_{k,j}| \quad (6)$$

A more appropriate selection of the family $\{k_{i,k}\}$ can be done considering the cross-correlations coefficients:

$$\kappa_{i,k} = \sqrt{\frac{\alpha_{i,i}}{\beta_{k,k}}} = \frac{\alpha_{i,i}}{\sqrt{\alpha_{i,i} \cdot \beta_{k,k}}}, \quad \tilde{\kappa}_{k,i} = \frac{1}{\kappa_{i,k}} = \sqrt{\frac{\beta_{k,k}}{\alpha_{i,i}}} = \frac{\beta_{k,k}}{\sqrt{\alpha_{i,i} \cdot \beta_{k,k}}} \quad (7)$$

Then α and β are evaluated:

$$\begin{aligned} \tilde{\alpha}_{i,i} &= \sum_{k=1}^m |c_{i,k}| \cdot \kappa_{i,k} = \alpha_{i,i} \cdot \sum_{k=1}^m |c_{i,k}| \cdot \frac{1}{\sqrt{\alpha_{i,i} \cdot \beta_{k,k}}} \\ \tilde{\beta}_{j,j} &= \sum_{k=1}^n |\tilde{c}_{j,k}| \cdot \tilde{\kappa}_{j,k} = \sum_{k=1}^n |c_{k,j}| \cdot \frac{\beta_{j,j}}{\sqrt{\alpha_{k,k} \cdot \beta_{j,j}}} = \\ &= \beta_{j,j} \cdot \sum_{k=1}^n |c_{k,j}| \cdot \frac{1}{\sqrt{\alpha_{k,k} \cdot \beta_{j,j}}} \end{aligned} \quad (8)$$

Finally the diagonal coefficients are updated:

$$\begin{aligned} \alpha_{i,i} + \tilde{\alpha}_{i,i} &= \alpha_{i,i} \cdot \left(1 + \sum_{k=1}^m |\mu_{i,k}| \right) \\ \beta_{i,i} + \tilde{\beta}_{i,i} &= \beta_{i,i} \cdot \left(1 + \sum_{k=1}^n |\mu_{k,i}| \right) \end{aligned} \quad (9)$$

with

$$\mu_{i,k} = \frac{c_{i,k}}{\sqrt{a_{i,i} \cdot b_{k,k}}} \quad (10)$$

If the states to be de-correlated have very low correlation then the correction terms $\sum_{k=1}^m |\mu_{i,k}|$ and $\sum_{k=1}^n |\mu_{k,i}|$ will be small. It has been shown in (Guivant, Nebot, 2001) that by using the RLR map representation the correlation between the states representing the relative landmarks that belong to different frames tends to be very small. It is also noted that the correlation between absolute states tends to be strong especially between states of close absolute landmarks. Then it is advantageous to design an algorithm that preserves the cross-correlation of any absolute state with any other state (relative or absolute) and ignores (de-correlate) any cross-correlation between two relative states associated to two relative landmarks that belong to different constellations (defined in different local frames). A normal EKF full SLAM performing this de-correlation in each update will result in excessively conservative results since over-bounding will be required in each update to de-correlate. This is not the case of the CEKF that requires only one full update after many local updates rendering in a less conservative over-bounding strategy. According to the CEKF a global update has to be done only when the vehicle abandons a local region. Assuming that the active states group includes all the absolute states and the local relative states then it implies that the passive states group comprehends all the relative landmarks states of the constellations where the vehicle is not navigating in. The local relative states are the states of the relative landmarks that belong to the constellation where

the vehicle is navigating in. In practice the most relevant information that the filter obtains is transmitted to the state corresponding to the vehicle pose, base landmarks and local relative landmarks states. It means that the improvement in the passive states covariance sub-matrix can be ignored in most cases without introducing significant errors. This procedure has a similar effect to adding some uncertainty to the passive states after the CEKF global update is done. Independently if the change in the passive states covariance sub-matrix is ignored or not, the cross-covariance between active and passive states has to be updated if no de-correlation is applied. Then the de-correlation will act over the relative states block of the active-passive cross-correlation sub-matrix terms. This imply that the memory and computation requirements will be $\sim O(N * N_a)$, assuming a constant number of landmarks N_a is used in each local region. Since $N_a \ll N$ the computation and memory requirement of the algorithm are dramatically reduced. The implementation of the strategy proceeds as follows: The complete optimal global update is done after n CEKF internal steps on the passive states:

$$\begin{aligned} k_2 &= k_1 + n \\ P_{ab,(k_2)} &= \phi_{(k_2)} \cdot P_{ab,(k_1)} \\ P_{bb,(k_2)} &= P_{bb,(k_1)} - P_{ba,(k_1)} \cdot \psi_{(k_2)} \cdot P_{ab,(k_1)} \\ X_{b,(k_2)} &= X_{b,(k_1)} - P_{ba,(k_1)} \cdot \theta_{k_2} \end{aligned} \quad (11)$$

In the sub-optimal CEKF the improvements in P_{bb} are ignored:

$$P_{bb,(k_2)} = P_{bb,(k_1)} - P_{ba,(k_1)} \cdot \psi_{(k_2)} \cdot P_{ab,(k_1)} = P_{bb,(k_1)} \quad (12)$$

The CEKF auxiliary matrix is still needed to evaluate the diagonal elements of P_{bb} to perform the de-correlation procedure. The objective is to transfer the new information to the states representing the vehicle pose, absolute landmarks and local relative landmarks, ignoring completely the covariance changes in the non-local relative landmarks states. From the viewpoint of the non-local relative landmarks, no change in their quality is obtained. In actual SLAM applications it can be observed that the cross-correlation factors between relative landmarks of different constellations have values of order 10^{-4} or smaller. This value becomes much smaller for distant constellations. This characteristic makes the conservative de-correlation close to optimal since very small virtual noise has to be added to the relative landmarks covariances to obtain the de-correlated matrix bounds. Further details on this approach are available in (Guivant, Nebot, 2002).

4 Environment Description and Feature Detection

The navigation map is built with features present in the environment that are detected by external sensors that provide information relative to the position of the vehicle. Recognizable features are essential for SLAM algorithms since they are responsible for bounding the navigation errors. One of the first tasks in the navigation system design is to determine the type of sensor required to obtain a desired localization accuracy in a particular outdoor environment. The most important factor that

determines the quality of the map is obviously the accuracy of the relative external sensor. For example, in the case of radar or laser sensors, this is determined by the range and bearing errors obtained when seeing a feature/landmark. These errors are function of the specification of the sensors and the type of feature used. If the shape of the feature is well known a-priori, such as the case of artificial landmarks, then the errors can be evaluated and the accuracy of the navigation system can be estimated. A different problem is when the navigation system has to work with natural features. The inspection of the environment can give an idea of the most relevant features that can be detected with a given sensor. The most appropriate sensor for the application will depend on the size of the operating area and environmental conditions. Figure 1 presents an outdoor environment where trees can be considered one of the most relevant features that a laser range sensor can identify. With larger areas or in environment with fog or dust a different sensor such as radar will be a better choice. Once the sensor is selected then a model to obtain accurate and consistent feature position estimation is required. For example, if the raw return from the laser is used as a measure of a distance to a tree then a significant error can be introduced due to the size, shape and inclination of the trunk. This problem is shown in Fig. 3 for various type of trees commonly found in these environments. Any algorithm designed to extract the location of these features needs to consider these problems to improve the accuracy of the feature location process. In this work a Kalman Filter was implemented to track the centre of the trunk by clustering a number of laser observations as representative of the circular surface of the trunk.



Fig. 3. Trees with different shape, size and inclination. The feature detection algorithm needs to consider these type of different trees to accurately determine the position of the feature.

4.1 Feature Position Determination

The landmark's position estimation can be improved by evaluating the diameter of the tree trunk. This will also make the observation information more independent of the sensor viewpoint location. The first stage of the process consists of determining the number of consecutive laser returns that belong to the cluster associated to an object, in this case a tree trunk. In the case of working with range and bearing sensors the information returned from a cylindrical objects is shown in Fig. 4. Depending on the angular and range resolution and beam angle, the sensor will return a number of ranges distributed in a semicircle. In Fig. 4 the cylindrical object is detected at four different bearing angles. An observation of the Diameter of the feature can be

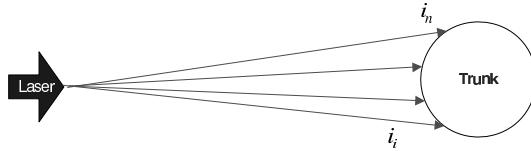


Fig. 4. Laser range finder return information from a cylinder type object

generated using the average range and bearing angle enclosing the cluster of points representing the object:

$$z_D = \Delta\beta \cdot r \quad (13)$$

Where $\Delta\beta$ and r are the angular width and average distance to the object obtained from the laser location. For the case of a laser returning 361 range and bearing observations distributed in 180 degrees:

$$\Delta\beta = (i_n - i_i) \cdot \frac{\pi}{NR}, \quad NR = 360, \quad r = \frac{1}{i_n - i_i + 1} \cdot \sum_{i=i_i}^{i_n} r(i) \quad (14)$$

The indexes i_n to i_i correspond to the first and last beam respectively reflected by the object. The measurement z_D is obtained from range and bearing information corrupted by noise. The variance of the observation z_D can then be evaluated:

$$\sigma_{z_D}^2 = G \cdot \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_{\Delta\beta}^2 \end{bmatrix} \cdot G^T, \quad G = \frac{\partial z_D}{\partial(r, \Delta\beta)} = [\Delta\beta \quad r] \quad (15)$$

$$\sigma_{z_D}^2 = (\Delta\beta)^2 \cdot \sigma_r^2 + r^2 \cdot \sigma_{\Delta\beta}^2$$

In outdoor applications the ranges are in the order of 3-40 meters. In this case we have that: $(\Delta\beta)^2 \cdot \sigma_r^2 \ll r^2 \cdot \sigma_{\Delta\beta}^2$, then $\sigma_{z_D}^2 \cong r^2 \cdot \sigma_{\Delta\beta}^2$. This fact indicates that the correlation between z_D and the range measurement error is weak and can be neglected. Additional noise ω_s is also included to consider the fact that this type of natural features will be in practice not perfectly circular and will have different diameter at different heights. Depending on the vehicle inclination two scans from the same location could generate a slightly different shape for the same object. The complete model with additional noise ω_s is:

$$\sigma_{z_D}^2 \cong r^2 \cdot \sigma_{\Delta\beta}^2 + \sigma_{\omega_s}^2 \quad (16)$$

Finally a Kalman filter to track each object is implemented assuming a process model with constant Diameter and initial condition generated with the first observation:

$$\begin{aligned} \dot{D} &= 0 \\ D(t_0) &= D_0 \\ \sigma_{D_0}^2 &= E\{D_0 \cdot D_0\} \neq 0 \end{aligned} \quad (17)$$

The Diameter of each feature is updated after each scan and then used to evaluate the range and bearing to the centre of the trunk. Figure 5 presents a set of experimental data points, the circumference correspondent to the estimated Diameter and the centre of the object estimated by the Kalman Filter after a few laser frames.

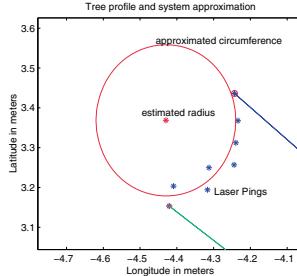


Fig. 5. Tree profile and system approximation. The dots indicate the laser range and bearing returns. The filter estimates the radius of the circumference that approximates the trunk of the tree and centre position.

5 Experimental Results

The navigation algorithms presented were tested in the outdoor environment of the type shown in Fig. 1 using a standard utility vehicle retrofitted with wheel and steering encoders and a laser range finder. The vehicle was started at a location with known uncertainty and driven in this area for approximately 20 minutes. Figure 6 presents the vehicle trajectory and navigation landmarks incorporated to the relative map. The system built a map of the environment and localized itself. The accuracy of this map is determined by the initial vehicle position uncertainty and the quality of the combination of dead reckoning and external sensors. The compressed algorithm was implemented using local regions of 30 meters as shown in Fig. 6. These regions are appropriate for the laser range sensor used in this experiment.

Figures 7 presents the position and heading difference obtained when using the optimal and sub-optimal approaches. It can be seen that the maximum discrepancy is smaller than 60 cm and a 0.2 degree respectively.

Finally Fig. 8 presents the correlation coefficient matrix. A black region represents the elements where the correlation was forced to zero. It can be seen that a large number of elements do not need to be maintained or stored using the proposed algorithm.

6 Conclusions

This paper presents an approach to reduce the computational requirement of SLAM algorithms from $\sim O(N^2)$ to approximately $\sim O(N * N_a)$, being N proportional

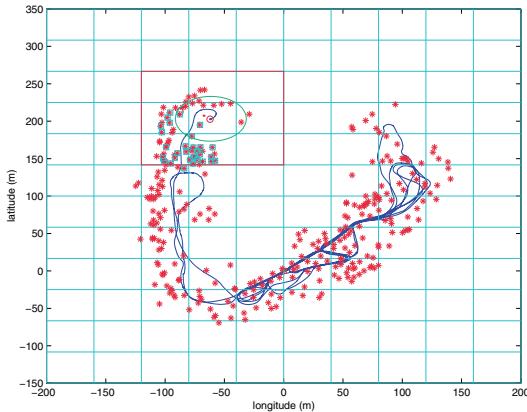


Fig. 6. Vehicle trajectory and navigation map obtained with the suboptimal algorithm

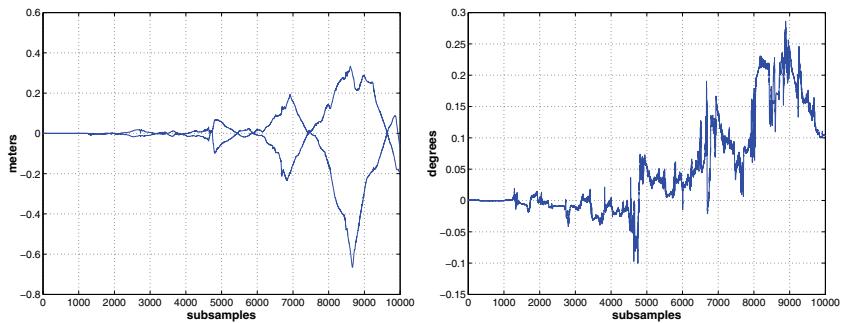


Fig. 7. Difference between full SLAM and conservative approach for latitude, longitude and heading estimation

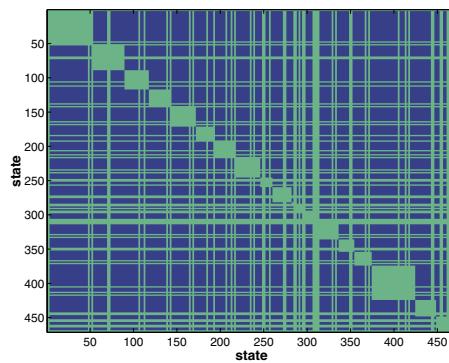


Fig. 8. Correlation coefficient matrix. The black elements correspond to the coefficients that were set to zero.

to the number of features incorporated to the map and N_a the states representing the landmarks in the local area. With this implementation the memory requirement are also reduced to order $N * N_a$. Since $N_a \ll N$ the computation and memory requirement of the algorithm are dramatically reduced. The complexity of feature detection was also presented with experimental results in an outdoor environment.

References

- Burgard W, Cremers AB, Fox D, Ahnel DH, Lakemeyer G, Schulz D, Steiner W, Thrun S (1999) Experiences with an interactive museum tour-guide robot. *Art. Int.*, 114(1-2): 3–55.
- Castellanos JA, Tardos JD (1999) Mobile robot localization and map building: A multisensor fusion approach. Kluwer Academic, Boston.
- Durrant-Whyte H, Majumder S, de Battista M, Scheding S (2000) A bayesian algorithm for simultaneous localisation and map building. In ISRR, USA.
- Elfes A (1989) Occupancy grids: A probabilistic framework for robot perception and navigation. PhD thesis, Department of Electrical Engineering, Carnegie Mellon University.
- Gordon NJ, Salmond DJ, Simith AFM (1993) Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113.
- Guivant J, Nebot E (2001) Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transaction of Robotics and Automation*, 17:242–257.
- Guivant J, Nebot E, Baiker S (2000) Localization and map building using laser range sensors in outdoor applications. *Journal of Robotic Systems*, 17(10):565–583.
- Guivant J, Nebot E (2002) Solving computational and memory requirements of feature based simultaneous localization and map building algorithms. <http://www.acfr.usyd.edu.au/publications>.
- Leonard JJ, Feder HJS (1999) A computationally efficient method for large-scale concurrent mapping and localization. In Ninth International Symposium on Robotics Research, pp 316–321, Utah, USA.
- Maybeck PS (1979) Stochastic models, estimation, and control, vol I. Academic Press, New York.
- Smith R, Self M, Cheeseman P (1987) A stochastic map for uncertain spatial relationships. In 4th International Symposium on Robotic Research. MIT Press.
- Thrun S, Fox D, Bugard W (1998) Probabilistic mapping of an environment by a mobile robot. In Proc. Of 1998 IEEE, pp 1546–1551, Belgium.

A Bayesian Algorithm for Simultaneous Localisation and Map Building

Hugh Durrant-Whyte¹, Somajyoti Majumder¹, Sebastian Thrun², Marc de Battista¹, and Steve Scheding¹

¹ Australian Centre for Field Robotics (ACFR) J04, The University of Sydney, Sydney NSW 2006, Australia,

² Department of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Abstract. This paper describes a full probabilistic solution to the Simultaneous Localisation and Mapping (SLAM) problem. Previously, the SLAM problem could only be solved in real time through the use of the Kalman Filter. This generally restricts the application of SLAM methods to domains with straight-forward (analytic) environment and sensor models. In this paper the Sum-of-Gaussian (SOG) method is used to approximate more general (arbitrary) probability distributions. This representation permits the generalizations made possible by particle filter or Monte-Carlo methods, while inheriting the real-time computational advantages of the Kalman filter. The method is demonstrated by its application to sub-sea field data consisting of both sonar and visual observation of near-field landmarks.

1 Introduction

Simultaneous Localisation and Mapping (SLAM) is a process by which a mobile platform can build a map of an environment and at the same time use this map to deduce it's location. In SLAM both the trajectory of the platform and the location of all landmarks are estimated on-line without the need for any *a priori* knowledge of location. SLAM is the subject of considerable current research activity (Disanayake, Newman, Durrant-Whyte, Clark, Csobra, 2000; Guivant, Nebot, Baiker, 2000; Leonard, Feder, 2000).

The complexity of the SLAM estimation problem is potentially huge (of dimension the number the number of landmarks). For these reasons, there has been a significant drive to find computationally effective SLAM algorithms. This has been achieved through the development and use of the Kalman and extended Kalman filter as the estimation algorithms of choice in SLAM algorithms. In these developments, simplification in the time update step and locality in the observation update step have resulted in algorithms that can process thousands of land-marks in real time on PC level architectures (Guivant et al., 2000).

However, the Kalman filter approach comes with a number of limitations. Most notably, the inability to represent complex environment or feature models, the difficulty of faithfully describing highly skewed or multi-modal vehicle error models, and the inherent complexity of the resulting data association problem. A parallel approach to vehicle navigation, which overcomes many of these limitations, is to consider navigation as a Bayesian estimation problem (Thrun, 2000b). In this method,

vehicle motion and feature observation are described directly in terms of the underlying probability density functions and Bayes theorem is used to fuse observation and motion information. Practically, these methods are implemented using a combination of grid-based environment modeling and particle filtering techniques. These methods have demonstrated considerable success in some challenging environments (Thrun, 2000a).

The work described in this paper was initially motivated by the session on navigation and subsequent discussions held at the 1999 International Symposium on Robotics Research (ISRR'99) (Hollerbach, Koditscheck, 2000). Of particular interest was an autonomous underwater vehicle (AUV) navigation example. In this case, the ability of the platform to undertake SLAM is essential as there is no *a priori* map, and no access to external reference (such as GPS) to provide position information for navigation. However, the AUV navigation problem is also characterised by very poor vehicle motion models, by highly unstructured and amorphous environments, and by poor quality sensor (sonar and visual) observations.

The fundamental approach taken in this paper is to agree that the Bayesian methodology is indeed the correct representational form for the sub-sea SLAM problem but then to seek a representation for the underlying probability distributions which yields a computationally efficient SLAM algorithm. In this paper, the Sum of Gaussians (SOG) method of describing general probability distributions is employed. This is a well known method of dealing with non-Gaussian probability distributions in an efficient manner (Alspace, Sorenson, 1972). The SOG method has substantial computational advantages in allowing Kalman filter-based methods to be extended to more general probability distributions with richer representational abilities.

2 Bayesian Formulation of the SLAM problem

2.1 Preliminaries

Consider a vehicle moving through an environment taking relative observations of a number of unknown landmarks using a sensor located on the vehicle. At a time instant k , the following quantities are defined:

- \mathbf{x}_k : The state vector describing the location and orientation of the vehicle.
- \mathbf{u}_k : The control vector, applied at time $k - 1$ to drive the vehicle to a state \mathbf{x}_k at time k .
- \mathbf{m}_i : A vector describing the location of the i^{th} landmark whose true location is assumed time invariant.
- \mathbf{z}_{ik} : An observation taken from the vehicle of the location of the i^{th} landmark at time k . When the specific landmark is not relevant to the discussion, the observation will be written simply as \mathbf{z}_{ik} .

In addition, the following sets are also defined:

- The history of vehicle locations: $\mathbf{X}^k = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k\} = \{\mathbf{X}^{k-1}, \mathbf{x}_k\}$.

- The history of control inputs: $\mathbf{U}^k = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\} = \{\mathbf{U}^{k-1}, \mathbf{u}_k\}$.
- The set of all landmarks: $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n\}$.
- The set of all landmark observations: $\mathbf{Z}^k = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\} = \{\mathbf{Z}^{k-1}, \mathbf{z}_k\}$.

2.2 Definition of the SLAM problem

In probabilistic form, the Simultaneous Localisation and Mapping (SLAM) problem requires that the probability distribution

$$P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) \quad (1)$$

be computed for all times k . This probability distribution describes the joint posterior density of the landmark locations and vehicle state (at time k) given the recorded observations and control inputs up to and including time k together with the initial state of the vehicle.

In general, a recursive solution to the SLAM problem is desirable. Starting with an estimate for the distribution $P(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{Z}^{k-1}, \mathbf{U}^{k-1})$ at time $k-1$, the joint posterior, following a control \mathbf{u}_k and observation \mathbf{z}_k , is to be computed using Bayes Theorem. This computation requires that a state transition model and an observation model are defined describing the effect of the control input and observation respectively.

The observation model describes the probability of making an observation \mathbf{z}_k when the vehicle location and landmark locations are known, and is generally described in the form

$$P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) \quad (2)$$

It is reasonable to assume that once the vehicle location and map are defined, observations are conditionally independent, and depend only on the map and the current vehicle state so that

$$P(\mathbf{Z}^k | \mathbf{X}^k, \mathbf{m}) = \prod_{i=1}^k P(\mathbf{z}_i | \mathbf{X}^k, \mathbf{m}) = \prod_{i=1}^k P(\mathbf{z}_i | \mathbf{x}_i, \mathbf{m}). \quad (3)$$

The motion model for the vehicle can be described in terms of a probability distribution on state transitions in the form

$$P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k). \quad (4)$$

That is, the state transition may reasonably be assumed to be a Markov process in which the next state \mathbf{x}_k depends only on the immediately proceeding state \mathbf{x}_{k-1} and the applied control \mathbf{u}_k , and is independent of both the observations and the map. With these definitions and models, Bayes Theorem may be employed to define a recursive solution to (1).

2.3 Formulation of the SLAM Problem

To derive a recursive update rule for the vehicle and map posterior, the chain rule of conditional probability is employed to expand the joint distribution of vehicle state, map and observation in terms of the vehicle and map

$$P(\mathbf{x}_k, \mathbf{m}, \mathbf{z}_k | \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) = P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) P(\mathbf{z}_k | \mathbf{Z}^{k-1}, \mathbf{U}^k) \quad (5)$$

and then in terms of the observation

$$P(\mathbf{x}_k, \mathbf{m}, \mathbf{z}_k | \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) = P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) P(\mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \quad (6)$$

where the last equality employs the assumptions established for the sensor model in (2). Equating (5) and (6) and rearranging gives

$$P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) = \frac{P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0)}{P(\mathbf{z}_k | \mathbf{Z}^{k-1}, \mathbf{U}^k)} \quad (7)$$

The denominator in (7) is independent of either the map or current vehicle state and can therefore be set to some normalising constant K . The total probability theorem can be used to rewrite the second term in the numerator in terms of the vehicle model and the joint posterior from time-step $k - 1$ as

$$\begin{aligned} P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) &= \int P(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{m} | \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) d\mathbf{x}_{k-1} \\ &= \int P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) P(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{Z}^{k-1}, \mathbf{U}^{k-1}, \mathbf{x}_0) d\mathbf{x}_{k-1} \end{aligned} \quad (8)$$

where the last equality follows from the assumed independence of vehicle motion from map and observations, and from the causality of the vehicle control input on vehicle motion. Equations (8) and (7) provide a recursive expression for the calculation of the joint posterior $P(\mathbf{x}_j, \mathbf{m} | \mathbf{Z}^j, \mathbf{U}^j, \mathbf{x}_0)$ for the vehicle state \mathbf{x}_j and map \mathbf{m} at a time j based on all observations \mathbf{Z}^j and all control inputs \mathbf{U}^j up to and including time j . The recursion is a function of a vehicle model $P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)$ and an observation model $P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m})$.

3 Solving the SLAM problem

There are a number of possible methods of solving (7). Here we briefly describe two methods at the extremes of the solution space: The Extended Kalman Filter (EKF), and the Particle filter or Monte-Carlo method. Subsequently, the Sum-of-Gaussian (SOG) method is proposed as an appropriate compromise between these two methods.

3.1 The Extended Kalman Filter (EKF)

The basis for the EKF is to describe the vehicle motion model in terms of a kinematic model subject to zero mean uncorrelated (Gaussian) errors in the form

$$P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \Leftrightarrow \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}, k) + \mathbf{w}_k, \quad (9)$$

where $\mathbf{f}(\cdot)$ models vehicle kinematics and \mathbf{w}_k motion disturbances, and to describe the observation model in terms of a geometric observation, again subject to zero mean uncorrelated (Gaussian) errors, in the form

$$P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) \Leftrightarrow \mathbf{z}(k) = \mathbf{h}(\mathbf{x}_k, \mathbf{m}) + \mathbf{v}_k, \quad (10)$$

where $\mathbf{h}(\cdot)$ describes the geometry of the observation and \mathbf{v}_k models observation error.

The key to the simplicity of the EKF solution is the fact that a product of Gaussian distributions is Gaussian, and the convolution of Gaussian distributions is also Gaussian. Thus, when the vehicle model is assumed Gaussian, the prediction stage (the integral or convolution in (7)) yields a Gaussian, and when the observation model is Gaussian, the update stage (the product in (7)) also yields a Gaussian. With these Gaussians described by their respective means and variances, the EKF proceeds to solve (7) in terms of only an estimated mean and covariance of the joint distribution (Guivant et al., 2000; Leonard, Feder, 2000).

The EKF however comes with many potential problems and limitations. The usual criticism that the non-linearity of vehicle motion and landmark observation models yield distributions that are not truly Gaussian is, in practice, rarely an issue. A more substantive problem is the difficulty of modeling natural environment features in a form that is amenable to use in an EKF. A second formidable problem is the fragility of the EKF method when faced with incorrect associations of observations to landmarks.

3.2 The Particle Filter Method

In contrast to the EKF, Particle filtering or Monte-Carlo methods aim to provide a complete representation of the joint posterior density using a large set of sample points, termed particles. These points provide a faithful approximation to the true shape of the full distributions employed. State propagation and observation models are also represented in the form of a sampled distribution. A number of related algorithms have been proposed to then fuse and propagate estimates in the general form of (7). Of note is the sampling importance resampling (SIR) or boot strap filter (Salmond, Gordon, Smith, 1993), the Monte-Carlo filter (Kitagawa, 1996) and the condensation algorithm (Isard, Blake, 1998).

Of particular relevance to this paper is the recent application of the particle filtering algorithm to indoor mobile robot navigation problems (Thrun, 2000a,b; Thrun, Fox, Burgard, 1998). In the case of localisation, a grid-based environment model is constructed to represent probabilistic landmark information. The vehicle motion is

also then described in probabilistic form. Successive observations are then used to compute a location posterior for the vehicle in terms of a set of samples or particles. Most importantly, this method overcomes a number of key problems with EKF localisation methods; data association (and the kidnapped robot problem) and severe non-linearities in vehicle motion.

Extending particle filter methods from localisation to map building is, however, complicated by the fact that the state space for the map is much larger than the state space for the vehicle alone. Sampling methods scale exponentially with state dimension and thus full posterior estimation, using particle filters for map building, is generally intractable.

3.3 The Sum of Gaussian (SOG) Method

One approach to reducing the computational complexity of Particle filter methods is to find a *functional* representation for probability distributions in the full Bayesian algorithm. The Sum of Gaussian (SOG) method provides such a representation. It is appropriate to note that the SOG method can approximate any distribution to an arbitrary (in probability) accuracy (Mazya, Schmidt, 1996). Further, SOG approximated Bayesian estimators achieve convergent optimality with respect to full Bayes estimators (Alspace, Sorenson, 1972; Ito, Xiong, 2000).

A general Gaussian is described in the form

$$\mathcal{G}(\mathbf{x}; \bar{\mathbf{x}}, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}})^T \Sigma^{-1} (\mathbf{x} - \bar{\mathbf{x}})\right), \quad (11)$$

where \mathbf{x} is the state to be estimated, and $\bar{\mathbf{x}}$ and Σ are the mean and variance characterizing the distribution. A Sum of Gaussians (SOG) or Gaussian mixture is described in the form

$$\begin{aligned} P(\mathbf{x}) &= \sum_{i=1}^n \alpha_i \mathcal{G}_i(\mathbf{x}; \bar{\mathbf{x}}_i, \Sigma_i) \\ &= \sum_{i=1}^n \frac{\alpha_i}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}}_i)^T \Sigma_i^{-1} (\mathbf{x} - \bar{\mathbf{x}}_i)\right) \end{aligned} \quad (12)$$

where α_i are a series of weights normally summing to 1. An example of a one-dimensional Gaussian sum is shown in Figure 1(a). The great advantage of using a SOG model is that the computations involved are straight-forward modifications of the standard EKF equations. This is clear from the properties of Gaussians and SOGs:

1. The product of two Gaussian distributions is Gaussian.
2. The product of two SOG distributions is a SOG distribution.
3. The convolution of two Gaussian distributions is Gaussian

4. The convolution of two SOG distributions is a SOG distribution.

1. is the well-known conjugate distribution property of a Gaussian.
2. Follows as multiplication distributes over addition.
3. Follows because the Fourier transform of a Gaussian is Gaussian and because convolution is multiplication in the frequency domain.
4. follows from the fact that convolution is a linear operator.

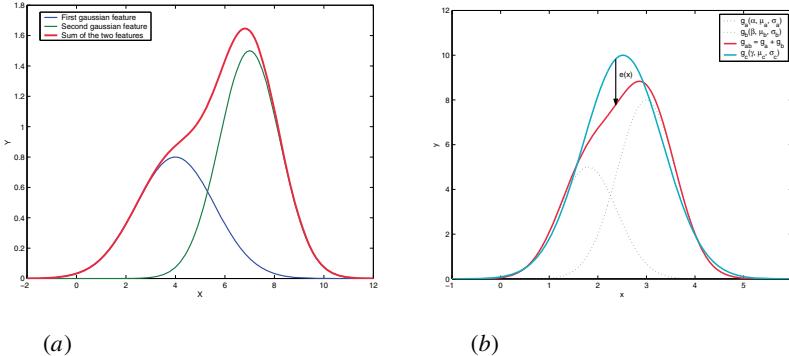


Fig. 1. Construction of SOG elements: (a) Sum of Gaussian model feature model (single dimension); (b) Re-sampling of SOG model to provide fewer sum elements.

There are two significant computational issues with SOG models. First, the product (or convolution) of two mixtures with (say) N_1 and N_2 Gaussians, results in a new mixture with $N_1 \times N_2$ Gaussians. Thus, the number of Gaussians used to model the density increases at each iteration of the estimation algorithm (7). Consequently, this requires re-sampling of the distribution as shown in Figure 1(b). The second significant problem is that in a Gaussian mixture the component functions are not orthogonal to each other (the SOG model does not form a basis for the density function).

Practically, the SOG method is implemented by first defining a SOG model for sensor observations and for vehicle motion. An observation model is defined in the form

$$P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) = \sum_{i=1}^{N_k} \alpha_{i,k} \mathcal{G}_i(\mathbf{z}_k; \bar{\mathbf{x}}_{i,k}, \mathbf{S}_{i,k}) \quad (13)$$

where the means $\bar{\mathbf{x}}_{i,k}$ and variances $\mathbf{S}_{i,k}$ are functions of the state \mathbf{x}_k and map \mathbf{m} which can be computed either from (10) or from a small sample approximation such as the Distribution Approximation Filter (Julier, Uhlmann, Durrant-Whyte, 2000). The motion model is defined in the form

$$P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) = \sum_{j=1}^{M_k} \beta_{j,k} \mathcal{G}_j(\mathbf{x}_k; \bar{\mathbf{x}}_{j,k}, \mathbf{P}_{j,k}) \quad (14)$$

where the means $\bar{\mathbf{x}}_{i,k}$ and variances $\mathbf{P}_{i,k}$ are functions of the previous state \mathbf{x}_{k-1} and control input \mathbf{u}_k computed from (9) or from a small sample approximation.

The SOG algorithm then proceeds to solve (7) by first computing the time update (a convolution) for each combination of Gaussian in the state motion model and prior distributions. This then yields a further SOG mixture which is subsequently multiplied by the SOG observation distribution to yield a posterior distribution which is also a SOG mixture. The resulting posterior must then generally be re-sampled to produce a lower complexity mixture. The convolution and multiplication steps in the SOG method are directly equivalent to the time and observation update equations in the EKF.

4 Implementation of the SOG Method in Sub-Sea SLAM

AUV navigation is a particularly appropriate application of the SOG method. The underwater environment is very unstructured; landmark features consist of rocks and other objects on the sea-bed. Motion models for vehicles are poorly understood, and the quality of sub-sea sensing is generally very low. Together this makes it notably difficult to apply conventional EKF methods to the navigation problem. However, real-time SLAM is essential in AUV applications. There is generally no map of a sub-sea domain, and there are generally no widely available positioning systems such as GPS to locate a vehicle.

4.1 AUV, Sensors and Sensor Modeling

Oberon, the AUV employed in this implementation, is shown in Figure 2(a) (Williams, Newman, G.Dissanayake, Durrant-Whyte, 2000). Oberon is equipped with three main sensors a 585 kHz or 1210 kHz (user selectable) pencil-beam sonar, a 675 kHz fan beam sonar, and a specially constructed underwater color camera. The fan-beam simply provides altitude information. The pencil-beam sonar and the camera are co-registered to provide terrain data for the AUV navigation system (Figure 2(b)). Together the sensors provide terrain data in the form of range and bearing, as well as other properties such as texture and colour. The pencil-beam sonar has a beam-width of 1.8° and is mechanically scanned in azimuth at a maximum rate of 180 degrees/second. A typical scan, registered to the ground plane is shown in Figure 2(c). At each selected position, the sonar pings and a complete amplitude wave-form is returned. In this work, a Sum of Gaussian (SOG) model is fitted to the amplitude return of this ping to provide a model of the return observation. The ping and two SOG models of differing resolution are shown in Figure 2(d).

4.2 Processing and Analysis of Field Data

The operational environment for this work is a reef inlet on the Pacific shoreline near Sydney with a maximum depth of 5m. During the trials described here, the AUV followed a course parallel to the shoreline of the inlet, followed by a similar

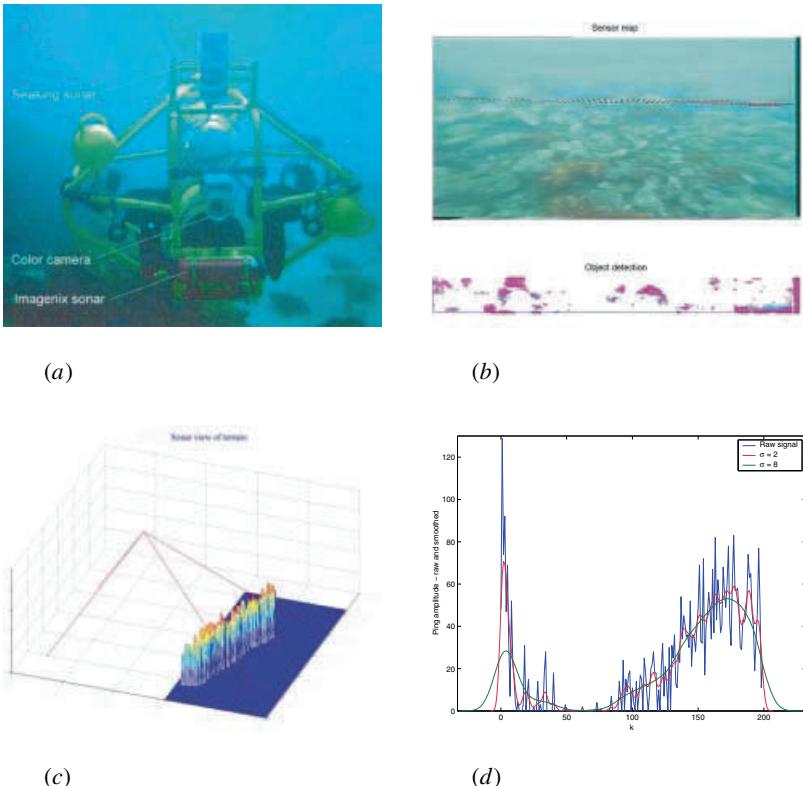


Fig. 2. Processing of experimental field data: (a) The experimental AUV, Oberon, showing location of the two sonars and vision system; (b) A typical data set showing sonar range data registered on camera image; (c) Projection of sonar data on to the ground plane, used for providing landmark data; (d) Fitting of Sum of Gaussian (SOG) distribution to a sonar ping (two possible resolutions shown)

course in the opposite direction. An approximate vehicle path is shown in Figure 3(a). The length of each run is approximately 20-25metres. The experimental data consists of time stamped sonar data and co-located camera data, along with other navigation data including altitude, depth, yaw rate, compass heading and propeller control input.

The sensor model for the sonar data is a SOG in the form of (13), an example of which is depicted in Figure 2(d). The motion model for the vehicle is taken to be a SOG derived from heading and thrust data in the form of (14). This is a relatively weak model, but in this instant suffices at the typical speed and update rate of the AUV. After each sonar scan (approximately every two seconds), a local SOG terrain map is obtained from data such as that shown in Figure 2(c). This map data is propagated through (7) using the AUV motion model. When new data is obtained, it is first converted into SOG form and multiplied by the propagated data to

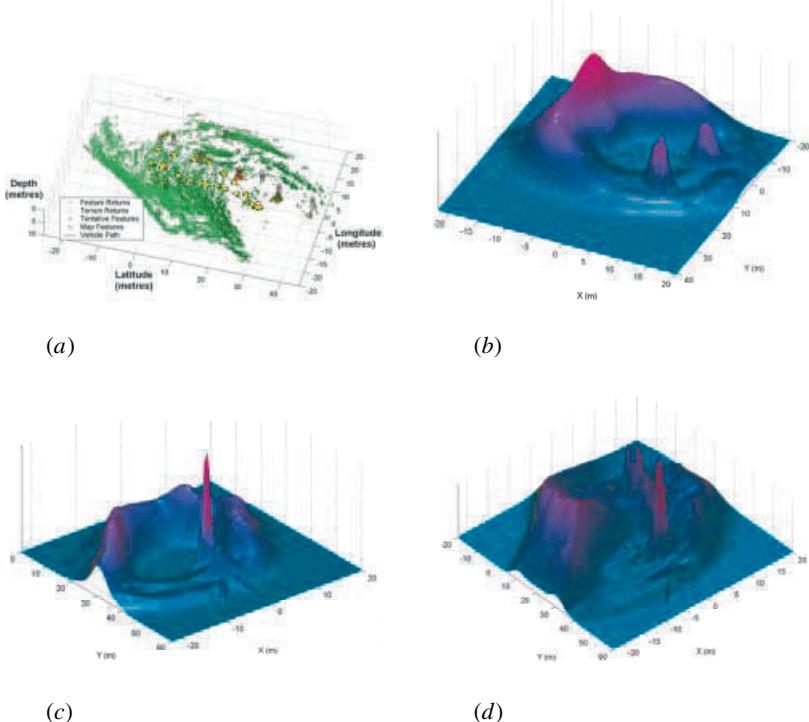


Fig. 3. Construction of a SOG terrain map from field trial: (a) Data set shown in the form of registered maximum returns together with a best computed vehicle path (Image courtesy Stefan Williams); (b) Constructed SOG terrain map after the first 100s of motion; (c) Constructed SOG terrain map after the 300s of motion;(d) Constructed SOG terrain map after completion of mission. Note that the large peaks in this image correspond to the large returns in the registered set (Figure (a) above). These are due to the placement of man-made dihedrals at these locations to allow a ground truth path to be estimated.

yield a posterior SOG. This SOG is then re-sampled before the next iteration of the algorithm.

Figure 3(b) shows the state of the SOG terrain model after the first 100 seconds of motion. The sides of the inlet in the neighbourhood of the vehicle are clearly visible. Also coming into view is the first of a set of man-made dihedrals. These are placed in the environment to provide a measure of ground-truth for the SLAM algorithm. Figure 3(c) shows the SOG model after 300 seconds of mission duration. The “SOG” nature of the terrain model is evident both in the surrounding reef structure and in the strongly reflective dihedral near the centre of the terrain image. Figure 3(d) shows the resulting SOG terrain map at the end of the mission. The reconstructed inlet structure should be compared to the overlay of points shown in Figure 3(a). The string of dihedrals positioned along the mission trajectory stand out as highly peaked Gaussians (these too can be compared to Figure 3(a)). There are

approximately 90 Gaussians in the re-sampled terrain. Re-sampling could be made both more or less stringent depending on the required application. The computational complexity involved is therefore equivalent to an EKF SLAM formulation with 90 landmarks.

5 Discussion and Conclusion

The terrain maps constructed using the SOG method are a robust and quite general representation of unstructured environments. The SOG method as applied to the SLAM problem also has the advantage of being computationally tractable and indeed can be implemented with many of the same rules that are employed in Kalman filtering.

There are a large number of questions and issues raised by the implementation of a full Bayes method in SLAM. Most importantly, is whether the full Bayes algorithm inherits the convergence properties of the Kalman filter-based SLAM methods (Dissanayake et al., 2000). This would require the terrain map to converge in probability monotonically and the platform location error to be bounded. A second issue is the incorporation of non-spatial feature descriptions as part of the map-building problem. A probability distribution provides a natural description of complex environment properties including texture, colour or reflectivity. It also seems natural to extend a density function to higher dimensions than simply landmark location; texture or colour for example. Another element of the general density modeling is to use external information, such as reports or constraints to provide additional ‘virtual’ sensor data (Stone, Barlow, Corwin, 1999). Finally, one major issue that seems to be resolved using the SOG method is that of data association. Normally, the problem of assigning measurements to landmarks is both complex and the main source of fragility in the navigation algorithm. In contrast, the SOG method deals with data association as a process in reinforcement or inhibition of density functions.

References

- Alspace D, Sorenson H (1972) Gaussian filters for nonlinear filtering problems. *IEEE Trans. Automatic Control*, 17(2):439–448.
- Dissanayake G, Newman P, Durrant-Whyte H, Clark S, Csobra M (2000) An experimental and theoretical investigation into simultaneous localisation and map building (SLAM). In Corke P, Trevelyan J (eds) *Experimental Robotics VI (ISER'99)*, pp 265–274. Springer-Verlag.
- Guivant J, Nebot E, Baiker S (2000) Localization and map building using laser range sensors in outdoor applications. *Journal of Robotic Systems*, 17(10):565–583.
- Hollerbach J, Koditscheck D (eds) (2000) *Robotics research, the ninth international symposium (ISRR'99)*. Springer-Verlag.
- Isard M, Blake A (1998) Condensation–conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28.
- Ito K, Xiong K (2000) Gaussian filters for non-linear filtering problems. *IEEE Trans. Automatic Control*, 45(5):910–927.

- Julier S, Uhlmann J, Durrant-Whyte H (2000) A new method for the non-linear transformations of means and covariances in filters and estimators. *IEEE Trans. Automatic Control*, 45(3):477–482.
- Kitagawa G (1996) A Monte-Carlo filter and smoother for non-Gaussian non-linear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.
- Leonard J, Feder H (2000) A computational efficient method for large-scale concurrent mapping and localisation. In Hollerbach J, Koditscheck D (eds) *Robotics Research, The Ninth International Symposium (ISRR'99)*, pp 169–176. Springer-Verlag.
- Mazya V, Schmidt G (1996) On approximate approximations using Gaussian kernels. *IMA J. Numerical Analysis*, 16(1):13–29.
- Salmond D, Gordon N, Smith A (1993) A novel approach to non-linear non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(1):107–113.
- Stone L, Barlow C, Corwin T (1999) Bayesian multiple target tracking. Artech House.
- Thrun S (2000a) Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999.
- Thrun S (2000b) Probabilistic algorithms in robotics. *AI Magazine*, 21(4).
- Thrun S, Fox D, Burgard W (1998) A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1):29–53.
- Williams S, Newman P, G.Dissanayake, Durrant-Whyte H (2000) Autonomous underwater simultaneous localisation and map building. In Proc. IEEE International Conference on Robotics and Automation (ICRA), pp 1793–1798, San Francisco, USA.

Part 2

Humanoid Robotics I

Session Summary

Hirochika Inoue

Chair

Dept. of Mechano-Informatics, School of Information Science and Technology, Univ. of Tokyo. 7-3-1, Hongo, Bunkyu-ku, Tokyo, 113-8656, Japan.

For a long time, the human-shaped robot has been considered the ultimate goal of robotic system development. One of the earliest humanoids was developed by Professor Kato and his group at Waseda University. Many other groups also attempted to build biped walking robots during the 1980s. In 1996, the announcement of a complete humanoid model “P2” by Honda opened a new era for humanoid R&D. Its well-designed appearance and stable walk gave a kind of shock to researchers who had attempted to build similar walking machines, and discouraged them to compete with Honda. However, at the same time, new challenges of humanoid R&D began. Japanese METI launched the Humanoid Robotics Project to provide technical infrastructures for developing a new humanoid platform and to seek practical applications of humanoid robots. Universities have joined in developing humanoid robots for fundamental research, while companies are introducing humanoid or home robots into the marketplace. Now, robots are not only objects to be built, but also tools to be used for studying humans and robots. We can view a humanoid as a human-shaped robot, but also, we can recognize the humanoid as novel computing machinery, which can not only compute, but also behave. We are obtaining a powerful new tool for the experimental study of behavior of humans and robots.

This session focuses on recent development of human-shaped robots and includes the following four papers. Kuniyoshi presents the ETL-Humanoid: a research vehicle for open-ended action imitation. Kagami introduces the University of Tokyo’s humanoid H6 and explains its low-level autonomy. Hirukawa reveals the software of the HRP project: “OpenHRP”, which was developed as an open architecture for humanoid robotics platforms. Lastly, Inaba reports a novel approach to humanoid hardware design that includes a spine structure as a backbone, and tendon drive actuation.

Imitation is a very familiar phenomenon to all of us humans. Children often imitate the behavior of their parents or the figures shown on TV. Adults also imitate others in more subtle ways. For example, when we attend a fancy dinner, we often take a glance at attendees sitting nearby, and imitate their manner so as not to offend the others. We also occasionally acquire the styles or habits of someone we respect or love, without being aware of it. Currently, the study of imitation is growing in a variety of fields, since imitation is related to, and binds together, the fundamental issues in human intelligence such as sensory-motor integration, category generation, contextual information processes, memory, learning, understanding of self and other agents, and social intelligence. The capability of action imitation constitutes a fundamental basis of higher human intelligence. Kuniyoshi first analyzes the concept

of imitation and discusses the essential problems underlying imitation. He takes a synthetic approach to understanding imitation capabilities, which requires a versatile humanoid robot platform. An overview of his group's full-body humanoid robot system is presented, along with their early experiments on behavior imitation with the humanoid. The robot can continuously interact with humans through visual, auditory and motion modalities in an unmodified everyday environment. For example, when a person approaches the robot showing a dual-arm motion, the robot spontaneously begins to copy it.

Kagami presents the low-level autonomy of the humanoid robots H6 and H7, which were developed by a group at The University of Tokyo. The low-level autonomy of a humanoid is divided into the following three categories: 1) walking functions such as online compensation of dynamic balance and online generation of walking trajectories, 2) manipulation functions such as arm motion planning and 3D vision based interactive planning, and 3) human interaction such as human identification and tracking, face recognition, voice recognition, and speech synthesis. For bipedal locomotion, an overview of an autonomous walking system is explained using a log-scale cycle-layered design. The layers include, online ZMP compensation, dynamically-stable walking trajectory generation with self-collision checking, and footstep sequence planning. For object manipulation, three software modules are discussed. They are: the auto-balancer for dynamically-stable standing motion generation, collision-free arm motion planning based on RRT (Rapidly-exploring Random Tree) method, and full-body dynamically-stable motion planning. For autonomy in human interaction, an experimental implementation of 3D-vision based human identification, face recognition software, and a natural language interface are explained. Integrating all of the above components, several interesting experiments are introduced.

Hirukawa presents an open architecture humanoid robotics software platform (OpenHRP for short) upon which various building blocks of humanoid robotics can be investigated. OpenHRP is a virtual humanoid robot platform with a compatible humanoid robot, and consists of a simulator of humanoid robots and a motion control library. OpenHRP also has a view simulator for humanoid robots with which humanoid vision software can be studied. The simulator is built on CORBA, which is a standard, interoperability software architecture. The real-time controller of the robot is run using ART-Linux, which is a real-time extension of Linux. Because of the unification of the controllers, and the consistency realized between the simulated and real robots, OpenHRP is a useful virtual platform for humanoid robotics on which various fundamental technologies can be developed. The virtualization of the platform is very important in order to inherit the software resources from one hardware instance to another very efficiently. Hirukawa stresses that OpenHRP will be a common software basis and is expected to become a standard software platform for humanoid research and development.

Inaba reports his novel challenge to build a humanoid with very complex physical structure and a massive parallel information network. The basic idea offers a new design scheme to realize whole-body limberness in a humanoid body. The de-

veloped humanoid “Kenta” has a flexible spine consisting of nine vertebrae, which form an S-curve spine like that of humans. The torso form is controlled by 40 motor-driven wires with tension sensors. The arm consists of two ball joints at its shoulder, an elbow joint, and a wrist ball joint, which are controlled by 14 wire-muscles. The leg consists of a hip ball joint, a knee joint, and an ankle ball joint, and is controlled by 10 wires. The neck is controlled by 6 wires. In order to sense and control a whole-body humanoid, 45 microprocessors (28 for 94 wire-muscles with tension sensors, 13 for 60 tactile sensors, 2 for acceleration sensors, 2 for communication hubs) are networked inside the body. Inaba stresses the importance of brain research for whole-body limberness of physically complex structures. Thus, he developed Kenta to implement evolutionary humanoid software on a massive parallel sensor-motor system like a human body.

ETL-Humanoid: A Research Vehicle for Open-Ended Action Imitation

Yasuo Kuniyoshi¹, Gordon Cheng², and Akihiko Nagakubo³

¹ Dept. of Mechano-Informatics, School of Information Science and Technology, The Univ. of Tokyo, Japan

² ATR Human Information Science Labs.

³ National Institute of Advanced Industrial Science and Technology [§]

Abstract. The capability of action imitation constitutes a fundamental basis of higher human intelligence. In this paper, we first analyze the concept of imitation and mark the essential problems underlying imitation, e.g. redundant sensory and motor degrees of freedom, adaptive mapping, and strong embodiment. It supports a synthetic approach to understanding imitation capabilities, which requires a versatile humanoid robot platform. An overview of our full-body humanoid robot system is presented, which is signified by its versatile physical capabilities and complete open architecture. Then we present our early experiment on multi-modal architecture and behavior imitation with the humanoid. It can continuously interact with humans through visual, auditory and motion modalities in an unmodified everyday environment. And when a person attends to the robot, starting to show a dual arm motion, the robot spontaneously starts to copy it.

1 Introduction

Imitation is a very familiar phenomenon to all of us humans. Children often imitate the behavior of their parents or the figures shown on TV. Adults also imitate others in more subtle ways. For example, when we attend an exotic styled dinner, we would take a glimpse of attendees sitting nearby, imitate their manner so that we do not offend the others. We also occasionally acquire the styles or habits of someone we respect or love, without being aware of it.

Despite its familiarity, we are quite far from scientific understanding of the nature of imitation, its cognitive process, underlying mechanisms, and effects on general cognition.

Probably due to popular negative implications of the word “imitation”, such as “fake” or “cheating”, very limited amounts of research have been devoted to the understanding of imitation until the last couple of decades. However, in the last several years, there is a sign of quickly growing interest to imitation in a variety of fields; animal ethology, cognitive psychology, brain science, intelligent robotics, complex adaptive systems science, economics, etc. This is due to the spreading understanding that imitation is related to, and bind together, the fundamental issues in human intelligence such as sensory-motor integration, category generation, contextual information processing, decision processes, memory, learning, understanding of self and other agents, and social intelligence.

[§] The present work has been carried out at Humanoid Interaction Laboratory, ETL, AIST.

In the future, intelligent robots will work in close proximity to humans in everyday life. If these robots do not consider the presence of humans and just pursue their static task goals, or do not develop new abilities after shipping, they will be annoying and maybe even useless to us. It will be very important that they understand the behavior of people around, being flexible enough to adapt, and learn how to behave and how to accomplish novel tasks.

Human behavior and task requirements change drastically depending on the given situation. Therefore, fixed response patterns will not serve the above purpose. A crucial functionality would be to interpret and respond to an open-ended variety of human actions. The robots must discover and learn such knowledge themselves. We call this issue open-ended developmental imitation.

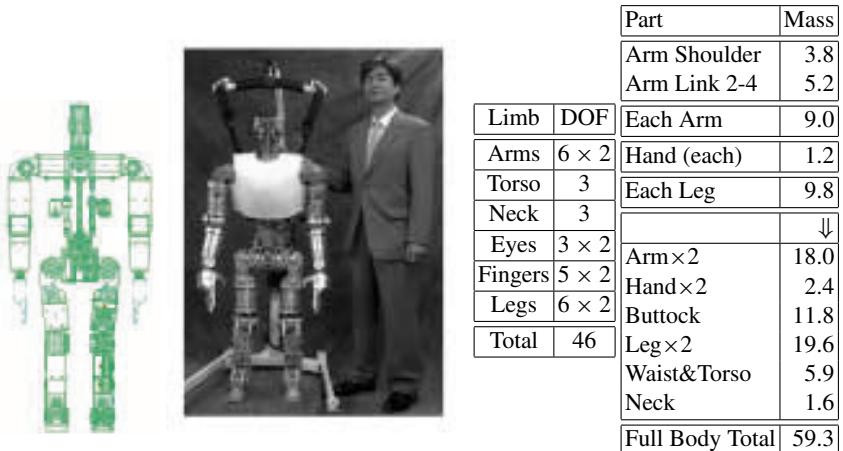


Fig. 1. ETL-Humanoid: The schematic figure, photo, joint configuration, and mass(kg) distribution of the mechanism.

2 Essential Issues in Imitation

Anyone would agree that “imitation” means “doing the same thing”. However, complication arises when we take one step further toward a detailed definition; so far there is no commonly agreed precise definition of imitation. Our stance is to focus on essential features which differentiate imitation from other traditional issues in intelligence research.

2.1 Do Mirrors Ape?

Submitting to the folk definition, “doing the same thing”, maybe a mirror should be classified as a perfect imitator? From our stance, this should be rejected. Then what is missing in a mirror? A mirror gives you a reversed image with no delay. However

these do not make a mirror image more remote from imitation than a video movie of yourself which is non-reversed and delayed. The problem with a mirror image is that it does not involve any interpretation or adaptation.

Here we state the first essential element of imitation: *Imitation is extracting only important information from the target behavior and reproducing them with self body and a given environmental situation.*

The cognitive process of imitation must involve some feature selection, interpretation, and adaptation in reproduction. A corollary is; *An exact copy is not imitation.* In other words, if there exists a static uniform mapping from the target motion to the resulting motion, it is not imitation.

Let us consider a more complicated example. Let us assume that we have a human subject seated on a chair. We place a infrared light beam sensor so that when the subject raises his/her right hand straight up, the beam is broken. We also place a motor-driven one-link mechanism which swings a bar from a horizontal posture to an upright posture whenever the controller receives a signal. We then wire the sensor signal to the controller. Then, whenever the subject raises the right hand, the “robot” also immediately raises its arm (you can even attach a wooden hand model onto the end of the motor-driven bar). Now, does this system really “imitate” the human subject’s behavior?

The problems with this system are the following:

1. A fixed mapping between perception and action: The stimulus-response mapping is completely fixed. There is no adaptation or categorization of varying data.
2. No redundant motor degrees of freedom: The simple robot is capable of only one type of fixed motion. This eliminates the possibility of adaptively changing its motion, even when we try to relax the constraint 1.
3. No redundancy in sensory information: The entire response relies on just a single bit of sensory information. There is no room for flexible interpretation of sensor information, even when we try to relax the constraint 1.
4. No context is considered: We can fake the system by inserting an arbitrary object into the infrared beam. The “robot” will raise the bar regardless of what the object is, or how it moved. According to a theoretical consideration about indexicality (Kuniyoshi, Inoue, 1993), in order to make sense, such a simple local perceptual information must be interpreted in combination with a correct context information which converts a complex classification problem into a simple low dimensional decision problem.
5. Weak embodiment: The bar-raising motion is trivial. There is no local constraints which the robot must fit by modifying its response motion.

2.2 Characteristics of Meaningful Imitation

The above considerations lead to the following list of important requirements for imitation to be meaningful as the research target in the context of intelligence research.

Redundancy: Both the motor and sensor degree of freedom should be highly redundant to allow variable sensory patterns and alternative instantiations of the target behavior. For this, it is important that these degrees of freedom are interchangeable, not assuming particular functional assignment to each.

Adaptation: The interpretation and action generation must be adaptive. In other words, “learning to imitate” is very important (Kuniyoshi, 1994). This is different from the common notion of “imitation learning” (Mataric, 1994; Miyamoto, Schaal, Gandolfo, Gomi, Koike, Osu, Nakano, Wada, Kawato, 1996) which focuses on the utility of the results of imitation.

Context dependency: The mapping from perception to action should rely on underlying context. Most of the past work rely on fixed implicit contexts (Hayes, Demiris, 1994). However, the context must be dynamically recognized by the system in order to cope with open-ended imitation.

Strong embodiment: The robot should have its own constraints to be met when it tries to reproduce the observed action. In the past, mapping over different arm kinematics (Kuniyoshi, Inaba, Inoue, 1994) and dynamics (Miyamoto et al., 1996), and environmental terrain shapes (Dautenhahn, 1995) have been addressed. But the whole body motion is not thoroughly investigated in this context.

Invariance within variation: This term was originally proposed by J. J. Gibson. Here it means that imitation maintains certain important features between the target and self, while other features change due to adaptation.

Complete versatile system: The above constraints together suggest a *complete system approach*, a well accepted notion in adaptive behavior research field. All of the past systems which were claimed to have imitation abilities were capable of only one kind of task. This does not meet the above requirements. We need to construct a complete agent with versatile task abilities in order to explore the essentials of imitation. From a theoretical point of view, what is important with imitation is that it is *not* a task-specific capability. This is also very clear from an application point of view; if the system can do only one task, imitation has no utility.

A first step to meet the above requirements is to build a robot with a high degree of freedom, with no fixed assignment of functionality to each, with multi-modal sensory responses, and continuous activity for getting involved in the contexts of interaction, and potential versatile task capabilities.

3 ETL-Humanoid: Our Research Vehicle

Past humanoid robots have mostly been designed for the reproduction of a particular task, or to replicate aspect(s) of a particular human skill. One of the first was WABOT-2, the piano playing robot of Wesada University, designed and built to read and play music (Kato, 1987). This was a great engineering effort in producing a system which performs a single highly skilled task.

WABIAN was one of the first full-size humanoid robots produced for research of bipedal locomotion, it is capable of walking, and recently dancing (Hashimoto et al., 1998). And more recently, a series of humanoid robot systems from Honda exhibited ground breaking performance in bipedal locomotion. These were primarily designed for walking, with capability of climbing stairs (Hirai, Hirose, Haikawa, Takenaka, 1998).

Aside from the biped research stream, the Cognobotics group at MIT have been developing human-like perceptual and motor subsystems for their humanoid, COG, in order to study human cognition (Brooks, Breazeal, Marjanović, Scassellati, Williamson, 1998). The Babybot Project at the univ. of Genova also addresses developmental cognitive robotics (Metta, Sandini, Konczak, 1999). So far, such approach mainly focused on interactions through vision and arm motion. More general capabilities with whole body actions are largely unexplored.

Recently, attempts are being made to add a wide variety of task abilities on top of the established biped locomotion capability (Inoue et al., 2000; Nishiwaki, Sugihara, Kagami, Kanehiro, Inaba, Inoue, 2000). A simplicistic interpretation of these is to put together the upper-body performances (e.g. hand-eye task modules) and the lower body performances (e.g. biped locomotion modules).

However, this divide-and-conquer approach may not cover all the human abilities. For example, human task performance is not limited to upright postures; we sometimes use all the limbs and even the trunk for locomotion, and sometimes use our leg or the head for object manipulation. As discussed in the next section, this kind of flexibility over high redundancy may be one of the key issues in adaptive human-robot interaction, especially, action imitation.

The following sections describe our effort in developing a multi-modal full-body humanoid system. Attempts have been made to build a physical structure similar to that of a person, its perceptual and motor response capabilities also have been geared to closely match that of humans. Sections 3.1 and 3.2 discuss the overall design of our system. Section 3.3, 3.4 and 3.5 provide the details of the physical robot system, designed to support full body humanoid interaction.

3.1 The overall design

We have been developing a complete humanoid robotic system, “ETL-Humanoid” (Fig. 1), or unofficially called “Jack”¹

Most of the design efforts are devoted to achieve generality and openness, assuming no particular task or posture. The robot possesses 46 degrees of freedom, with the height and weight of an average Japanese person. It was intended to be an experimental platform to explore a novel principle of controlling complex embodied systems characterized by exploitation of natural physical dynamics of the body, and sparse control at the boundaries of global structure of interaction dynamics (Kuniyoshi, Nagakubo, 1997a,b).

¹ The title and the primary figure in a Robin Williams movie – a story of a child having an adult body.

The idea of exploiting the physical interaction dynamics between a body and the environment is shared by a number of researchers, for example (Raibert, 1986). However, what we are interested in is the more general problem of identifying many different dynamics structures and navigating through them to achieve task goals, which is an open problem.

The design criteria is to keep the system's mobility range and strength as humanly as possible. Also the shape of the system is made to closely match that of a human. In order to achieve these objectives we also take into consideration compactness and modularity, while maintaining high power to weight ratio for the overall system. The details of the whole system will be presented in the following sections.

3.2 Design Criteria

In order to pursue the research framework stated so far, we need a humanoid system whose embodiment is as close as possible to humans. This leads to unique set of physical specifications (Nagakubo, Kuniyoshi, Cheng, 2000): 1) the joint torques must be able to support its own body weight (whole body, see Section 3.5); 2) the joints should be torque controllable and back drivable, 3) allow motion strategies to exploit inertia, such as, ballistic motion or dynamic motion, 4) the overall dimensions must be as close as possible to a small adult human (see Section 3.3); 5) it should have a smooth surface to allow arbitrary contact with surrounding objects; 6) the degrees of freedom and joint motion ranges should be close to humans to allow a broad range of motion.

In addition to these design considerations, the following criteria also needs to be taken into account, the overall system should be light in weight while keeping the requirement of high power. The overall mechanical system should be kept in a modular fashion, allowing ease of access and maintainability. Compactness will also need to be kept in order to keep overall proportion of the system's shape.

3.3 Mechanical configuration

The complete ETL-Humanoid robot system has 46 degrees of freedom: 12 d.o.f (6 each) for the arms, 3 d.o.f for torso, head and neck having 3 d.o.f, 6 d.o.f (3 each) for the eyes, 10 d.o.f (5 each) for fingers, and 12 d.o.f (6 each) for the legs (see Fig. 1).

As stated the robot is similar to the size of a average Japanese person, withholding technological constraints and trade-offs the system also has similar ranges of mobility as a person. Fig. 1 shows the complete design of the ETL-Humanoid, a photograph of the final system is also shown. The overall system is 1.6m in height, and weighs 59.3kg, Table 1 presents the weight distribution of the entire system. Table 1 shows the mobility range, power and speed for each joint of the system.

As discussed the complete system has 46 degrees of freedom, most of them are driven by synchronous AC servo motors. DC motors are used where space is limited, e.g. the forearm, the wrist, the eyes and the fingers. In each bending joint, the motor torque is transmitted via a belt to a harmonic drive which is fixed to the

Table 1. Full Mobility Range, Power and Joint Speed of ETL-Humanoid. Note: these mobility ranges have been imposed to simulate human like mobility ranges.

Joint	Motion Range [deg]	Redct. Ratio	Cont. Trq. [Nm]	Peak Trq.[Nm]	Max. Vel. [deg/sec]	Cont. Power [W]
Shoulder Sagittal	-40 – +160	50	33	99	486	134
Shoulder Frontal	-20 – +120	50	33	99	486	134
Upperarm	-100 – +50	50	20	60	858	152
Elbow	0 – +140	50	20	60	858	152
Lowerarm	-70 – +80	50	5	15	534	16
Wrist	-90 – +90	50	5	15	534	16
Pelvis Sagittal	-45 – +57	180	176	528	208	389
Pelvis Frontal	-48 – +48	180	176	528	208	389
Pelvis Rotation	-180 – +180	50	24	72	748	162
Hip Sagittal	-20 – +120	100	84	252	315	225
Hip Frontal	-20 – +50	100	52	156	273	118
Thigh Rotation	-45 – +110	100	36	108	324	98
Knee	-5 – +141	100	84	252	315	225
Ankle Sagittal	-40 – +45	100	52	156	273	118
Ankle Frontal	-23 – +23	100	25	75	585	126

limb structure via a joint torque sensor. In each rotational (around a longitudinal axis) joint, the motor is connected directly to a harmonic drive.

Keeping with the requirement of compactness, a custom network of small communication and control circuitries (nodes) are used to perform the motor controls, as well as sensor data collections. All of these nodes have been placed inside the mechanism in a distributed manner. The configuration of these control networks will be discussed further in the next section.

As it is shown in Figures 1 and 2, the kinematic configuration of the arms and the legs are very close to humans. The motors are distributed evenly over the whole body, in order to avoid additional weight and size due to complicated transmission mechanisms. This also provides a more proportional human-like shape and weight distribution. One example is the motor placement of the rotation around the longitudinal axis of the arms and the legs, where the motor has been placed within the rotating cylindrical parts.

The overall power to weight ratio of this system is enormous. For example, our arm system can support a weight of up to 4.2[kg] at the wrist in the worst posture (full arm extension) continuously, and of 18.2[kg] at the peak for several seconds. The ratio of payload and the self weight is 0.47 for the continuous action and 2.02 for the instantaneous action. This outperforms most of existing arm mechanisms. For example, the ratio for Mitsubishi PA-10 is about 0.3. The legs are even more powerful. The instantaneous action performances is particularly important in our research context.

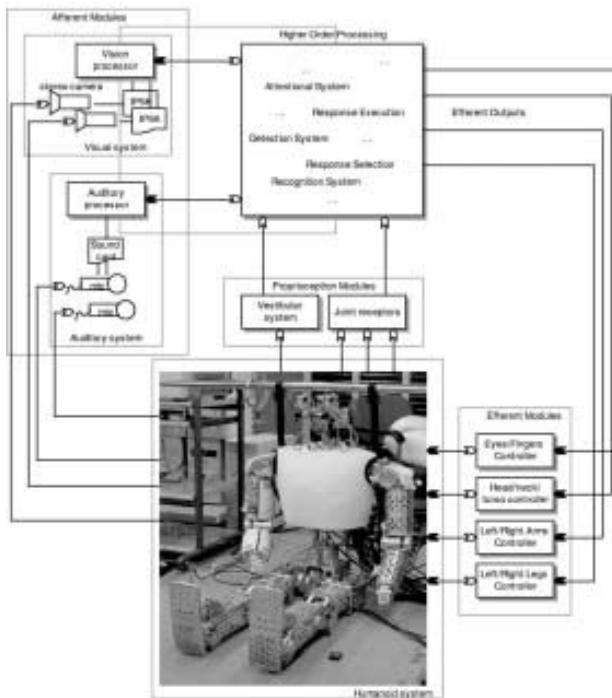


Fig. 2. Overview of Processing

3.4 Embedded Controllers

Fig. 2 shows an overview of the processing of our system. As discussed in our previous paper, the motor controls and sensor data collections are performed by a custom high speed network, designed specifically for our system (Nagakubo et al., 2000). This high speed network is made up of a number of custom communication circuitries (nodes), connected to a host computer forming the overall network.

As shown in Fig. 2, the system is divided into a number of sub-systems or modules. In the first version of our system, presented in (Nagakubo et al., 2000), we presented a cluster of 4 PCs for the controlling of the upper body of our system. In the latest version, we have been able to maintain the same number of overall processors used, however we can now control of the whole system. Therefore, gaining an overall reduction and reducing the space required, making it more compact for internal installation. The control is now divided as follows: 1 control network for both of the legs; 1 for the both arms, neck and torso; 1 for the head and eyes; and 1 for the fingers on both hands. This configuration provides us with a more compact solution for the overall integration of the control networks. Currently, the communication between each host controllers is conducted over ethernet. We expect to move to a IEEE-1394 network in the near future.

For communication and high-level motor control, we are using the Linux operating system at the host end of the network. The host is responsible for the updating of control parameters, data collections and acting as a bridge to the other computers on the host network. Currently the control and sensors data collection operates on a 10kHz cycle, providing low-level current, velocity, and positional control. The uniqueness of our system is that any of the motors can be driven in any of these modes, at any instant of time. For example, in our real-time mimicking experiment the motors were driven in a mix-mode manner in order to achieve human ballistic-like motion (Cheng, Kuniyoshi, 2000b).

One drawback of the Linux operating system, is that it does not provide support for hard real-time. In robotics community, RT-Linux is widely used. But it has many limitations which become problematic in scaling up to complex systems. We have developed a new real-time Linux called ART-Linux (Ishiwata, 1999), which is currently used in our humanoid control system. The ART-Linux incorporates modern real-time mechanisms which are missing in RT-Linux while maintaining complete compatibility for standard Linux applications.

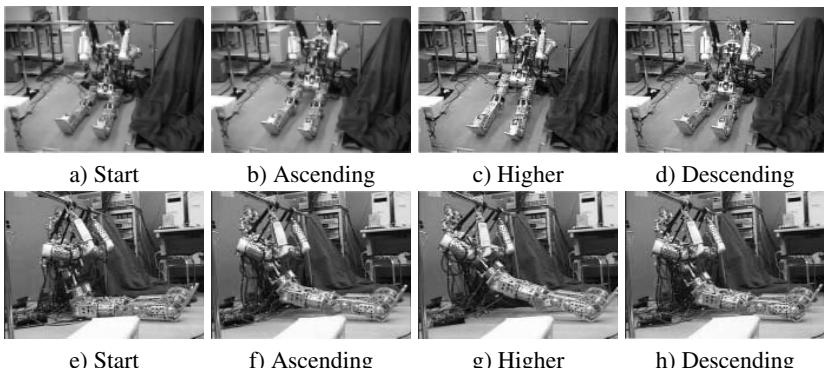
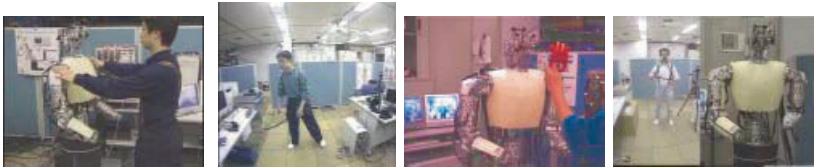


Fig. 3. Physical strength of our humanoid is demonstrated through this “Chin Up” exercise. The robot was able to physically support its body weight while rising its body up to the supporting bar. a–d) shows the frontal view, e–h) present the experiment from a side view. Note, the black strip is only used as a passive harness, act as a safety line to prevent the system from falling backward.

3.5 Physical Performance

One of the important design criteria of the system is that it should be able to support itself in a number of postures. Fig. 3 shows the strength of our humanoid, it is able to support itself while raising itself up to a supporting bar. The experiment demonstrate the attribute that the system is strong enough for supporting itself. The execution of the motion was conducted in a slow manner, the overall exercise lasting 21 seconds. This show that the system’s ability to support its own body weight



a) Physical Interaction b) Motion Flow c) Auditory Tracking d) Mimicking

Fig. 4. a) Physical Interaction b) Motion Detection, c) Auditory Tracking d) Real-Time Mimicking

for more than a quick instant, the system was in the air for at least 16 seconds during the “Chin Up”. The driving of the motor was based on torque control with positional information. The position information is only used in the two ways: as target(starting/ending) positions, and as a synchronizer (left and right arms). The maximum combined power consumption (peak) of the system in performing this posture was around 132.6 watts. Most of the power was asserted at the shoulders, elbows and hips.

Another experiment showing the system’s physical capability, involves the physical interaction of our system with a person. Figure 4a) shows this interaction, the experiments show that the system is able to support its own body, via its own torso. This experiment also shows that the system is capable of physical handling, thus being light-in-weight, backdrivable and compliant.

4 Multi-modal Perception and Action

Our goal is to investigate open-ended imitation abilities within the context of continuous multi-modal multi-dof interactions.

The following sections provide an overview of each subsystem of the perception action processing of our humanoid. Particular attention has been placed on vision processing, presented in Section 4.2. A discussion of humanoid motor control and the mapping between a person and our system is presented in Section 4.3. A brief discussion of the auditory system is presented in Section 4.4.

4.1 Sensor information processing components

For real-time stereo vision processing we are using a pair of Hitachi IP5005 vision processors, running the Linux operating system. The processing are performed at full video rate, 30Hz. Stereo auditory processing is done on a PC with a SoundBlasterTM sound card, also running the Linux operating system. The processing of the auditory information is done at 50Hz, whereas the sampling is performed at 44kHz.

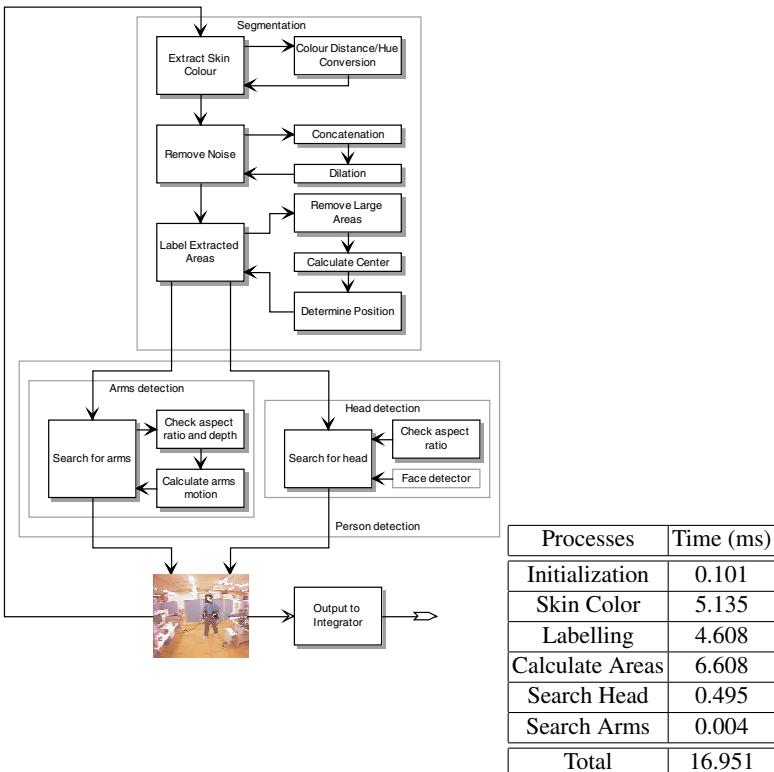


Fig. 5. Vision Processing: Stages and cycle time

4.2 Vision processing

In this section we describe the vision processes of the detection of a person, and the motion of the upper body of the person (head, left and right arm). The basic information that we are interested in is as follows: the position of the head in the scene and the motion of the arms. Figure 5 present the key stages of our vision processing.

The detection of the upper body of a person from the environment is based on visual segmentation of skin color from a given scene. The process of skin color detection is based on Color distance and Hue extraction. The overall process is as follows:

1. segmentation is made between the environment and the person.
2. select head, based on a set of attributes (e.g. color, aspect ratio, depth etc.)
3. extract the arms from the scene based on a set of attributes (e.g. color, aspect ratio, depth, etc.)

Position tracking of these extracted features is based on error filtering of the past locations. Once the segmentation and extraction have been performed, the motion

is then determined. The motion is calculated by taking the derivative of the visual information of the arms being tracked. The derivatives calculated for arms are then pass on to be used for motor control. The position of the head in the scene is used to orient the robot toward the person. Experimental results of this of our humanoid in a mimicking session with a person is presented in Section 5

4.3 Humanoid motion

The motor control of our system has been designed to be open, in the sense that at each joint, current(force), velocity, angular and position control has been provided. These schemes can be changed at any instance of system operation, allowing full freedom in the control of motion.

Humanoid-Human Mappings The current mapping between our humanoid and the person is done directly. The corresponding arm motion is mapped directly to the output of the corresponding humanoid motion. The control mappings are as follows, detected head motion controls the head, neck, and torso motion. The rotation of the torso and the head allows the humanoid to keep track of the person in the horizontal direction. While the neck moves in the vertical direction to ensure a full view of the person is seen. Each arm is mapped in the same way, the vertical motion of the detected arm is mapped to the motor joint at the elbow and at the shoulder (vertical – allows the shoulder to move in the forward/backward direction), the horizontal motion of the arm is mapped to the second motor on the shoulder (horizontal – allows the shoulder to move in the outward/inward direction).

These mappings allow a number of motion primitives to be produced, individual arm motion, up and down, and side to side. Some motions that have been realized based on these primitives include, swinging each arm in and out of phase; swing side to side while moving the arms up and down². As stated the behavior of the system has not been exactly specified, rather they emerge through interaction. As shown in Fig. 6, the system changes in a number of ways depending on activation signals³.

Motor output As suggested by human kinetic studies of human motion, human joint motion is spring like (Kelso, 1982; Schmidt, Lee, 1999). Based on this idea we adopted a scheme which produces a similar motion as humans. The velocity of each motor is given by Equation (1).

$$V_j(t) = V_{max} \frac{\log_{10}(U_i(t) + 1.0)}{2.0} \quad (1)$$

² a motion that we have found difficult to manually program.

³ more details are provided in (Cheng, Kuniyoshi, 2000a).

where j is the motor joint number. i is the index for each i_{th} output. $U_i(t)$ is the i_{th} output at instant t given by the magnitude of the motion. $V_j(t)$ is the velocity at j_{th} motor joint. V_{max} is the maximum velocity allowed for each motor joint.

In order to yield further human like ballistic motion, we selected to use a mixed control of velocity and current(force). This further allows the arm to be free and compliant, permitting physical interaction to be possible. The safety of the system is governed by a joint limiting watchdog process.

4.4 Spatial Hearing

Additionally, spatial hearing has also been incorporated into our humanoid system. The process of spatial hearing is based on Psychophysical studies of human hearing (Blauert, 1999). The processing is based on interaural differences, allowing the orientation of the sound source to be determined. Thus providing a way in which the sound source can be tracked. For more detail of our experiments using this processing see (Cheng, Kuniyoshi, 2000a).

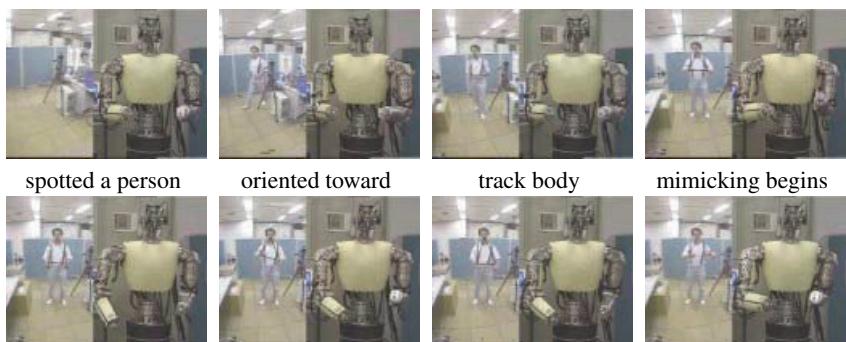


Fig. 6. This experiment demonstrates our humanoid robot mimicking the upper body motion of a person in real-time. At the same time, other parts of the system are also running, the vision as well as the auditory system are simultaneously processing the continuous array of stimuli, competing for its attention. In other instances, a loud noise from across the room could capture the attention of the system, shifting its focus towards the sound source.

5 Proto-Imitation Experiments

Experimental results of this of our humanoid in a mimicking session with a person are shown in Fig. 6. First, showing the orienting of our humanoid toward a person, tracking the person. Then once the two arms have been detected, the upper body motion of the person is then mimicked by our humanoid robot. The snapshots of this figure was taken from an experiment running continuously for around 4–5 minutes.

To demonstrate the robustness and seamlessness of our system, we have performed other experiments and demonstration lasting more than 20–30 minutes continuously. These experiments also include the interaction of auditory response and physical interaction.

Table 5 shows the average processing time for each of the key stages of our vision processing. On average the overall processing of a video image utilized 16.951 ms, allowing the system to keep within the 30Hz cycle of the video input signal.

6 Summary and Conclusions

In this paper we first analyzed the concept of imitation and extracted its essential characteristics. This suggested important requirements for a robotic system to be used as a research platform for imitation research. Then we presented a humanoid robot system that meet such requirements. A multi-modal action-perception system has been constructed for the humanoid, which exhibits real-time motion mimicking in the context of continuous multi-modal interaction with humans in an uncontrolled environment. The motion mimicking does not rely on full geometric model of a human body. Rather, it performs a mapping at a primitive level.

The presented mimicking functionality does not exhibit all the essential characteristics of imitation. However, this is the first real-time motion mimicking system which captures the multi-modal continuous interaction context. It sets a basis for further investigation of imitation as a pathway to humanoid intelligence.

Acknowledgments

We wish to acknowledge kindly the support of the COE program and the Brain Science program funded by the Science and Technology Agency (STA) of Japan and ETL, AIST. Also, numerous collaborators contributed to the development of our humanoid robot ⁴.

⁴ Yasuo Kuniyoshi is the leader of the project. He did the overall specification and conducted the system development. He designed the latest versions of the embedded control system, sensors, cabling, power systems, outer cover, and fine improvements of the mechanical parts. Akihiko Nagakubo did all the mechanical design of the core parts of body as well as the core idea of the novel control system architecture. Fuminori Saitoo did the design of the eye movement mechanism. Youichi Ishiwata designed the basic part of the embedded LAN protocol and also developed an original real time Linux kernel called ART-Linux. Gordon Cheng developed the entire control software and realized all the demonstrated robot behaviors. He also did all the hardware problem shooting, getting the entire system working. Much of the detailed design and fabrication were completed by a group of external contractors; Tokyo Seiki Co. did the core mechanism, Kyowa Dengyo Co. did the torque sensors, Komaden Co. did the installation and cabling of circuitry as well as the hand and the outer cover, Takenaka Seisakusho Co. did the control electronics.

References

- Blauert J (1999) Spatial hearing: The psychophysics of human sound localization. The MIT Press, revised edition. Second printing, ISBN 0-262-02413-6.
- Brooks RA, Breazeal C, Marjanović M, Scassellati B, Williamson MM (1998) The cog project: Building a humanoid robot. In IARP First International Workshop on Humanoid and Human Friendly Robotics, pp I–1, Tsukuba, Japan.
- Cheng G, Kuniyoshi Y (2000a) Complex continuous meaningful humanoid interaction: A multi sensory-cue based approach. In Proceedings of IEEE International Conference on Robotics and Automation, vol 3, pp 2235–2242, San Francisco, U.S.A.
- Cheng G, Kuniyoshi Y (2000b) Real-time mimicking of human body motion by a humanoid robot. In Proceedings of the 6th Intelligent Autonomous Systems, pp 273–280, Venice, Italy.
- Dautenhahn K (1995) Getting to know each other – artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16:333–356.
- Hashimoto S et al. (1998) Humanoid robots in Waseda University – Hadaly-2 and WABIAN. In IARP First International Workshop on Humanoid and Human Friendly Robotics, pp I–2, Tsukuba, Japan.
- Hayes G, Demiris J (1994) A robot controller using learning by imitation. In Borkowski A, Crowley JL (eds) Proceedings of the 2nd International Symposium on Intelligent Robotic Systems, pp 198–204, Grenoble, France. LIFTA-IMAG.
- Hirai K, Hirose M, Haikawa Y, Takenaka T (1998) The development of Honda humanoid robot. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation, pp 1321–1326, Leuven, Belgium.
- Inoue H et al. (2000) HRP: Humanoid robotics project. In Humanoids2000, Boston, USA. In Review.
- Ishiwata Y (1999) How to use Linux as a real-time OS. Interface, pp 109–118. (In Japanese).
- Kato I (1987) Wabot-2; autonomous robot with dexterous finger-arm. In Proceedings of IEEE Robotics and Automation, vol 5 of 2.
- Kelso JAS (ed) (1982) Human motor behavior: An introduction. Lawrence Erlbaum Associates, Publishers. ISBN 0-89859-188-0.
- Kuniyoshi Y (1994) The science of imitation – towards physically and socially grounded intelligence. Technical Report RWC TR-94001, Real World Computing Project, Tsukuba-shi, Ibaraki-ken, Japan.
- Kuniyoshi Y, Inaba M, Inoue H (1994) Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822.
- Kuniyoshi Y, Inoue H (1993) Qualitative recognition of ongoing human action sequences. In Proc. IJCAI93, pp 1600–1609. Morgan Kaufmann.
- Kuniyoshi Y, Nagakubo A (1997a) Humanoid as a research vehicle into flexible complex interaction. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97).
- Kuniyoshi Y, Nagakubo A (1997b) Humanoid interaction approach: Exploring meaningful order in complex interactions. In Proceedings of the International Conference on Complex Systems.
- Mataric MJ (1994) Learning motor skills by imitation. In Proc. AAAI Spring Symposium Toward Physical Interaction and Manipulation.
- Metta G, Sandini G, Konczak J (1999) A developmental approach to visually-guided reaching in artificial systems. *Neural Networks*, 12(10):1413–1427.

- Miyamoto H, Schaal S, Gandolfo F, Gomi H, Koike Y, Osu R, Nakano E, Wada Y, Kawato M (1996) A kendama learning robot based on bi-directional theory. *Neural Networks*, 9(8): 1281–1302.
- Nagakubo A, Kuniyoshi Y, Cheng G (2000) Development of a high-performance upper-body humanoid system. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00).
- Nishiwaki K, Sugihara T, Kagami S, Kanehiro F, Inaba M, Inoue H (2000) Design and development of research platform for perception-action integration in humanoid robot : H6. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00), pp 1559–1564, Takamasu, Japan.
- Raibert MH (1986) Legged robots that balance. The MIT Press. ISBN 0-262-18117-7.
- Schmidt RA, Lee TD (1999) Conditions of practice, chapter 11, pp 316–320. Human Kinetics, third edition. ISBN 0-88011-484-3.

Low-level Autonomy of the Humanoid Robots H6 & H7

Satoshi Kagami¹, Koichi Nishiwaki², James Kuffner², Kei Okada², Yasuo Kuniyoshi², Masayuki Inaba², and Hirochika Inoue²

¹ Digital Human Lab., National Institute of Advanced Industrial Science and Technology.
2-41-6, Aomi, Koto-ku, Tokyo, 135-0064, Japan.

² Dept. of Mechano-Informatics, School of Information Science and Technology, Univ. of Tokyo. 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan.

Abstract. This paper describes our research efforts aimed at developing several low-level autonomous capabilities required for remote-operation tasks involving humanoid-type robots. The low-level autonomy considered falls into three categories: 1) walking functions, such as online dynamic balance compensation and online walking trajectory generation, 2) manipulation functions such as arm motion planning and 3D vision based interactive planning, and 3) human interaction functions such as human identification and tracking, face recognition, and voice recognition/speech synthesis. We describe experimental results implemented on the humanoid robot research platforms H6 and H7.

1 Introduction

Recently, research on humanoid-type robots has become increasingly active, and a broad array of fundamental issues are under investigation (ex. (Brooks, 1996; Hirai, 1997; Kawamura, Wiles, Pack, Bishay, Barile, 1996; Murase, Sakai, Inaba, Inoue, 1998; Yamaguchi, Inoue, Nishino, Takanishi, 1998)). In particular, techniques for bipedal dynamic walking, soft tactile sensors, motion planning, and 3D vision continue to progress. However, in order to achieve a humanoid robot which can safely operate in human environments, not only the fundamental components themselves, but also the successful integration of these components will be required. At present, almost all humanoid robots that have been developed have been designed for bipedal locomotion experiments. In order to satisfy the functional demands of locomotion as well as high-level behaviors, humanoid robots require good mechanical design, hardware, and software which can support the integration of tactile sensing, visual perception, and motor control.

Autonomous behaviors are currently still very primitive for humanoid-type robots. It is difficult to conduct research on high-level autonomy and intelligence in humanoids due to the development and maintenance costs of the hardware. Thus, we have focused on remote-operation tasks in order to research the integration and development of low-level autonomy, while allowing a human operator to perform high-level task control and monitoring. Currently, the Humanoid Robotics Project (HRP:METI Japan) includes a sub-project for network-based operation of a humanoid robot (Inoue, Tachi, Tanie, Yokoi, Hirai, Hirukawa, Hirai, Nakayama,

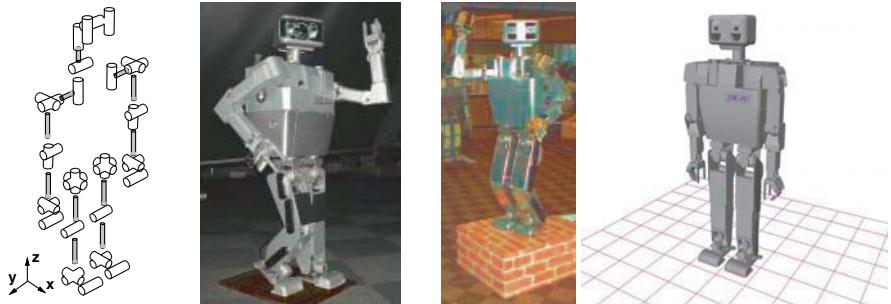


Fig. 1. H6 DOF arrangement and photo, H7 photo and 3D model

Sawada, Nishiyama, Miki, Itoko, Inaba, Sudo, 2000). However, rather than developing low-level autonomy, it strongly depends on sophisticated virtual reality technology and the availability of a skilled human operator in order to control the robot.

We believe low-level autonomous functions will be required in order to conduct research on higher-level autonomous behaviors for humanoids. Thus, low-level autonomy is useful not only for remote-operation applications, but also for ultimately progressing from low-level functions to high-level autonomous behavior in human environments.

In this paper, we describe several research experiments using the humanoid robots H6 & H7, which are controlled remotely. The ultimate goal is to achieve humanoids which are able to operate safely in human environments.

2 Low-level Autonomy of Humanoid Robots

We divide the requirements of low-level autonomy for network operated humanoid-type robots into the following three categories:

Bipedal Locomotion: The walking controller should be able to respond to high-level commands given by human operators, such as “walk forward”, or “walk to the kitchen”. The robot executes the action autonomously without relying on explicit low-level control by the user. In this paper, the design of a layered, autonomous walking software system is described. The layers include: 1) online ZMP compensation method, 2) dynamically-stable walking trajectory generation with self-collision avoidance, 3) footstep sequence planning.

Object Manipulation: Tasks derived from high-level commands, such as “pick up the hammer” are considered in this category. In this paper, three software modules are discussed: 1) dynamically-stable standing motion generator “autobalancer”, 2) RRT (Rapidly exploring Random Tree) based arm path planning 3) full-body, dynamically-stable motion planning.

Human Interface: Robot-human interaction should ideally be natural and require minimal effort from the user. We consider the interaction derived from human operators who are a) in front of the remotely-controlled humanoid robot, and b) in front of the remote operation software interface. High-level tasks such as “give the bottle to John” are considered, using experimental implementations of 1) stereo 3D vision based human identification, 2) face recognition software, 3) natural language interface and speech synthesis.

3 Humanoid Robots H6 & H7

Previously, the child-sized humanoid “H5” (H:1270mm, W:33kg) was developed as a prototype for investigating dynamic bipedal locomotion and dynamically-stable trajectory generation (Kagami, Kanehiro, Tamiya, Inaba, Inoue, 2000a; Nagasaka, Inaba, Inoue, 1999b; Nishiwaki, Nagasaka, Inaba, Inoue, 1999).

Subsequently, we developed the humanoids “H6” (H:1370mm, W:55kg) (Nishiwaki, Sugihara, Kagami, Kanehiro, Inaba, Inoue, 2000) and “H7” (H:1470mm, W:58kg) as a platform for researching vision/tactile and motion coupling behaviors (Fig. 1). Initially, dynamically-stable walking pattern generation, motion planning, and 3D vision functions were studied using H6 (Kagami, Nishiwaki, Kitagawa, Sugihara, Inaba, Inoue, 2000b; Kuffner, Nishiwaki, Kagami, Inaba, Inoue, 2001). The following two subsections describe the controller & software components of H7, which were extended from those of H6, since they share mostly a common architecture.

3.1 H7 Controller

Availability of processing resources is important for humanoid-type robots which simultaneously perform dynamic balance calculations, sensor processing (including 3D vision), motion planning, and other concurrent tasks. Currently, our robots use PC/AT clones suitable for industrial (FA) use, which exhibit both high performance and future extensibility.

H7 has dual PentiumIII-750MHz(100MHzFSB) processors with 256MB SDRAM. This board comes with a PICMG connector for backplane attachment, with one PICMG connector for the CPU board, two ISA connectors and three PCI connectors(Fig. 2).

- Mother board + backplane board
- Wireless ethernet(PCI)
- IEEE1394 image capture card(PCI)
- Sound card(PCI)
- Robot interface card RIF-01 ×2(ISA)

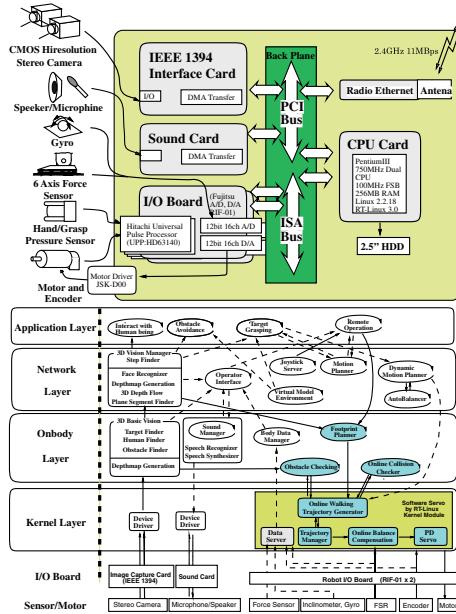


Fig. 2. H7 Controller & Software Components

3.2 H7 Software Components

Humanoid robot research platforms should provide a variety of fundamental software components ranging from low-level quick/smooth motion control to high-level vision/sensor based behaviors. In order to accomplish both low-level legged locomotion calculations as well as high-level behaviors, RT-Linux (Yodaiken, M.Barabanov, 2001) is adopted for interleaving hard-realtime tasks. Since Linux is not a real-time OS originally, RT-Linux has two special mechanisms: 1) a scheduler for real-time processes, and 2) a two-level interrupt handler. Currently, online stability control as well as walking trajectory generation is implemented as an RT-linux kernel module.

There are six primary software components in H7 (Fig. 2):

kernel motor servo, online ZMP compensation, online walking trajectory generation. Implemented as RT-Linux realtime module.

onbody stereo, voice input/output, obstacle avoidance, footprint planning, collision detection. Implemented as a linux user process.

network high-level vision, voice recognition, motion planning, autobalancer. Implemented on a network-distributed PC cluster.

application joystick based controller for walking, grasping, human-interface.

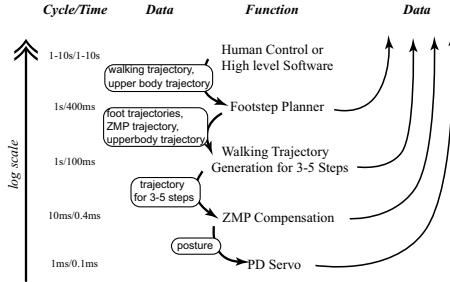


Fig. 3. Log-scale Cycle Layered Architecture for Bipedal Locomotion

4 Autonomy in Bipedal Locomotion

4.1 Layered Design of Bipedal Locomotion

One of the primary challenges to achieving autonomy in bipedal locomotion is to satisfy both balance & geometric constraints while simultaneously achieving a high-level goal. Since the lowest control servo-loop is running at a roughly 1[ms] cycle, it is unrealistic to attempt to implement everything inside this loop. Instead, a hierarchical architecture is needed.

A Log-scale cycle layered architecture is proposed to achieve this goal(Fig. 3). In this architecture, balance feedback using the ZMP is running at about 10[ms] cycle, while trajectory generation is running at about 1[s] cycle. The trajectory generation module generates 3 to 5 steps (or seconds) of trajectory. After generating a trajectory, stereo vision is used to confirm the target foot landing positions. The trajectory is also checked for any potential self-collisions between links of the robot using a convex-hull based conservative model of the robot. In this way, the output trajectory can be verified to satisfy both dynamic balance & geometric self-collision constraints (Fig. 2).

As the trajectory layer generates several future steps during each cycle, higher level software can perform environmental recognition and motion planning at a longer cycle time. The planning layer uses Plane Segment Finder output to determine appropriate foot landing locations, and uses a simple 2D open-space search to generate a walking target trajectory.

Currently, the highest level consists of a human operator. Since robots cannot yet decide high-level tasks on their own, we adopted remote-operation tasks for testing and experiments.

4.2 ZMP Compensation

For humanoid-type robots, it is difficult to “replay” directly dynamically-stable walking trajectories correctly in the real world. Even if a given trajectory satisfies ZMP constraints derived from an accurate model, errors and uncertainty in control and irregularities of the walking surface can cause a loss of balance. Therefore,

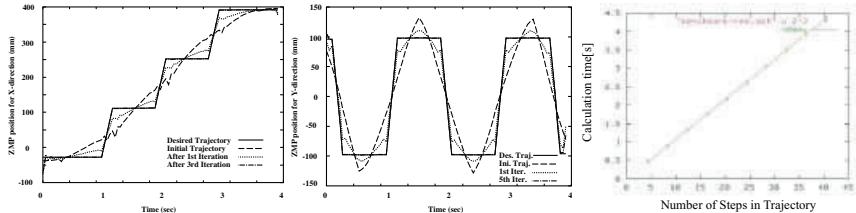


Fig.4. ZMP Trajectory Convergence (front and side direction), and Calculation Time vs. Number of Walking Steps

various local compliance control methods have been proposed (Honda Co. Ltd., 1993a,b, 1998; Nagasaka, Inaba, Inoue, 1999a). Currently, a torso position compliance method is adopted to track a given ZMP trajectory. This method attempts to follow a given ZMP trajectory by adjusting the horizontal motion of the torso online. It consists of two parts: one is a ZMP tracking mechanism, and the other is inverse pendulum control used to maintain dynamic balance (Online ZMP compensation in Fig. 2) (Kagami et al., 2000b).

4.3 Walk Trajectory Generation

For walking, it is impractical for users to control each DOF interactively. We have previously proposed a dynamically-stable trajectory generation method for humanoid robots (Kagami et al., 2000b). From a given input motion and a desired ZMP trajectory, the algorithm generates a dynamically-stable trajectory. A simplified robot model is introduced that represents the relationship between its center of gravity and ZMP. Horizontal shifts of the relative torso position can be used to effectively follow a given desired ZMP trajectory.

Let the z axis be the vertical axis, and the x and y axes define the sagittal and lateral planes respectively. Let the total mass of the robot be m_{total} , and the center of the gravity (COG) be $\mathbf{r}_{cog} = (r_{cog_x}, r_{cog_y}, r_{cog_z})$, and the total net external force be $\mathbf{f} = (f_x, f_y, f_z)$. The ZMP $\mathbf{p}_{cog} = (p_{cog_x}, p_{cog_y})$ around the point $\mathbf{p} = (p_x, p_y, h)$ on the horizontal plane $z = h$ is defined as the point where the moment is $\mathbf{T} = (0, 0, T_z)$. Then the following differential equation is obtained:

$$\mathbf{p}_{cog}^{err}(t) = \mathbf{r}_{cog}^{err}(t) - \frac{m_{total}r_{cog_z}(t)\ddot{\mathbf{r}}_{cog}^{err}(t)}{f_z^o(t)} \quad (1)$$

Here let \mathbf{p}_{cog}^{err} be the error between the ideal ZMP \mathbf{p}_{cog}^* and the current ZMP \mathbf{p}_{cog} . Let \mathbf{r}_{cog}^{err} be the error between the ideal center of gravity trajectory \mathbf{r}_{cog}^* and the current trajectory \mathbf{r}_{cog} .

The torso position is shifted in the horizontal plane in order to minimize these errors. An iterative numerical method is adopted to eliminate approximation errors arising from the simplified model (see (Nishiwaki, Kagami, Kuniyoshi, Inaba, Inoue, 2001) for details). Figure 4(left & center) shows the convergence of the initial

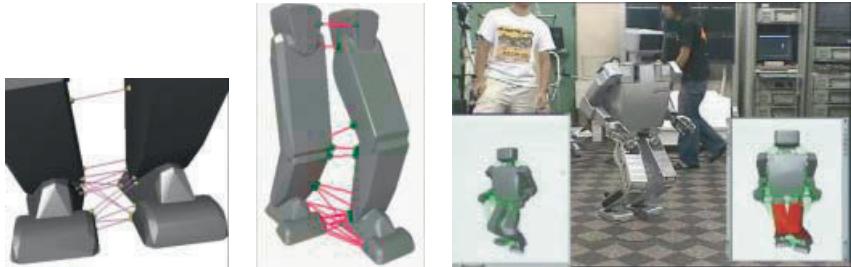


Fig. 5. Maintaining pairs of closest points between leg links (left & center) and Self-collision Checking (right) {lower left: full trajectory monitor window, lower right: minimum distance posture (colliding)}

ZMP towards the ideal ZMP trajectory and converged trajectories at each iteration step.

4.4 Self-Collision Checking

This algorithm is only heuristic, but it has the advantage of being simple enough to be used for online trajectory generation. Currently, it can calculate approximately 3 to 5 steps of stable trajectory within 100[ms](Fig. 4(right)). Thus, the motion of the robot can be updated every 100[ms]. However, our current system updates the robot motion on a 1[s] cycle since processing resources are required for other high-level software such as stereo vision and path planning.

Self-collisions occur when one or more of the links of a robot collide. Self-collisions can result in damage to the robot itself, or through a loss of balance or control, cause human injury or damage to its surrounding environment. Thus, detecting and avoiding self-collisions is fundamental to the development of robots which can be safely operated in human environments.

We have developed an efficient geometric approach to detecting link interference (Kuffner, Nishiwaki, Kagami, Kuniyoshi, Inaba, Inoue, 2002). We rely on fast, feature-based minimum distance determination methods for convex polyhedra (Mirtich, 1998). Minimum distances between selected pairs of link convex hull models is used in order to conservatively guarantee that a given trajectory is free of self-collision. Full body trajectories can be checked in advance for potentially self-colliding postures prior to being executed on the robot.

In our current implementation, the minimum distances between all possible relevant body link pairs for a 30 DOF humanoid (435 pairs) can be calculated in approximately 2.5[msec] on average on an 866 MHz Dual Pentium III. For typical walking trajectories, 19 pairs of links on the lower body need to be checked. Figure 5(left & center) illustrates closest points between pairs of lower body links. Before a trajectory is executed on the robot, it is tested for potential self-collisions. This can be done in approximately 20[msec] while running an online joystick control program (shown in Fig.5(right)).

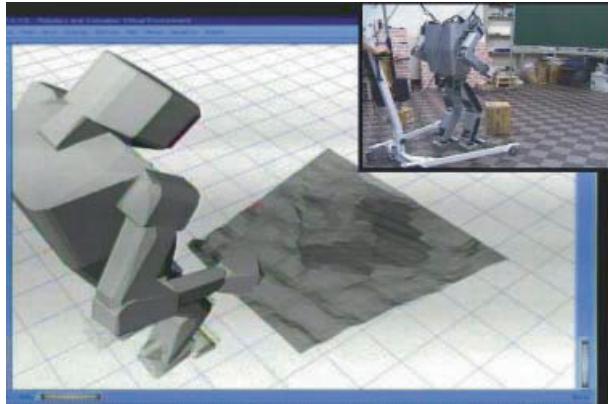


Fig. 6. Floor Object Model Derived from Stereo Vision

4.5 Verifying Walking Space

Stereo vision output can be used to help confirm potential foot landing positions. In order to generate accurate 3D vision information, we use a high resolution CMOS stereo camera system with an IEEE1394 connection are attached on the head (Videre Mega-D: 1280x1024 pixels, with a field of view of approximately 90[deg]). The accuracy of a single stereo image is about 1.5[cm] around the foot area at a range of about 1.7m (Fig. 6).

4.6 Plane Segment Finder

Obtaining the locations and orientations of planar surfaces is very useful in indoor or other artificial environments. In order to sense 3D planar surfaces, we proposed a Plane Segment Finder by combining depth map generation and 3D hough transformation (Kagami, Okada, Inaba, Inoue, 2000d). The process includes: 1) precise depth map generation, 2) 3D hough transformation for identifying plane segment candidates, 3) fitting the candidates to the depth map so that planar regions and non-planar regions can be distinguished, and 4) tracking segmented planar surfaces in order to maintain real-time planar segmentation. 3D planes are described using the following parametric notation:

$$\rho = (x_0 \cos(\phi) + y_0 \sin(\phi)) \cos(\theta) + z_0 \sin(\theta) \quad (2)$$

If ϕ, θ, ρ each have a discrete resolution of M , the algorithm has a computational complexity of $O(M^3)$ for voting each depth data element. In order to reduce this cost, we adopted a randomized hough transform method (Kälviäinen, Hirvonen, Xu, Oja, 1994). Instead of voting the entire ϕ, θ, ρ space for each depth point, local normal vectors are calculated corresponding to a single vote in ϕ, θ space. Then algorithm thus incurs a cost of $O(M)$ for voting each depth data element. Figure 7 shows an experiment in which planar surfaces of a set of stairs are discovered.



Fig. 7. Top: Plane Segment Finder Result, Center&Bottom: Climbing 20cm Step



Fig. 8. Online Joystick Control Walking Experiment. (Torso displacements: Pitch, Roll, Yaw and Sink)

4.7 Joystick Control Interface

Leveraging the low-level autonomy described previously, we have implemented a joystick interface to control the robot. The joystick (Microsoft Sidewinder) has 4 analog inputs (x, y, θ, ρ) and 9 buttons. The desired walking direction and orientation is indicated using the three main analog inputs (x, y, θ). The remaining analog input (ρ) can be switched between individually controlling the torso height, roll, yaw, pitch, and walking cycle. Currently, pre-defined arm motions and gestures are bound to the buttons. Figure 8 shows an experiment with H7 controlled by the joystick interface.

5 Autonomy in Object Manipulation

5.1 Autobalancer

The “AutoBalancer” software reactively generates dynamically-stable motions of a standing humanoid robot on-line, given an input motion ((Kagami et al., 2000a; Tamiya, Inaba, Inoue, 1999)). The system consists of two parts: 1) a planner for considering state transitions defined by the nature of the contacts between the legs and the ground, and 2) a dynamic balance compensator which maintains balance by formulating and solving a second order nonlinear constrained optimization problem. The latter can compensate for the centroid position and the tri-axial moments

of any standing motion, using all joints of body in real-time. The complexity of Auto Balancer is $O((p + c)^3)$, where p is the number of DOFs and c is the number of constraint equations (Autobalancer in Fig. 2).

5.2 Stereo Vision

The stereo vision hardware is the same as described in the previous section. We proposed a real-time depth map generation system using only standard PC hardware and a simple image capture card (Kagami, Okada, Inaba, Inoue, 2000c). Four key techniques are adopted to obtain accurate range data and achieve real-time performance: 1) use of a recursive (normalized) correlation method, 2) cache optimization, 3) online consistency checking, 4) leveraging the MMX/SSE(R) multimedia instruction set.

5.3 RRT-based Motion Planning

As the technology and algorithms for real-time 3D vision and tactile sensing improve, humanoid robots will be able to perform tasks that involve complex interactions with the environment (e.g. grasping and manipulating objects). The enabling software for such tasks includes motion planning for obstacle avoidance, and integrating planning with visual and tactile sensing data.

Arm Motions: Collision-free trajectories for a single arm (6-7 DOF) can be computed by searching the arm configuration space. In order to cope with the high-dimensional search, we use a randomized path planner with a sampling heuristic, known as *RRT-Connect*, which has been specifically designed to quickly handle single-query path planning problems without any preprocessing of the configuration space (Kuffner, LaValle, 2000). The planner incrementally builds two Rapidly-Exploring Random Trees (RRTs) (LaValle, Kuffner, 2000) of free configurations rooted at the start and the goal. RRTs are biased to expand towards the unexplored regions of the space, and have been shown to ultimately converge toward a uniform distribution over the free configuration space (Kuffner, LaValle, 2000).

We have implemented a prototype path planner that provides an interactive high-level control mechanism for object manipulation tasks. Through a graphical user interface, an operator can click and on a target location. Inverse kinematics is used to compute goal configurations from target reference frames. The software will then automatically compute the motions necessary to complete the task.

The output of the stereo vision system can be used to view an environment depth map in real-time within the simulation environment. Figure 9(left) shows a trajectory calculated by the planner to place the robot's hand inside a box.

Full-body Motions: We have also developed a modified RRT-based search strategy to generate dynamically-stable, collision-free trajectories for the whole body. Collections of statically-stable robot postures are used to as samples in order to plan



Fig. 9. Arm Motion Planning Using Stereo Vision Output (left) and Dynamically-stable motion for retrieving an object (center: simulation, right: H6 robot).

statically-stable collision free paths. Paths are converted to dynamically-stable trajectories for the whole body using Autobalancer (for details, see (Kuffner et al., 2001)).

Figure 9(center & right) shows a dynamically-stable, collision-free crouching motion for the H6 robot to reach a target object beneath an obstacle.

6 Autonomy in Human Interaction

In order to facilitate smooth interaction between humanoid robots and their human operators, good user interfaces are needed. Two main types of user interaction exist: a) humans interacting directly with the robot, and b) humans using a remote operation interface. We consider speech recognition and synthesis, as well human face tracking and recognition.

6.1 Voice Interface

One of the challenges of using speech recognition software with humanoids is the many audible sounds generated by the robot itself possibly adding unwanted noise to the input signal. Therefore, the voice recognition software should be designed not to be overly sensitive to noise. We have adopted software developed by Dr. Hayamizu at ETL. Some of the advantages of this software is that it can run under Linux (and thus, on our onbody processor), and the dictionary is customizable. Task-based dictionaries which contain several keywords for specific tasks are prepared, and made relatively robust in terms of noise. The corresponding speech synthesis software is a commercial product developed by Fujitsu, which also runs on Linux. Figure 10 shows a walking experiment based on voice commands.

6.2 Human Detection, Tracking, and Face Recognition

In order to identify and recognize a human in the robot's environment, appropriate visual detection, tracking, and face recognition software were developed using

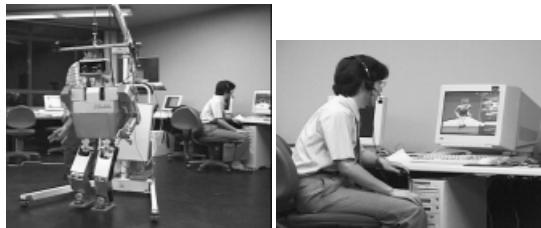


Fig. 10. H6 Voice Control



Fig. 11. H6 Identifying and Tracking a Human from the Depthmap While Walking (Left:Depth map, Center:Distance Labeling, and Right:Bounding Box)



Fig. 12. H6 Interacting With a Human

the stereo vision system described previously. First, candidate human shapes are identified by segmenting the depth image generated by the stereo vision software. Figure 11 shows a typical segmentation result. Then, the raw image from the segmented region is sent to a commercial, neural-net based face recognition software (MIRO). With a proper face database, the robot can be made to automatically recognize and distinguish between human users. Figure 12 shows human interaction using the face recognition interface.

7 Conclusion

In this paper, three types of low-level autonomous functions for remote controlled humanoid type robot were discussed, 1) bipedal locomotion, 2) object manipulation, and 3) human interaction.

For bipedal locomotion, an overview of an autonomous walking system was described that uses a log-scale cycle layered design. The layers include: 1) online ZMP compensation method, 2) dynamically-stable walking trajectory generation with self-collision avoidance, 3) footstep sequence planning.

For object manipulation, three software modules were discussed: 1) dynamically-stable standing motion generator “autobalancer”, 2) RRT (Rapidly exploring Random Tree) based arm path planning 3) full-body, dynamically-stable motion planning.

For autonomy in human interaction, experimental implementations of 1) stereo 3D vision based human identification, 2) face recognition software, 3) natural language interface and speech synthesis were described.

The humanoid robots H6 and H7 were used as platforms for research and experimentation of these various functions. We found that only a PC console with a microphone and joystick were enough to control the humanoid robot, despite the numerous degrees of freedom and complex balance constraints. The successful remote operation of humanoid-type robots has obvious applications in hazardous environments. However, we also believe that remote-operation task research is also useful for stimulating the development of broader low-level autonomy in humanoid robots.

References

- Brooks RA (1996) Prospects for human level intelligence for humanoid robots. In Proc. of the First Int. Symp. on HUmanoid RObots (HURD '96), pp 17–24.
- Hirai K (1997) Current and future perspective of honda humanoid robot. In Proc. of 1997 IEEE Intl. Conf. on Intelligent Robots and Systems (IROS'97), pp 500–508.
- Honda Co. Ltd. (1993a) Walking control system for legged robot. Japan Patent Office (A) 5-305583.
- Honda Co. Ltd. (1993b) Walking control system for legged robot. Japan Patent Office (A) 5-200682.
- Honda Co. Ltd. (1998) Walking pattern generation system for legged robot. Japan Patent Office (A) 10-86080.
- Inoue H, Tachi S, Tanie K, Yokoi K, Hirai S, Hirukawa H, Hirai K, Nakayama S, Sawada K, Nishiyama T, Miki O, Itoko T, Inaba H, Sudo M (2000) Humanoid robotics project of MITI. In Proc. of IEEE International Conference on Humanoid Robotics (Humanoid2000).
- Kagami S, Kanehiro F, Tamiya Y, Inaba M, Inoue H (2000a) Autobalancer: An online dynamic balance compensation scheme for humanoid robots. In Proc. of Fourth Intl. Workshop on Algorithmic Foundations on Robotics (WAFR'00), pp SA-79–SA-89.
- Kagami S, Nishiwaki K, Kitagawa T, Sugihara T, Inaba M, Inoue H (2000b) A fast generation method of a dynamically stable humanoid robot trajectory with enhanced zmp constraint. In Proc. of IEEE International Conference on Humanoid Robotics (Humanoid2000).

- Kagami S, Okada K, Inaba M, Inoue H (2000c) Design and implementation of onbody real-time depthmap generation system. In Proc. of International Conference on Robotics and Automation (ICRA'00), pp 1441–1446.
- Kagami S, Okada K, Inaba M, Inoue H (2000d) Plane segment finder. In 5th Robotics Symposia, pp 381–386.
- Kälviäinen H, Hirvonen P, Xu L, Oja E (1994) Comparisons of probabilistic and non-probabilistic hough transforms. In Proceedings 3rd European Conference on Computer Vision, pp 351–360.
- Kawamura K, Wiles DM, Pack T, Bishay M, Barile J (1996) Humanoids: Future robots for home and factory. In Proc. of the First Int. Symp. on HUmanoid RObots (HURD '96), pp 53–62.
- Kuffner JJ, LaValle SM (2000) RRT-Connect: An efficient approach to single-query path planning. In Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'2000), San Francisco, CA.
- Kuffner JJ, Nishiwaki K, Kagami S, Inaba M, Inoue H (2001) Motion planning for humanoid robots under obstacle and dynamic balance constraints. In Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'2001), pp 692–698.
- Kuffner JJ, Nishiwaki K, Kagami S, Kuniyoshi Y, Inaba M, Inoue H (2002) Self-collision detection and prevention for humanoid robots. Submitted to IEEE Int'l Conf. on Robotics and Automation (ICRA'2002).
- LaValle SM, Kuffner JJ Jr (2000) Rapidly-exploring random trees: Progress and prospects. In Proc. of Fourth Intl. Workshop on Algorithmic Foundations on Robotics (WAFR'00).
- Mirtich B (1998) Vclip: Fast and robust polyhedral collision detection. ACM Transactions on Graphics, 17(3):177–208.
- Murase Y, Sakai K, Inaba M, Inoue H (1998) Testbed hardware model of the hrp virtual platform. In Proc. of '98 Annual Symposium of Robotics-Mechatronics, pp 2P2–89–091.
- Nagasaka K, Inaba M, Inoue H (1999a) Stabilization of dynamic walk on a humanoid using torso position compliance control. In Proceedings of 17th Annual Conference on Robotics Society of Japan, pp 1193–1194.
- Nagasaka K, Inaba M, Inoue H (1999b) Walking pattern generation for a humanoid robot based on optimal gradient method. In Proc. of IEEE International Conference on Humanoid Robotics (Humanoid20Proc. of 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics No. VI).
- Nishiwaki K, Kagami S, Kuniyoshi Y, Inaba M, Inoue H (2001) Online generation of desired walking motion on humanoid based on a fast generation method of motion pattern that follows desired ZMP. In Proc. of 18th Annual Conference of Robotics Society of Japan, pp 985–986.
- Nishiwaki K, Nagasaka K, Inaba M, Inoue H (1999) Generation of reactive stepping motion for a humanoid by dynamically stable mixture of pre-designed motions. In Proc. of IEEE International Conference on Humanoid Robotics (Humanoid20Proc. of 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics No. VI), pp 702–707.
- Nishiwaki K, Sugihara T, Kagami S, Kanehiro F, Inaba M, Inoue H (2000) Design and development of research platform for perception-action integration in humanoid robot : H6. In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00), vol 3, pp 1559–1564.
- Tamiya Y, Inaba M, Inoue H (1999) Realtime balance compensation for dynamic motion of full-body humanoid standing on one leg. Journal of the Robotics Society of Japan, 17(2): 268–274.

- Yamaguchi J, Inoue S, Nishino D, Takanishi A (1998) Development of a bipedal humanoid robot having antagonistic driven joints and three dof trunk. In Proc. of the 1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp 96–101.
- Yodaiken V, M. Barabanov (2001) RT-Linux. URL <http://www.rtlinux.org>.

OpenHRP: Open Architecture Humanoid Robotics Platform

Hirohisa Hirukawa¹, Fumio Kanehiro¹, and Shuuji Kajita¹

National Institute of Advanced Industrial Science and Technology, METI.
1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568 Japan

Abstract. This paper introduces an open architecture humanoid robotics platform (OpenHRP for short) on which various building blocks of humanoid robotics can be investigated. OpenHRP is a virtual humanoid robot platform with a compatible humanoid robot, and consists of a simulator of humanoid robots and motion control library for them which can also be applied to a compatible humanoid robot as it is. OpenHRP also has a view simulator of humanoid robots on which humanoid robot vision can be studied. The consistency between the simulator and the robot are enhanced by introducing a new algorithm to simulate repulsive force and torque between contacting objects. OpenHRP is expected to initiate the exploration of humanoid robotics on an open architecture software and hardware, thanks to the unification of the controllers and the examined consistency between the simulator and a real humanoid robot.

1 Introduction

The Ministry of Economy, Trade and Industries of Japan has run Humanoid Robotics Project (HRP for short) since 1998 for five years (Inoue et al., 2000, 2001). The leader of HRP is Hirochika Inoue from the University of Tokyo, and Manufacturing Science and Technology Center (MSTC) is the secretary of the project. Four copies of a humanoid robot (called HRP-1), teleoperations cockpit for them and a virtual humanoid robot platform (V-HRP for short) (Nakamura et al., 2000; Nakamura, Hirukawa, Yamane, Kajita, Fujiwara, Kanehiro, Nagashima, Murase, Inaba, 2001) had been developed in phase one of HRP as the research platform, and various applications of humanoid robots are under development in phase two on the platform. We call the organization of the project *platform-based approach*. It is the antithesis of usual robotics projects in which elementary technologies are developed at first and they are integrated into a system at the final stage.

The architecture of the platform is being made open, step by step. The software of HRP-1 have been developed by Honda R&D as well as its hardware, and it is provided as a black box. We have replaced the controller for biped locomotion by our own one which has been developed on V-HRP. Besides, V-HRP has also been replaced by a new simulator on which we can develop the controllers portable to the hardware without any modification. Finally, a new humanoid robot HRP-2 is to be developed, and the controllers examined on HRP-1 will be applied to HRP-2.

The simulator is build on CORBA (Object Management Group, 2001) which is a standard software architecture, and the realtime controller of the robot is run

on ART-Linux (Ishiwata, Matsui, 1998) which is a realtime extension of Linux. Besides, the simulator and the controllers are now white boxes. Therefore, we call the package of the simulator and controllers with the compatible humanoid robot *OpenHRP* which stands for Open Architecture Humanoid Robot Platform.

The unification of the controllers for the simulated and real robot is realized by hardware abstraction and synchronization mechanism and employing ART-Linux in which realtime processing is available at the user level. Thanks to the unification, the controllers can share significant amount of software with the simulator, including the parameter parser, kinematics and dynamics computations and the collision detector. The consistency between the simulation and experiments is enhanced by introducing a new algorithm to simulate repulsive force and torque between contacting objects even when soft spring-damper mechanism is embedded in the feet of a humanoid robot. This is essential since that of HRP-1 is very soft. Because the unification of the controllers and the consistency between the simulated and real robots are realized, OpenHRP can be a useful virtual platform for humanoid robotics on which various fundamental technologies can be developed. The virtualization of the platform is very important to inherit software library from one hardware to another efficiently. For example, Honda R&D took only nine months or so to replace humanoid robot P2 by P3, since Honda R&D also has a nice virtualized platform and most software could be ported with the minimum effort.

This paper is organized as follows. Section 2 overviews the configuration of OpenHRP and that of HRP-1. Section 3 presents how the unification of the controllers for the simulator and the real robot is realized. Section 4 introduces how to isolate the non-realtime ORB for sharing the codes of the simulator with those of the realtime controller and a realtime collision detector is presented as an example of the code sharing. Section 5 shows examples of the simulation and experiments. Section 6 concludes the paper.

2 Overview of OpenHRP

2.1 Configuration of OpenHRP

The configuration of OpenHRP is shown in Fig. 1. OpenHRP can simulate the dynamics of structure-varying kinematic chains between open chains and closed ones like humanoid robots (Yamane, Nakamura, 1999). It can detect the collision between robots and their working environment including other robots very fast and precisely on which the forward dynamics of the objects are computed. It can also simulate the fields of vision of the robots, force/torque sensors and gradient sensors according to the simulated motions. We call the simulations *sensor simulations*. The sensor simulations are essential to develop the controllers of the robots. OpenHRP is implemented as a distributed object system on CORBA(Common Object Request Broker Architecture) (Object Management Group, 2001). A user can implement a controller using an arbitrary language on an arbitrary operating system if it has a CORBA binding.

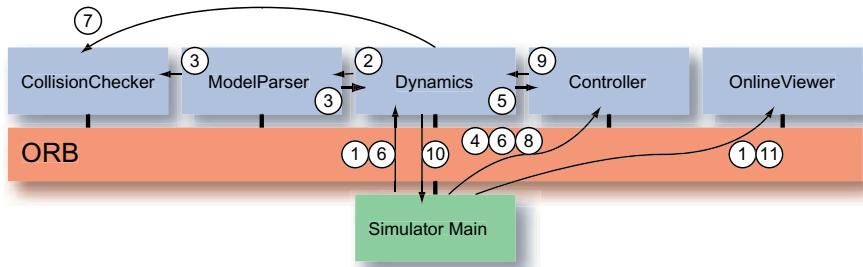


Fig. 1. OpenHRP Overview

The dynamics simulator of OpenHRP consists of five kinds of CORBA servers (see Fig. 1) and these servers can be distributed on the Internet and executed in parallel. Each server can be replaced with another implementation if it has the same interface defined by IDL (Interface Definition Language). Using the language independence feature of CORBA, ModelParser and OnlineViewer are implemented using Java and Java3D, other servers are implemented using C++. The functions of each server are as follows.

ModelParser This server loads a VRML file describing the geometric models and dynamics parameters of robots and their working environment, and provides these data to other servers.

CollisionChecker The interference between two sets of triangles is inspected, and the position, normal vector and the depth of each intersecting point are found. RAPID (Gottschalk, Lin, Manocha, 1996) is enhanced to this end.

Dynamics The forward dynamics of the robots are computed.

Controller This server is the controller of a robot, which is usually developed by the users of OpenHRP.

OnlineViewer The simulation results are visualized by 3D graphics and recorded.

Using the servers, the forward dynamics of the robots are computed in the following procedure. The total control flow is shown in Fig. 1.

Setting up of the simulation environment

(1) ModelParser reads a VRML file via HTTP protocol. The kinematics and dynamics parameters are sent to DynamicsServer and the geometric model is to CollisionChecker.

Execution of the dynamics simulation (2) Controller reads the outputs of the simulated sensors while communicating with DynamicsServer. (3) Let Controller and DynamicsServer execute the computations. Note that these computations can be run in parallel. The outputs of Controller are the torques of the actuators, and those of DynamicsServer are the updated states of the robot. (4) While the forward dynamics is computed, CollisionChecker is called to find the position, normal vector, and the depth of each intersecting point. (5) After these computa-

tions, let Controller send the control outputs to DynamicsServer. (6) Controller sends the outputs to DynamicsServer.

Visualization and recording (7) Acquire the current states of the world from DynamicsServer. (8) Send them to OnlineViewer which visualizes the simulated world and records it.

In order to evaluate the performance of OpenHRP, biped locomotion of a sample humanoid robot with 29DOF is simulated. The specifications of the used computer include CPU:Intel PentiumIII 933MHz, Memory:512MB, and OS:Linux-2.2.17. The computation time except the visualization is 25[ms] per the unit integration time, which is usually set around 1 [ms].

2.2 Humanoid robot HRP-1S

As an example of compatible humanoid robots with the simulator, HRP-1 is investigated in this paper. The configuration of the controller hardware of humanoid robot HRP-1 is shown in Fig. 2. The humanoid robot is called HRP-1S in the following to clarify that the Honda controller is replaced by our own one. The realtime con-

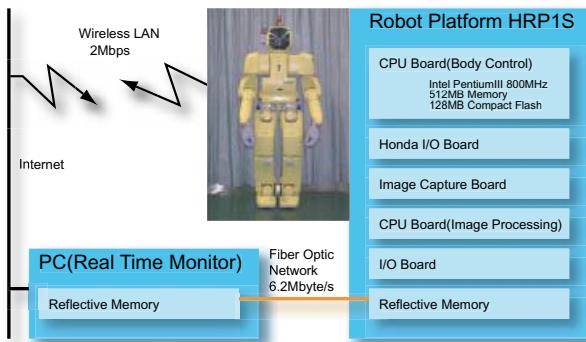


Fig. 2. Controller hardware of HRP-1S

troller runs on a CPU board in the backpack of HRP-1S, whose operating system is ART-Linux.

Using the optical fiber network connecting the pair of the reflective memory mounted on the backpack and the PC outside the robot, the internal states of the robot can be monitored at the PC in realtime.

3 Unification of the controllers

As mentioned above, the controllers developed on OpenHRP can be applied to the real counterpart without any modification. The unification of the controllers for the virtual and real robot has been realized by introducing software adapters for

two robots respectively and employing ART-Linux on which real-time processing is available at the user level. This is a distinguished feature, considering that RT-Linux (Yodaiken, Barabanov, 2001) realizes it only at the kernel level. Thanks to this feature of ART-Linux, users can implement realtime applications as if they are non-realtime ones. This is the first key to realize the identical controller for the virtual and real robot.

By the unification, the controllers can share softwares with the dynamics simulator of OpenHRP, including the parameter parser, kinematics and dynamics computations and the collision detector.

This feature can make the development of the controllers more efficient and the developed controllers more reliable. Besides, it becomes easier to feedback the experimental results for the improvements of the dynamics simulator when the run-time environments of the controllers are identical.

The controller must be implemented by exactly same methods with the same signatures for the virtual and real robots, to realize the unification of the controllers; the following two requisites must be hold to this end. (1) abstraction of the controller API where the software body looks like the same as the hardware body, and (2) a synchronizing mechanism that can absorb the difference between the speed of time in the simulated and the real world. The tricks to satisfy these requisites are described in the following.

The unification of the controllers does not make sense if the results of the dynamics simulation are not consistent with those of the experiments. The soles of HRP-1 is very soft because a spring-damper mechanism is embedded in each foot. We have developed an algorithm to simulate general spring-damper mechanisms precisely to keep the consistency, which is also presented in the following.

3.1 Unification architecture

Hardware abstraction The first requisite is realized based on the plug-in architecture (Kanehiro, Inaba, Inoue, Hirukawa, Hirai, 2001), where application software is separated into two layers at the adapter level and the counterpart beyond the adapter can be replaced. The software architecture of simulated HRP-1S and the real one is shown in Fig. 3. The controller API of OpenHRP is not identical with that of Honda I/O board shown in Fig. 2. The unification of a controller is realized by introducing the adapter whose API is the abstracted controller API mentioned above. The emulation adapter is layered on the controller API of OpenHRP which reads the outputs of the sensor simulators and write the inputs of DynamicsServer, and the hardware adapter is put on the API of Honda I/O board of HRP-1S. The API of the adapters is identical shown as follows.

```
class robot_adaptor
{
public:
    virtual bool open(int argc, char *argv[]) ;
    virtual bool close() ;
```

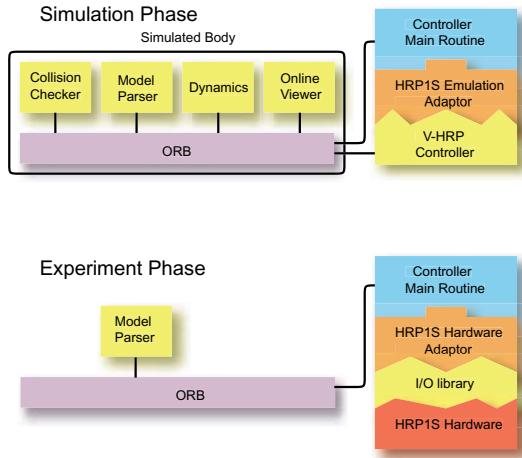


Fig. 3. Software architecture of simulated and real humanoid robot HRP-1S

```

virtual bool read(robot_state *rs);
virtual bool write(motor_command *mc);
};

```

Synchronization mechanism Though various speedup techniques for the dynamics simulation have been proposed to the present, but it is still difficult to compute the forward dynamics of robots with many degrees of the freedom including the visualization in realtime. Generally speaking, time goes by at a slower speed in the simulated world than in the real world. Besides, the speed is not constant in the virtual world due to the fluctuation of the required computation time.

Therefore, the synchronization mechanism of the controller must not be embedded in the main logic of the controller, but in the hardware adapter. Then the adapter can manage a change of the speed of time by a synchronization mechanism.

Note that the outputs of the controller can not always be updated in time in the real world. So it is possible that the controller fails to handle the robot even when it has succeed to move the robot in the virtual world.

4 Determining the repulsive force

4.1 Finding the interference state

We have proposed a complete algorithm to find the constraints which is imposed on a polyhedral object in contact with another (Hirukawa, Matsui, Takase, 1994). The key idea to determine the constraints is finding separating planes between the neighborhoods of a contact point in two objects. But there are two reasons why the complete algorithm does not work to find the interference state in the dynamics simulation;

- CollisionChecker detects the interference between two sets of triangles which are obtained from the decomposition of the boundary representation of two objects under consideration. Then the topological information between the triangles has been lost, and it is not possible to analyse the neighborhoods for finding the separating planes.
- The objects may penetrate each other during the numerical dynamics simulation, and therefore no separating plane exists between the neighborhoods any more.

The first problem is solved by recovering the topological data to find the neighborhoods, which can be done in $O(1)$ time using the OBB-tree structure (Gottschalk et al., 1996). There is no way to avoid the second one, when we simulate the dynamics on a floating point arithmetic. Therefore, the only option for us is finding a ‘reasonable’ constraint at each intersecting point. We choose the constraints that give the smallest penetration depth between two objects.

4.2 Finding the repulsive force

This section presents how to assemble the interference date obtained by the above geometric algorithm for finding the repulsive force between the objects. Let $n_i : 3 \times 1, i = 1 \dots M$ be the constraining vector at the i -th penetrating point, $p_i : 3 \times 1, i = 1 \dots M$ the position of the point, and $d_i : 3 \times 1, i = 1 \dots M$ the depth of the point, where M is the cardinal number of the penetrating points.

Then we try to find the infinitesimal translation $\Delta X : 3 \times 1$ and rotation $\Omega : 3 \times 1$ of the constrained object which make the objects at the interference state from the contact state without the overlapping of their inside. In other words, we want to find the infinitesimal screw of the object which make the depth at the penetrating points as close as possible. This problem can be formulated as the following minimization problem.

$$\text{minimize} \quad E^T E \quad \text{subject to} \quad A \begin{pmatrix} \Delta X \\ \Omega \end{pmatrix} \geq 0, \quad (1)$$

where

$$E = A \begin{pmatrix} \Delta X \\ \Omega \end{pmatrix} - D, \quad A = \begin{pmatrix} n_1^T & (p_1 \times n_1)^T \\ n_2^T & (p_2 \times n_2)^T \\ \vdots & \vdots \\ n_M^T & (p_M \times n_M)^T \end{pmatrix}, \quad D = \begin{pmatrix} -d_1 \\ -d_2 \\ \vdots \\ -d_M \end{pmatrix}. \quad (2)$$

This is a linear programming problem with a nonlinear objective function, and can be solved by many package software. An alternative way is finding the solution with the least square error by

$$\begin{pmatrix} \Delta X \\ \Omega \end{pmatrix} = A^+ D, \quad (3)$$

where $(\cdot)^+$ stands for the pseudo inverse of a matrix, and projecting the solution orthogonally onto the polyhedral convex cone defined by inequality (1). The current implementation uses the latter way.

Then the spring element of the repulsive force is found by

$$\begin{pmatrix} F \\ \tau \end{pmatrix} = -K \begin{pmatrix} \Delta \hat{X} \\ \hat{\Omega} \end{pmatrix}, \quad (4)$$

where F and τ are the force and torque respectively, $K : 6 \times 6$ a spring constant matrix and $\Delta \hat{X}$ and $\hat{\Omega}$ are the solution of the above minimization problem.

The algorithm to find the damper element is examined next. Let V_0 and ω_0 be the relative translational velocity and the angular velocity between the objects in contact respectively. Then the proposed algorithm finds the orthogonal projection $(\tilde{V}, \tilde{\omega})$ of (V_0, ω_0) onto the polyhedral convex cone defined by

$$A \begin{pmatrix} V \\ \omega \end{pmatrix} \geq 0. \quad (5)$$

The damper force along the normal vector at the penetrating points is computed from $(\tilde{V}, \tilde{\omega})$ based on a nonlinear damper model (Hunt, Crossley, 1975; Marhefka, Orin, 1996) by

$$\begin{pmatrix} F \\ \tau \end{pmatrix} = -\lambda \begin{pmatrix} \Delta X^n \tilde{V} \\ \Omega^n \tilde{\omega} \end{pmatrix}, \quad (6)$$

where $\lambda = \frac{3}{2}\alpha K$, α is a constant defining the linear dependence of the coefficient of restitution on the impact velocity, and the n -th power of a vector is defined by the vector whose elements are the n -th power of the original elements respectively. In the nonlinear model, the damping is dependent on the penetration depth, and then the damping increases with the depth of the penetration continuously as the objects is coming into contact (Hunt, Crossley, 1975; Marhefka, Orin, 1996). This feature is essential to simulate the biped walking in which a foot of the robot collides the floor at a finite velocity.

The damper force along the tangent vector can be found from the projection $(\tilde{V}, \tilde{\omega})$ of (V_0, ω_0) into the null space of the coefficient matrix of the left hand side of inequality (5) by

$$\begin{pmatrix} F \\ \tau \end{pmatrix} = -C \begin{pmatrix} \tilde{V} \\ \tilde{\omega} \end{pmatrix}, \quad (7)$$

where $C : 6 \times 6$ is the damper coefficient matrix to simulate the tangent force generated by the friction.

In total, the repulsive force and torque by the spring-damper model is given by

$$\begin{pmatrix} F \\ \tau \end{pmatrix} = -K \begin{pmatrix} \Delta \hat{X} \\ \hat{\Omega} \end{pmatrix} - \lambda \begin{pmatrix} \Delta \hat{X}^n \tilde{V} \\ \hat{\Omega}^n \tilde{\omega} \end{pmatrix} - C \begin{pmatrix} \tilde{V} \\ \tilde{\omega} \end{pmatrix}. \quad (8)$$

Note that the final output is the orthogonal projection of the force and torque in Eq.(8) onto the polyhedral convex cone defined by an enhanced inequality of (1) in which the friction cones are taken into account. The projection is necessary to avoid the generation of the force which is outside the friction cone at each contact point.

5 Sharing software between the simulator and the controllers

5.1 Isolation of ORB

The controller and the dynamics simulator can share significant amount of codes. For example, collision detection is one of major building block of the simulator, and it is also essential in the controller to avoid the self-collision of a moving humanoid robot. It is needless to say that basic vector and matrix operations are included both the simulator and the controller. The parameters parser can also be shared. The forward kinematics computation of robots are common too. The unified controller makes the code sharing easier, and therefore the development of the controllers more efficient. Another good news of the code sharing is that the controllers can be more reliable since the building blocks borrowed from the simulator has been already examined intensively by the simulation.

However, there is a barrier to reuse the code in the controller. That is, the servers like `CollisionChecker` or `DynamicsServer` are implemented as CORBA servers as shown in Fig. 1. Though realtime functions are included in the specifications of CORBA since version 2.4 and we can find the implementations of the functions, but the overhead of IIOP (Internet Inter-Orb Protocol) used in CORBA is not small enough for the controller which must update the outputs at a few milliseconds. This overhead can be bypassed by the following architecture.

Let the servers like `CollisionChecker` or `DynamicsServer` consist of two layers. The lower layer is a normal library which is independent to CORBA, and the higher layer wraps the library by CORBA interface and converts the data structures between the library and the interface. For example, these servers call `ModelParser` through the ORB when reading a VRML file describing the parameters of robots, but they access non-CORBA interfaces when they control the robots in realtime. The internal configuration of the controller is shown in Fig. 4.

5.2 Realtime collision checker

A realtime collision checker is presented here as an example of shared codes between the simulator and the controller. When a humanoid robot is moving, it is desirable that the self-collision between the links of the robot is checked in realtime for enabling emergency stopping of the robot.

Let N be the number of the links of a humanoid robot. Then the number of pairs of the links is $\binom{N}{2}$, and the collision detection must be executed for $\binom{N}{2}$ pairs. Taking into account that the interference between consecutive links does not occur since the movable range of the joints are limited, we still need to check $\binom{N}{2} - (N - 1)$

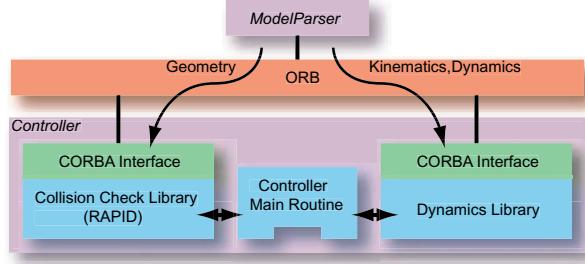


Fig. 4. Internal configuration of the controller

pairs. This number of the combination becomes 350 in the case of HRP-1S whose $N = 29$. The average computation time for the collision detection is about 10 [ms], when HRP-1S takes many random posture. Here, the geometric model of HRP-1S consists of about 10,000 polygons. This is not fast enough, because the cycle of the controller is 5 [ms] and we assume that its 20% can be assigned for the collision inspection. From the above experiment, about 35 pairs can be checked in 1 [ms].

Then we propose the following approach combining an off-line pre-processing and a reduced on-line computation.

1. Check the self-collision at a rough resolution for each joint, and try to find safe range for some joint.
2. Re-check the self-collision at a fine resolution for the same range found at the previous step.
3. While keeping the joint of the arms within the safe region, check the self-collision in realtime only for the suspicious pairs.

In the case of HRP-1S, we need not check the collisions between the arm and the leg in the same side if we keep the pitch joint of the shoulder or the roll joint in the safe range.

There is no significant safe range for the collision between the left and right legs, but each roll link of the hip and each pitch link of ankles are covered by the other links and therefore need not be checked.

We assume that only legs are moving during walking, 16 pairs must be checked between the legs, 16 pairs between two links around the hands and the legs, 4 pairs between the main body and foot links, and 36 in total, which can be done in realtime. An example posture while walking is shown in Fig. 5(a). An experimental result is shown in Fig. 5(b). The required time is longest when the robot is standing, and decreases while walking.

6 Simulations and experiments

The modules which have been implemented on OpenHRP include

- (1) feedback controller for biped locomotion,

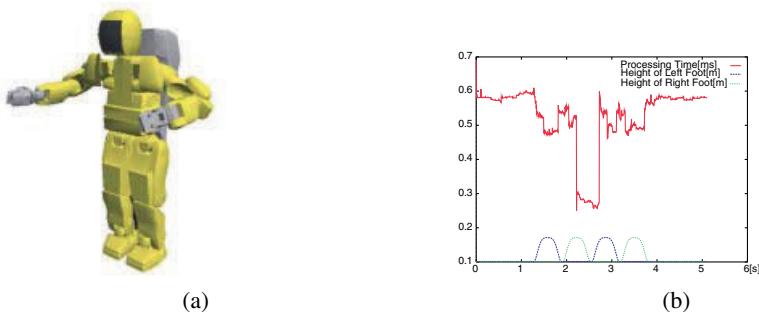


Fig. 5. Safe posture for walking and the computation time for the self-collision

- (2) balance controller at the standing position,
- (3) collision avoidance planner for arm motions.

(1) tries to stabilize the biped locomotion of the robot whose desired trajectories has been generated by an off-line pattern generator (Nemoto et al., 2000). (2) keeps the balance of the robot which is supposed to carry out some task at the standing position (Nakamura et al., 2000). (3) plans collision avoidance motions of an arm of the robot (Nakamura et al., 2000).

The dynamics simulation and the corresponding experiments are carried out to examine the proposed algorithms. The first example is a damped oscillation of the robot. The robot is inclined slightly at the standing position, and the external force to incline the robot is removed. Then the robot shows a damped oscillation along the pitch axis, when no feedback control is applied. Figure 6 shows the torque along the pitch axis of a foot of the robot. The curve with the higher first peak is the result

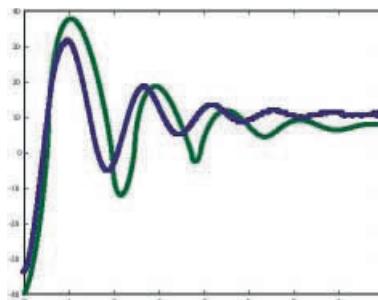


Fig. 6. Damped oscillation of HRP-1S

of the simulation, and the other is that of the experiment. The cycle of the oscillation and the attenuation rate seems to coincide sufficiently. The curves converge to slightly different values, which is caused by the calibration error to find the posture of the robot with the zero inclination.

The next example is the comparison of the normal repulsive force from the floor to a foot of the robot while it is walking with a feedback control. The above curve is

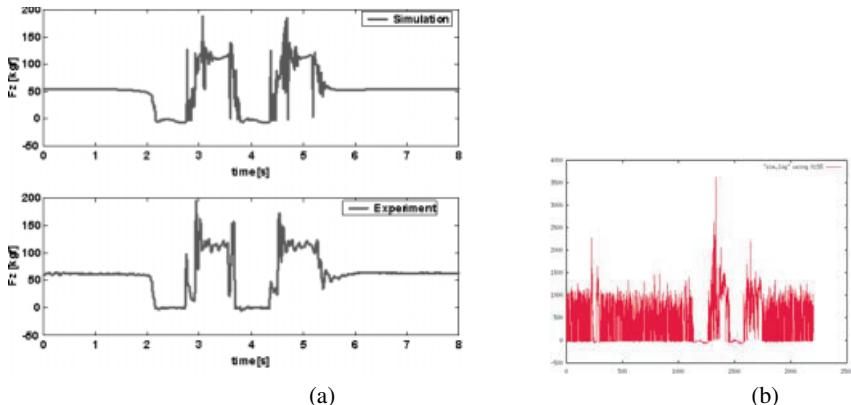


Fig. 7. Normal repulsive force from the floor from the spring-damper model and the rigid object model

from the simulation and the below is from the experiment. The curves also seem to coincide sufficiently. Figure 7(b) shows the corresponding curve when the repulsive force is found based on the conservative law of momentum by V-HRP (Nakamura et al., 2000). The robot seems to oscillate at a high frequency, since the simulation is based on a rigid object model. It is very difficult to evaluate feedback laws on such a simulation. This is one of the major feature of the proposed algorithm.

Figure 8 shows a snapshot while the robot is walking on a rough terrain.

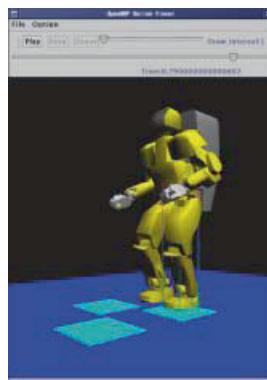


Fig. 8. Walking on a rough terrain

The proposed geometric algorithm makes possible to execute such a simulation.

7 Conclusions

This paper presented an open architecture humanoid robot platform OpenHRP. The results can be summarized as follows.

- The unification of the controllers for the virtual and real robot has been realized by introducing the adapters for two robots respectively with the synchronization mechanism, and employing ART-Linux on which real-time processing is available at the user level.
- The consistency between the simulator and the robot is enhanced by the new algorithm to simulate general spring-damper mechanism.
- Thanks to the unification, the controllers can share softwares with the dynamics simulator including the parameter parser, kinematics and dynamics computations and the collision detector. This feature can make the development of the controllers more efficient and the developed controllers more reliable. A real-time collision checker for humanoid robots has been developed as an example of the shared code.

We claim that a humanoid robot platform deserves to be call a platform if the identical software can be used either on the simulator or on the robot and if the consistency between them is satisfactory kept. We believe that OpenHRP is expected to be the first humanoid robot platform in this sense and that it can initiate the exploration of humanoid robotics on an open architecture hardware and software.

Acknowledgments

The authors thank Hirochika Inoue, Yoshihiko Nakamura and Katsu Yamane from the University of Tokyo, and Kazuhito Yokoi, Kenji Kaneko and Kiyoshi Fujiwara from AIST for their cooperations.

References

- Gottschalk S, Lin MC, Manocha D (1996) OBB-Tree:a hierarchical structure for rapid interference detection. In Proc. of ACM Siggraph 96.
- Hirukawa H, Matsui T, Takase K (1994) Automatic determination of possible velocity and applicable force of frictionless objects in contact from a geometric model. IEEE Trans. Robotics and Automation, pp 309–322.
- Hunt KH, Crossley FRE (1975) Coefficient of restitution interpreted as damping in vibroimpact. ASME Journal of Applied Mechanics, pp 440–445.
- Inoue H et al. (2000) HRP:humanoid robotics project of MITI. In Proc. of the First IEEE-RAS International Conference on Humanoid Robots.
- Inoue H et al. (2001) Overview of humanoid robotics project of METI. In Proc. of the 32nd ISR.
- Ishiwata Y, Matsui T (1998) Development of Linux which has advanced real-time processing function. In Proc. 16th Annual Conference of Robotics Society of Japan, pp 355–356.

- Kanehiro F, Inaba M, Inoue H, Hirukawa H, Hirai S (2001) Developmental software environment that is applicable to small-size humanoids and life-size humanoids. In Proc. of the 2001 IEEE International Conference on Robotics & Automation.
- Marhefka DW, Orin DE (1996) Simulation of contact using a nonlinear damping model. In Proc. IEEE Int. Conf. Robotics and Automation, pp 1662–1668.
- Nakamura Y et al. (2000) V-HRP:virtual humanoid robot platform. In Proc. of the First IEEE-RAS International Conference on Humanoid Robots.
- Nakamura Y, Hirukawa H, Yamane K, Kajita S, Fujiwara K, Kanehiro F, Nagashima F, Murase Y, Inaba M (2001) Humanoid robot simulator for the METI HRP project. In Proc. of the 32nd ISR.
- Nemoto Y et al. (2000) Development of a basic motion control library for humanoid. In Proc. 32nd ISR.
- Object Management Group (2001) OMG. URL <http://www.omg.org/>.
- Yamane K, Nakamura Y (1999) Dynamics computation of structure-varying kinematic chains for motion synthesis of humanoid. In Proc. of the 1999 IEEE International Conference on Robotics & Automation.
- Yodaiken V, Barabanov M (2001) RT-Linux. URL <http://luz.cs.nmt.edu/~rtlinux>.

Building Spined Muscle-Tendon Humanoid

Masayuki Inaba, Ikuo Mizuuchi, Ryosuke Tajima, Tomoaki Yoshikai, Daisuke Sato, Koichi Nagashima, and Hirochika Inoue

Department of Mechano-Informatics, The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Abstract. Human can perform variety of limber whole-body motions using numerous muscles and huge number of various sensors. The human brain has all the connections to the sensors and muscles, and learn how to manage them for whole-body motions. In this research, we have aimed to build a complex body with physically massive parallel sensor-motor systems to enter the next stage for studying humanoid brain systems. It is designed to have a flexible spined torso and a whole-body with fully muscle-tendon driven systems. In this paper the design and implementation of the first model of the humanoid is described with some experiments.

1 Introduction

So far several whole-body humanoid robots have been developed (Hirai, 1997; Kato, Ohteru, Kobayashi, Shirai, Uchiyama, 1974; Nishiwaki, Sugihara, Kagami, Kanehiro, Inaba, Inoue, 2000). However, we often feel that the whole-body motions of the robots are restricted or unnatural because of the rigid torso. If we want to put a controllable flexibility in the torso, the mechanisms to drive it becomes a crucial point to be solved. Although there have been some studies on humanoid which has a movable torso (Yamaguchi, Inoue, Nishino, A.Takanishi, 1998), these researches concentrated on the movability of the torso at the hip, and there was no flexibility in the torso. One of the method to build the flexible torso is to use a spine like a vertebrate animal. The spine with multiple vertebra requires multiple actuators to be connected in each elements. One of the method to solve the problem is to control vertebra with muscle-tendon driven systems (Mizuuchi, Inaba, Inoue, 2001). However, the multiple muscle-tendon systems in a torso may give interference between each others and make hard to keep repeatability in various complex motions. This means the artificial robot that has a vertebrate type spine with muscle-tendon systems requires a capability to solve the problem like vertebrate animals do. One of our goals in this research is to provide a research platform to extend humanoid brain to manage this kind of complex bodies.

We have put several design policies for this research aim. The first is to adopt stiffness-controllable flexible torso with muscle-tendon systems. The second is to accept impact forces in a ball joint with a large surface contact. The third is to have new manufacturing systems for prototyping. Because many parts such as vertebrae uses free-formed surfaces, new method to handle free-formed surfaces are indispensable. The fourth is to keep inheritance of brain and mother software for development. The 'mother' software to generate the body models and basic program

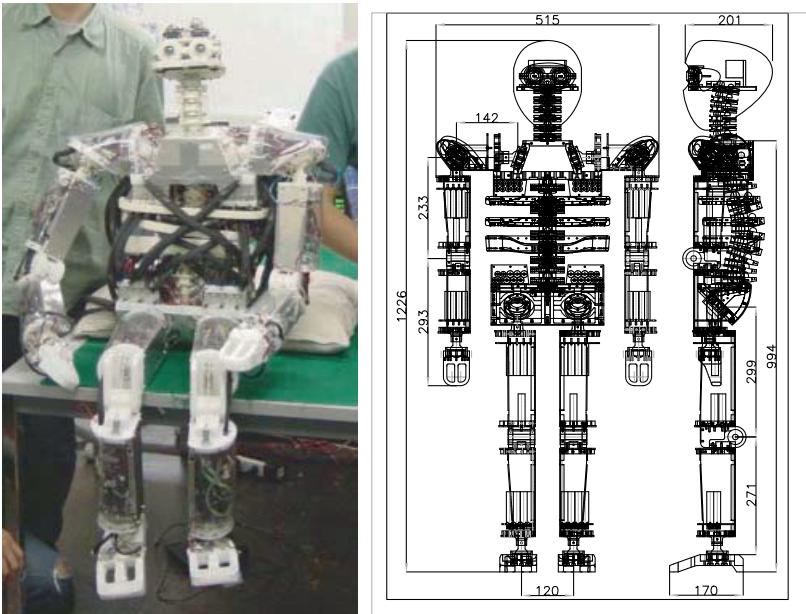


Fig. 1. Kenta, a spined muscle-tendon humanoid, Right: CAD design of Kenta

frames of controllers, which is used in the development of our humanoid series should be inherited in new design of humanoid.

2 Design of spined muscle-tendon humanoid “Kenta”

2.1 Design concept of the spined torso

By having a flexible spine whose flexibility is variable, robots will have several characteristics below.

The increase of DOF of the torso: A flexible spine has more degrees of freedom than a rigid torso. It means that the movable area of the robot or of the arms would be expanded. It also means that, by using the increased DOFs, motions of the robot can be more efficient compared to a rigid torso. (Figure 2.)

Multiple joint structure: There is some differences between multi-joint spine and single concentrated-joint spine (Yamaguchi et al., 1998). For example, in the motion to rise the upper body from the floor, a flexible spine robot can move each joint one by one from the neck joint to the hip joint. This procedure is more efficient than rising the upper body from floor using only hip or crotch joint. Furthermore, the motions of the flexible spine robots could be nearer to natural (human-like) motions.

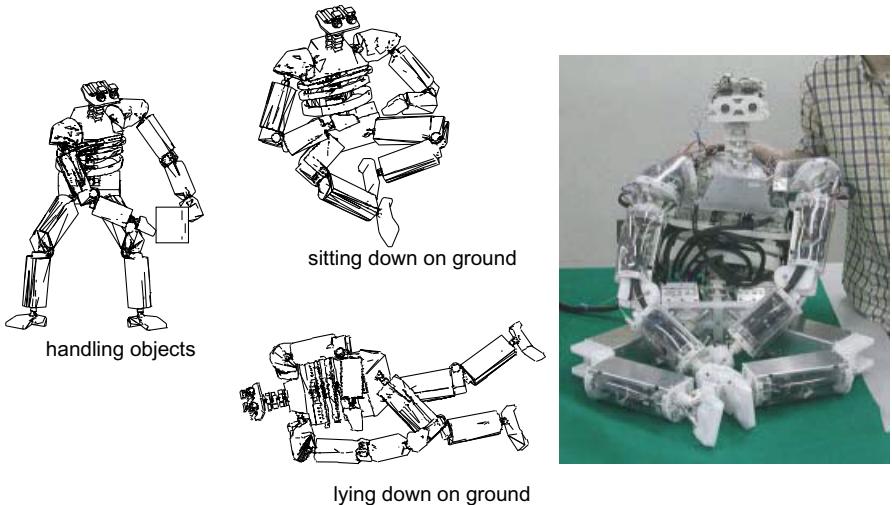


Fig. 2. 96 DOF whole body tendon-driven humanoid Kenta

Softness in material: By the softness of the spine, the robot can absorb the shock, and it has a safety at the physical contact with a human or objects.

Variable flexibility: For example, when the robot needs to lift up a heavy object from the floor, the spine should be comparatively hard. The variability of the flexibility is very important.

2.2 Mechanical structure of the torso

The design of the spine structure of Kenta follows the basic policy: based on the investigation of the structure and characteristics of human's spine, we try to follow the design and function as much as possible.

The vertebrae Kenta's spine consists of 9 vertebrae (2 types). The 9 vertebrae and hip and clavicle construct 10 ball-and-socket joints of the spine. Three ribs are attached to the three of the vertebrae. Between each two joints, there is a 'disk' made of silicone rubber, and there are also ligaments made of tension spring between the joints.

Normal human's spine has a curvature. By the curvature, the strength against the axial force is increased. The vertebrae of Kenta's spine have two kinds of inclination (each angle is 10 degree) (the left figure of the Fig. 3), to organize the curvature (Fig. 3). The movable range of roll-, pitch-, and yaw-rotation of each joint is approximately ± 10 degrees. But, the range of pitch-rotation is somewhat limited physically to about $-3 + 10$.

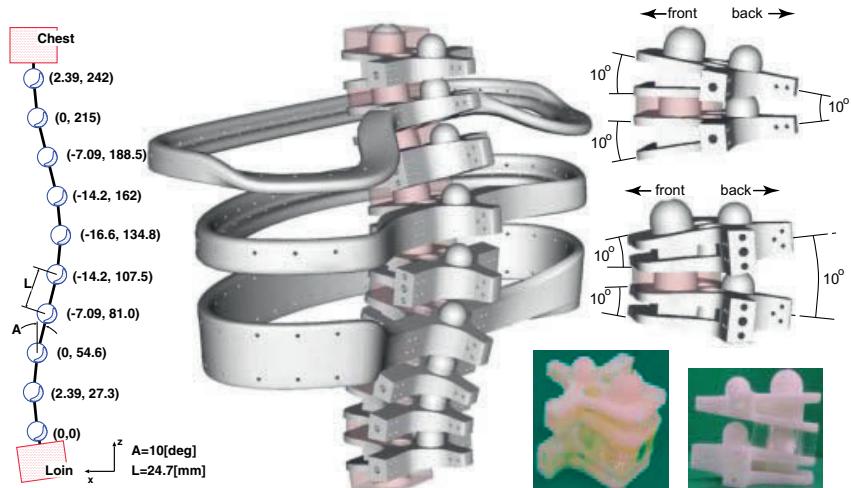


Fig. 3. Left: the link arrangement of the spine, Middle: CAD design of spine, Right: two kinds of inclination of vertebrae (an elastic part is between the two.)

The joint between vertebrae Each vertebra has a spherical protuberance on the top and a spherical dent on the bottom, as if there is a ball between the vertebrae (Fig. 3). The silicone rubber ‘disk’s are made using a plastic mold. The disks are put between the vertebrae in pressured condition and contribute to the flexibility of the spine. Each vertebra has two transverse processes (side protuberances) and a spinous process (back protuberances); process means a protuberance. Ligaments, which connects the processes, retain the structure and contribute to the flexibility as well. Expecting the function of ligaments, the vertebrae of Kenta’s spine have the processes and there are the ligaments made of tension spring. The lower the position of the spring is, the higher the spring coefficient becomes. The small ball at the back of the top surface is to prevent the joint’s over-bending.

The rib While human’s ribs protect the internal organs, robots do not have any internal organs. The ribs of the robot’s spine play another role as muscle-fixing parts. By fixing muscles to the ribs, the muscles are apart from the center of rotation of the trunk. This increases the force to change the posture of the spine, by the same tension of muscle; while the speed of changing the posture by the same actuator speed is decreased.

The actuators There are 40 motors to move the spine; twenty in the shoulder-block, and the other twenty in the hip-block. In the shoulder block, ten are in the right block, and the other ten are in the left block; in the hip block, ten in right and the other ten in left. The motors placed in inner side have power rather than speed,

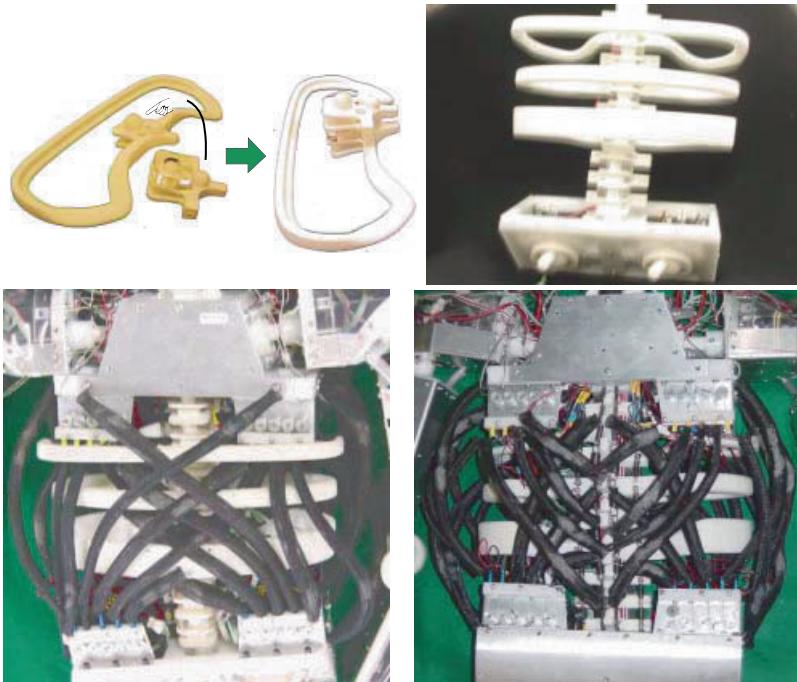


Fig. 4. The arrangement of muscles of Kenta's spine (left:front, right:back)

and the ones in outer side have speed rather than power. This is by the differences of gear ratios.

Figure 4 shows the arrangement of Kenta's muscles. At the front side, muscles are placed traversing the front surface, while at the back side, all muscles are fixed to the spinous processes (back protuberances) of the vertebrae. At the right and left sides, muscles traverse the surfaces.

2.3 The manufactureing methods

The complex shapes of Kenta's parts such as vertebrae, ribs, neck, eye-balls, etc. were designed on 3D-CAD softwares. At the manufacturing stages, the designed shapes were transferred all in the form of digital data. Parts were manufactured using NC machine tools, FDM-method (Fused Deposition Modeling), and SLS-method (Selective Laser Sintering). The left figure of the Fig. 4 shows a part of the one-body rib-vertebra. It is quite difficult for conventional CAD softwares to design this kind of complex shape, which can be designed using 3D modeling tools more easily.

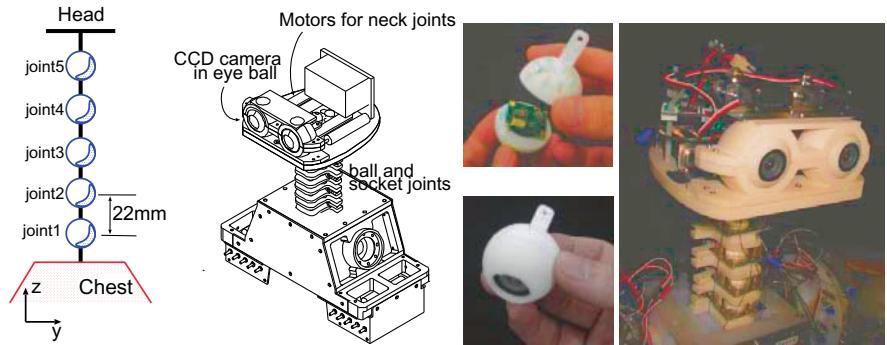


Fig. 5. The structure of Kenta's head and eye-balls

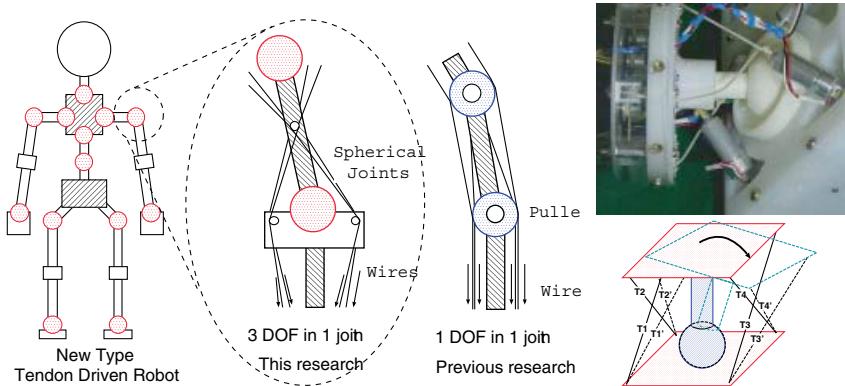


Fig. 6. Left: pulley-wire vs. non-pulley-wire, Right: ball-socket joint of crotch

2.4 The neck and head

The left of Fig. 5 shows the link arrangement of the Kenta's head. The structure of neck is like a small-scale version of the spine. The neck has five ball-and-socket joints like spine's vertebrae(the middle of the Fig. 5). Each joint consists of two small vertebrae and a silicone-rubber disk between the vertebrae. The elastic disks generates the force to restore to the natural posture when the neck is not in the natural posture. The neck structure is driven by six muscles. the posture of the structure is decided by the balance of muscle's tension, rubber-disks' force, and the external forces containing gravity. The structure of eyes are also ball-and-socket structures. Each eye-ball contains a CCD color camera. The eyes moves (pan and tilt) synchronously by two actuators.

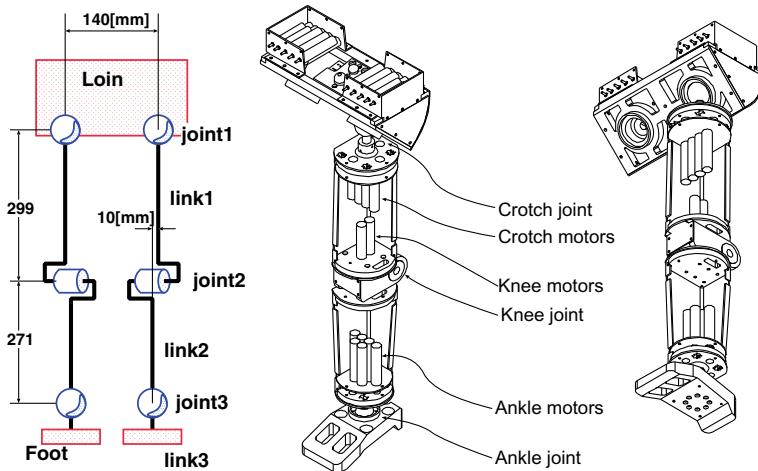


Fig. 7. Left: The DOF arrangement of Kenta's leg, Right: 3D view of Kenta's leg

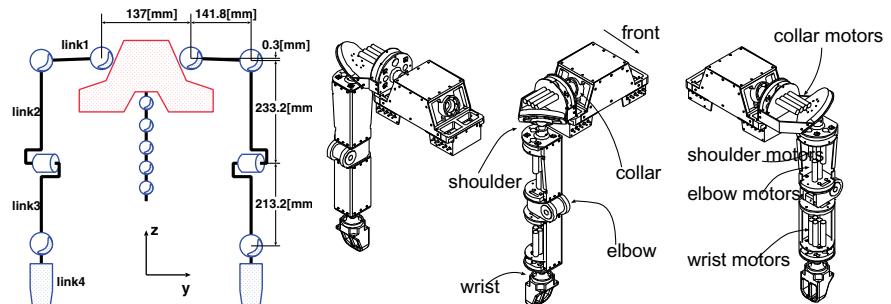


Fig. 8. Left: The DOF arrangement of Kenta's arm, Right: 3D view of Kenta's arm

2.5 The arms and legs

As a way to build a muscle-driven system, pulley-wire system and ball-socket joint system can be considered (Fig. 6). While pulley-wire system is easier to control because Jacobian matrix of joint-angle and muscle-lengths does not depend on the angle of the joint, ball-socket has more simple structure and less number of parts. Kenta's structure is the latter because of Kenta's huge degrees of freedom. Each ball-and-socket joint (spherical joint) is driven by four tendons. By arranging each tendon slantingly, the movable range of yaw-angle is expanded.

Legs: The left figure of Fig. 7 shows the arrangement of DOF of Kenta's legs. Each crotch joint has 3 DOF, each knee joint has 1 DOF, and each ankle joint has 3 DOF. The movable range of hip and ankle joints is 55 degrees in roll and pitch rotation, and 45 degrees in yaw angle. The right figure of Fig. 7 shows 3D view



Fig. 9. Left: tension sensor unit with a photo interrupter, Right: crotch joint

of Kenta's leg in 3D CAD software. Motors to pull the muscles are embedded in the thigh and calf. Each leg is equipped with 10 motors.

Arms: The left figure of Fig. 8 shows the arrangement of DOF of Kenta's arm. Each shoulder consists of two ball-socket joints as shown in the left of Fig. 8. It is closer to human's shoulder's movability than conventional single-jointed shoulder. The right figure of Fig. 8 shows 3D view of Kenta's leg in 3D CAD software. Motors to pull the muscles are embedded in the structure. Each arm has 14 motors.

2.6 The sensors of Kenta

Muscle-length sensors: By measuring the lengths of muscles, it is possible to control the lengths. All of Kenta's muscles are actuated by geared DC-motors, each of which is equipped with a rotary encoder. The lengths of muscles are calculated using the signal of the rotary encoders. One thing to be noticed is that the relation between the numerical value of rotary encoder and the angle of motor's axis is not linear since using pulley.

Muscle-tension sensors: By measuring the tensions of muscles, it is possible to control the tensions. The antagonistic muscles to drive a joint always have to have tensions. Every muscle of Kenta has a tension-sensor. Figure 9 shows the tension-sensor unit attached to Kenta's muscles. When the tension of the wire changes, the length of the coil spring in the cylinder. By measuring the distance between the piston and the bottom of the cylinder, the tension is detected. The measurement of the distance is done by a photo interrupter which uses infrared light emitting diode and infrared photo transistor.

Motor-current sensors: In the motor-driver circuit, there is a $0.5[\Omega]$ resister connected in series in the motor-power line. By measuring the drop of the voltage through the resister, the motor-current can be detected.

Vision sensors: Kenta is equipped with two CCD color cameras. Using the two cameras, it can calculate the depth by stereoscop. The depth map, 3D optical

flow, color labeling, plane segment finder (Okada, Kagami, Inaba, Inoue, 2001), etc. are implemented.

3D-accelerometers and gyroscope: There are eight 3-axis-accelerometers; in the hip, five vertebrae, the chest, and the head. They help to detect the information about the posture of the spine, and the information on Kenta's inclination, by detecting the vector of gravity force. There is also a gyroscope in the head block. Reflective eye motions are done using this sensor.

Tactile sensor: Kenta has 62 tactile sensor unit (FSR: force sensing register). By using them, Kenta can obtain information of contact to the environments. These sensors can be also used for direct teaching of posture. By touching the tactile sensors directly, human can teach postures and motions.

3 Design of the Software System Structure

In order to handle the 96 actuators and over 400 sensors, Kenta's control system is based on distributed system, that provides extensibility of sensor and actuator units. The left figure of the Fig. 10 shows the software system diagram of Kenta.

Layers: It consists of three layers. The bottom layer is the system in Kenta's body, and top two layers are in the remote host computer. The lowest layer is the onbody-LAN, which connects about 45 distributed microprocessors. Each microprocessors has the interface for a multi-master serial-bus called I²C-bus, 8 AD converters, 16 PWM generators. The middle layer was named nervous system and works as a hardware abstraction layer. The data flow between the middle layer and lower layer(onbody-LAN) is hardware-dependent; AD value of tension-sensors or counted pulse of motor-encoders. The interface between upper layer is body-independent; numerical values are in physical quantity. The top layer contains a geometric model of robot, motion data, and other high level softwares.

Communication System: The right figure of the Fig. 10 shows the onbody-LAN of Kenta. There are four subnets; hip-network, chest-and-head-network, arms-network, and legs-network. In each subnet there is a hub processor. Each hub processor collects sensor information from distributed processors connected to the subnet and sends them to the HostPC through high-speed serial line, and receive actuator-control information from the HostPC and distributes them to the dispersed processors through the onbody-LAN. We have developed some kinds of electric circuit boards to construct the onbody-LAN and distributed controllers (Fig. 10).

Hardware abstraction layer: The hardware abstraction layer called 'nervous system', the middle layer in the left figure of the Fig. 10, communicates with the hubs of the onbody networks. It is also a TCP/IP socket server (interface for the upper layer). When a request for connection from an upper layer (Software or character-UI), it opens a connection and starts to send/receive sensor/actuator information to/from the upper layer.

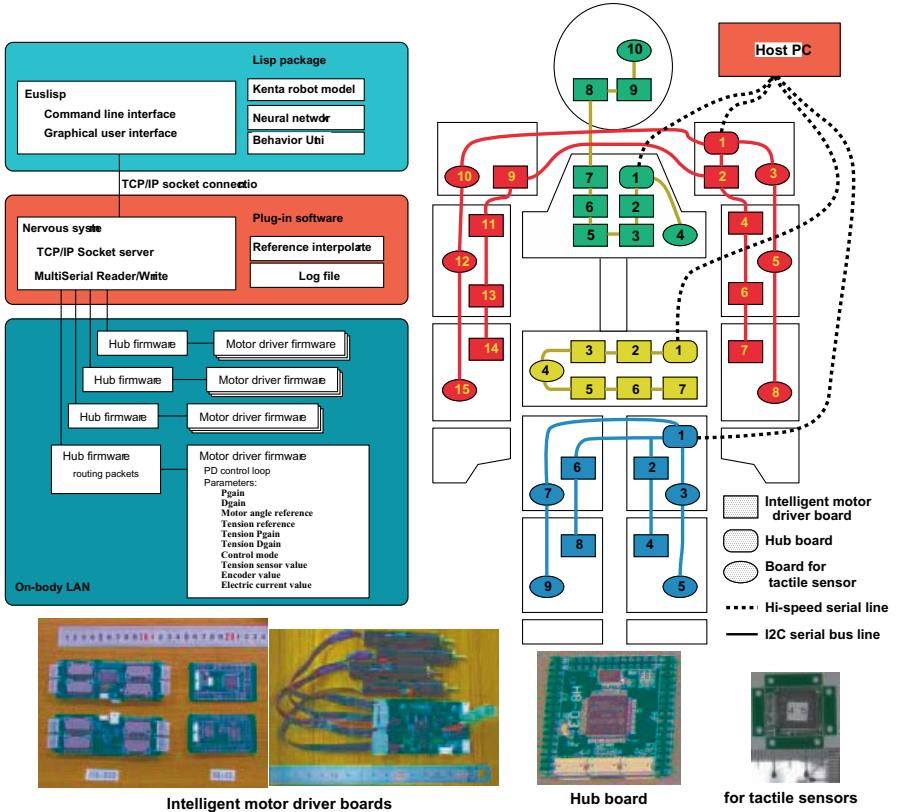


Fig. 10. Software system of Kenta (left: block diagram, right: the onbody-LAN, bottom: developed circuit boards)

Top level software: The geometric model of Kenta is implemented at the top layer of the three layers. By using the model, the calculation of muscle-length from posture of joints, the inverse calculation (posture of joints from muscle-length) using a neural network, etc. can be done. The geometric model of the robot is written on an object-oriented lisp, EusLisp (Matsui, Inaba, 1990). In the model, there are classes of ball-and-socket joint, rotational joint (knees and elbows), links, and wires (muscles). When rotating Kenta's joints in the model, the changed posture of Kenta is shown in the graphical window.

4 First steps to manage the body

4.1 Geometric model and muscle control modes

One method to manage the body is using a geometric model in conjunction with some control modes of muscles (Mizuuchi et al., 2001). The geometric model can

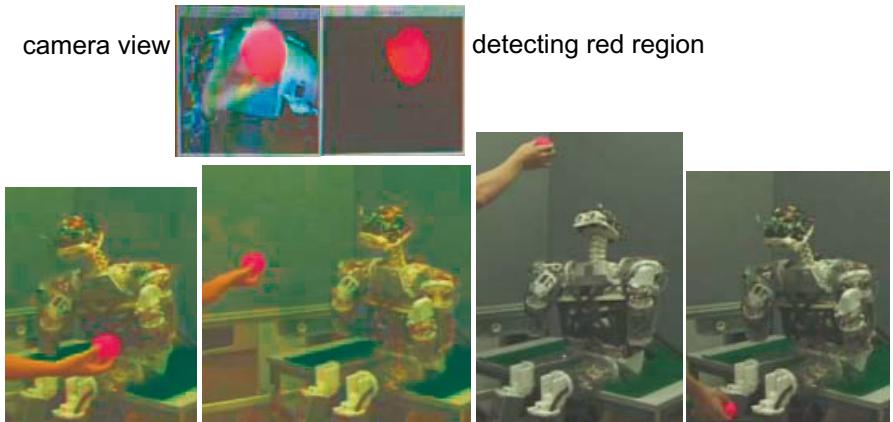


Fig. 11. Tracking an object by coordination of eye-balls, neck, and spine

approximate the lengths of muscles of any posture of the body. Depending on the posture, the error range of the muscles' lengths could be a certain extent. When there are some interferences between the muscles and bodies, if the muscles are controlled in the length-control mode, motors have to generate large torques. To avoid such case, muscles can be controlled in other control-mode using the tension-sensors' information. By using the information of tension sensors, muscle's tension can be limited. These control modes are implemented in the onbody microprocessors and the cycle of the control is 1 millisecond. In Fig. 11, Kenta is tracking a color object by coordination of eye-balls, neck, and spine. Vision processing of object-tracking is done in a remote host computer. Each of the eye-balls, neck and spine is independently tracking the object, and the three gains of the tracking-control are adjusted to balance the task.

4.2 Direct teaching

A simple and strong way to make whole-body motions of a complex structure is direct teaching. By controlling all muscles in the tension-control mode, each tension of them is kept in a constant value. When we directly change the posture of the robot in this condition, the lengths of all the muscles are modified to keep the tension. By recording the lengths while directly teaching, we can obtain the possible combinations of the lengths. At the stage of playing back, the muscles are controlled in the length-control mode, so as to reproduce the posture or the motion.

Figure 12 shows a motion using some directly taught postures. In (T1) and (T2) of the figure, two postures are being taught by a man touching directly. (P1), (P2) and (P3) of the figure shows three postures recorded at the teaching phase. (P2) and (P3) correspond with (T1) and (T2) respectively. By the combination of the postures directly taught, some motions are realized.

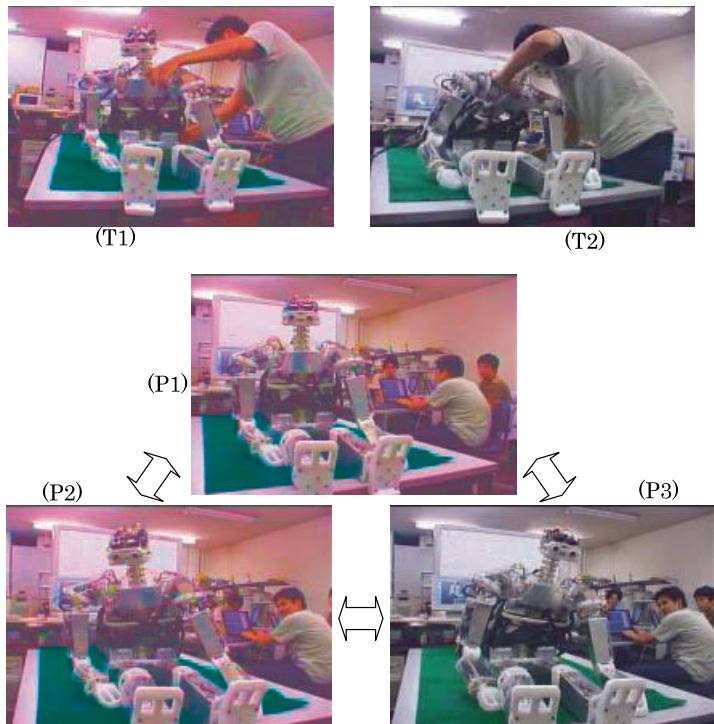


Fig. 12. Teaching postures and playing back them

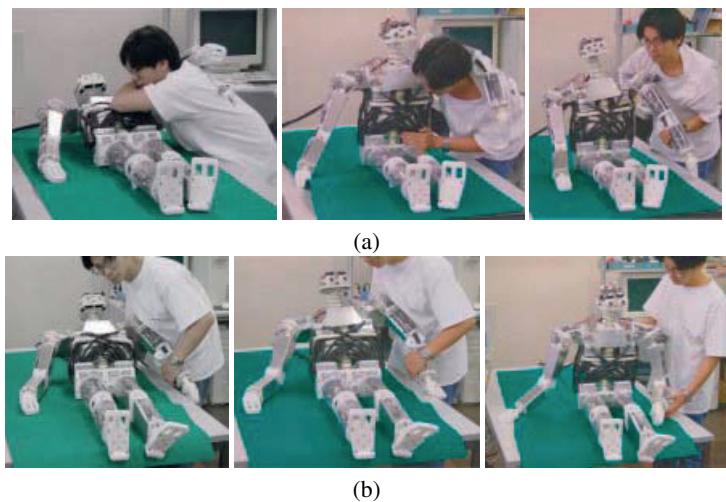


Fig. 13. Experiments of teaching movable space (a)passive mode (b) replay with partially passive mode

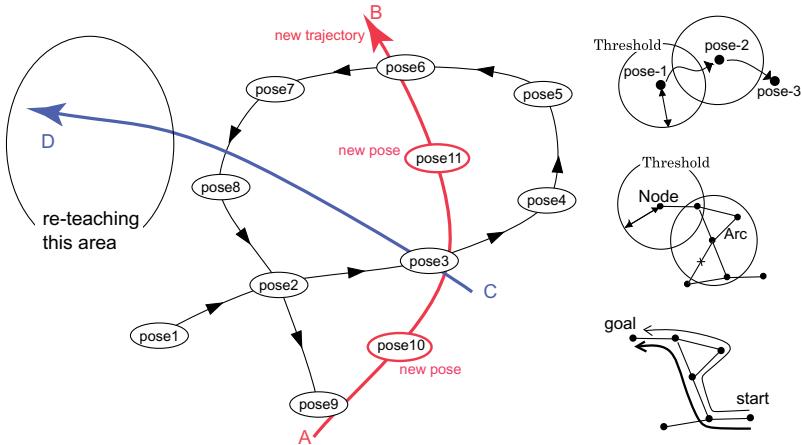


Fig. 14. Left: the concept of posture database, Right: (top: creating new nodes automatically using a threshold, middle: creating and trimming arcs, bottom: path-planning)

Figure 13 shows an experiment to teach getting up motion. The sequence (a) is the motion in fully passive mode. In the sequence (b), the robot follows the selected joints of the sequence (a).

4.3 Automatic expansion of motion space

It is quite difficult to decide the lengths of all muscles so as to take a posture, because the structure of Kenta's spine and whole-body is very complex. Even the geometric model described in Section 4.1 cannot derive the precise lengths of muscles. To help the directly teaching, we use the ‘posture database’. The left figure of the Fig. 14 shows the concept of it. This database is automatically created when the robot moves. The element in the database includes the information of the muscles’ lengths and 3D-accelerometers. The elements are segmented by the sensor information. When the distance between a data and each element of the database is larger than a threshold, a new node created by the data will be added to the database(see the top figure of the right side of the Fig. 14). In the left figure of the Fig. 14, pose1 to pose9 are automatically generated in some motion. Pose10 and pose11 are created in the other motion of A to B. The elements of the database are guaranteed to be a combination of the muscle-lengths without the interference. The amount of data which should be recorded can be lessen compared to recording the postures at all instances.

Figure 15 shows a whole-body motion including spine-motion, which is generated by direct-teaching and using the posture-database and posture-history.

We can use the postures of the database to develop motions and also obtain a new motions using the postures of the database. There are the arcs connecting nodes (elements of the database). The arcs are firstly defined as paths which robot

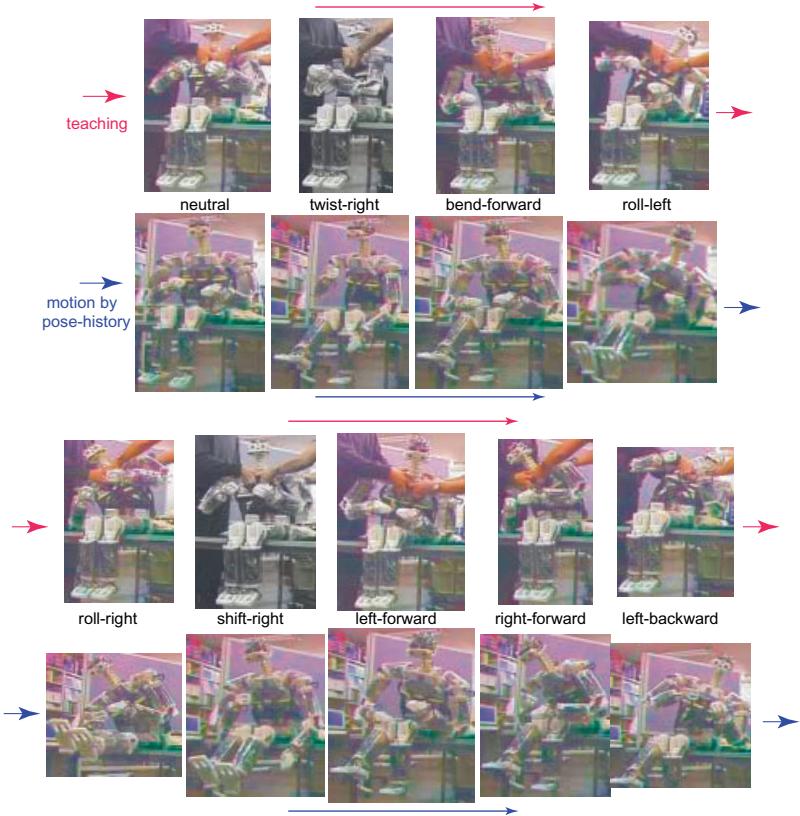


Fig. 15. A motion using pose history

once followed. There also can be new arcs which robot has not followed because there are some couple of nodes near enough to create a new arc (see the middle figure of the right side of the Fig. 14).

5 Summary and Concluding Remarks

This paper describes the design concept and its implementation of a spined muscle-tendon humanoid “Kenta”. First steps of controlling Kenta’s body including the ten-jointed spine are also described; calculation of muscle-length and joint-angle using a geometric model, realization of tracking motion, direct teaching and playback, and a method of automatic expansion of motion space.

In the future, humanoids will be expected more sorts of tasks and their bodies will be more complex. At that stage, the software which can manage the complex body and many various redundant incomplete information will be crucial. Human can manage his complex body using such huge information. Kenta can be one of

the testbeds for studying such kind of brain software. This research has been supported by Research for the Future Program of the Japan Society for the Promotion of Science: Micro-Mechatronics and Soft-Mechanics (JSPS-RFTF96P00801).

References

- Hirai K (1997) Current and future perspective of Honda humanoid robot. In Proc. of the IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS'97), vol 2, pp 500–508.
- Kato I, Ohteru S, Kobayashi H, Shirai K, Uchiyama A (1974) Information-power machine with senses and limbs (Wabot 1). In First CISM – IFToMM Symposium on Theory and Practice of Robots and Manipulators, vol 1, pp 11–24. Springer-Verlag.
- Matsui T, Inaba M (1990) Euslisp: An object-based implementation of lisp. *Journal of Information Processing*, 13(3):327–338.
- Mizuuchi I, Inaba M, Inoue H (2001) A flexible spine human-form robot – development and control of the posture of the spine. In Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol 1, pp 2099–2104.
- Nishiwaki K, Sugihara T, Kagami S, Kanehiro F, Inaba M, Inoue H (2000) Design and development of research platform for perception-action integration in humanoid robot : H6. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00), vol 3, pp 1559–1564.
- Okada K, Kagami S, Inaba M, Inoue H (2001) Plane segment finder: Algorithm, implementation and applications. In Proceedings of the 2001 IEEE International Conference on Robotics & Automation, pp 2120–2125, Seoul, Korea.
- Yamaguchi J, Inoue S, Nishino D, A.Takanishi (1998) Design and development of research platform for perception-action integration in humanoid robot : H6. In Development of a bipedal humanoid robot having antagonistic driven joints and three dof trunk, vol 1, pp 96–101.

Part 3

Dynamics and Control

Session Summary

Bernard Roth

Chair

Dept. of Mechanical Engineering, Stanford University, Stanford, CA, USA

Three papers are presented in this section. The first is concerned with the running of a hexapod robot. The second deals with dynamic walking of a quadruped robot. The final paper deals with both optimizing the geometry and the dynamic analysis of two three-degree-of-freedom in-parallel platform type manipulators. These three papers are largely unrelated to each other. However, there is a very important common thread in the first and second paper, where, in both cases, the authors comment upon the importance of designing their robots so that the passive characteristics of the mechanical structure contribute to the effective operation of the robot. This mindful design, of the mechanical structure to support the dynamic behavior of a robot, is by far the most effective and elegant way of obtaining good dynamic performance of all types of robotic systems.

In the paper “Stride Period Adaptation for a Biomimetic Running Hexapod,” Professor Cutkosky and his students present a detailed analytical and experimental study of the running performance of a small insect-like hexapod robot. They are interested in how the running speed and hop height is affected by stride frequency and actuation-thrust duration. They develop a control strategy that depends only upon binary sensing of each leg’s ground-contact-duration. Their approach is very robust in that their machine takes advantage of the passive properties of the structure to allow it to run over a wide range of open-loop stride frequencies and actuator duty cycles. One of the features of this robot is that it is manufactured by a shape deposition method, which easily allows for embedded actuators and purposefully compliant legs. The implemented control law attempts to maximize the stride frequency, and obtain the most useful work from the actuators without exceeding their bandwidth. The authors’ approach seems highly successful in allowing their robots to remain simple, inexpensive and robust while also being able to autonomously tune themselves to changes in operating conditions.

The paper “Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain by Using Neural System Model.” is a study in obtaining medium-speed mobility of a mammal-like quadruped on irregular terrain. The authors’ system depends upon a system model that they call a “neural system model”. This model relies on two key components: one is the central pattern generator (CPG) used to control the gate. The second key feature is the use of reflexes (flexor reflexes for stimuli to be avoided, and extensor reflexes to extend the leg in order to avoid falling down). The authors have designed a quadruped robot named Tekken, which weighs 3.1 kg, and has a body length of 40 cm and a height of about 20 cm. The paper reports on laboratory experiments at various walking speeds and in a trot gait. Good results, on uneven terrain, were obtain by virtue of the dynamic coupling of the neural and mechanical system that occurred when the robot interacted with the environment.

The paper “Design, Modeling and Application of Parallel Robots” considers the design and dynamic analysis of two in-parallel platform manipulators. Both manipulators are derived from the popular DELTA geometry. The DELTA is a three-degree-of-freedom manipulator in which a platform is translated by the action of three in-parallel legs, each driven by a motor with an axis located in the base plane. In the new configuration, called NUWAR (New University of Western Australia Robot) the motor axes are no longer coplanar. Using the motor-orientation angles as parameters, the author shows that it is possible to obtain configurations with better workspace volume than the standard DELTA. This paper then considers a DELTA type manipulator with three, parallel, prismatic joints in the fixed base. Brute force optimization is used to maximize a utility function that is the weighted sum of a manipulability index and a space utilization index. The final part of the paper deals with the application of Hamilton’s canonical equations to solving the forward and inverse dynamics of the NUWAR design. He uses the so-called extended space, which is the union of the joint and task space coordinates. The author explains why he believes that the use of extended space and Hamilton’s equations are more efficient than that of either the Lagrangian or Newton-Euler formulations.

Stride Period Adaptation for a Biomimetic Running Hexapod

Jonathan K. Karpick, Jorge G. Cham, Jonathan E. Clark, and Mark R. Cutkosky

Center for Design Research, Stanford University, Stanford CA 94305, USA

Abstract. We demonstrate an adaptation strategy for adjusting the stride period in a hexapedal running robot. The robot is inspired by discoveries about the self-stabilizing properties of insects and uses a sprawled posture, a bouncing alternating-tripod gait, and passive compliance and damping in the limbs to achieve fast, stable locomotion. The robot is controlled by an open-loop clock cycle that activates the legs at fixed intervals. For maximum speed and efficiency, this imposed stride period should be adjusted to match changes in terrain or loading conditions. An ideal adaptation strategy will complement the design philosophy behind the robot and take advantage of the self-stabilizing role of the mechanical system. In this paper we describe an adaptation scheme based on measurements of ground contact timing obtained from binary sensors on the robot's feet. We discuss the motivation for the approach, putting it in the context of previous research on the dynamic properties of running machines and bouncing multi-legged animals, and show results of experiments.

1 Introduction

We have built a family of hexapedal robots that are inspired by discoveries concerning the locomotion of insects and, in particular, of the cockroach. These animals run rapidly (between 10 and 50 body-lengths/second depending on the species) over rough terrain using a combination of open-loop muscle activation patterns and “preflexes,” i.e., passive mechanisms that stabilize the animals’ motion in response to perturbations (Ahn, Full, 1997)(Full, Autumn, Chung, Ahn, 1998)(Kubow, Full, 1999)(Meijer, Full, 2000). Like the insects that inspired them, the robots employ passive mechanical properties that enable them to run quickly (over 5 body-lengths per second) and over hip-height obstacles (see Fig. 1) without complex closed-loop control (Clark, Cham, Bailey, Froehlich, Nahata, Full, Cutkosky, 2001). Although this approach works well in the laboratory, there are questions about its versatility. How effectively can a particular open-loop control and set of mechanical properties function over a range of conditions which may include variations in ground slope and hardness and changes in loading? Furthermore, the animals or robots themselves may change over time. A limb may become damaged or the mechanical properties may vary with temperature. A way to address this problem is adaptation, in which the parameters of the predominantly open-loop control are automatically adjusted to optimize performance as conditions change.

Figure 2 illustrates an approach in which adaptation is combined with prefexes for stable running. An open-loop, feed-forward, motor controller generates the pattern of actuator commands to achieve a steady alternating-tripod gait. The kinematic

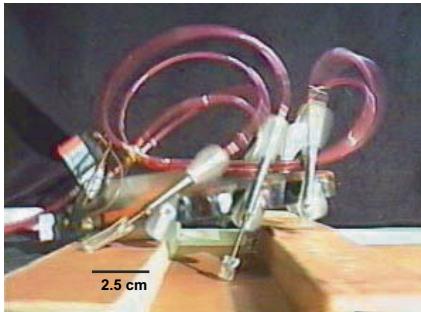


Fig. 1. The hexapedal robot, Sprawlita, has a body and legs fabricated by Shape Deposition Manufacturing (Clark et al., 2001) and features embedded actuators and compliant legs. Here the robot is crossing a belly-height obstacle without using sensory feedback and without significantly slowing down or being knocked off course

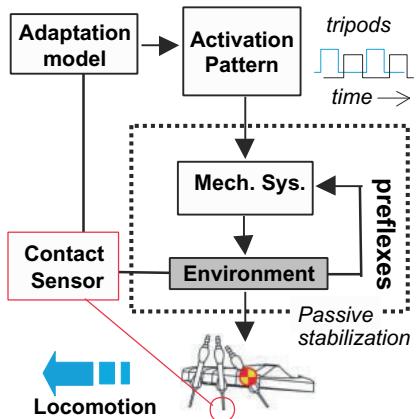


Fig. 2. A combination of stabilizing passive mechanisms, or “preflexes,” and sensor-based adaptation of an open-loop feed-forward controller provides insects and small robots with a robust, stable and versatile approach to running over rough terrain

arrangement and passive compliance and damping of the limbs achieve the locomotion and provide stable response to perturbations. In running insects, an important reason for employing slow adaptation in combination with preflexes is that neural conduction speeds are too slow for feedback control to act during each stride period. Robots, of course, do not necessarily have the same limitation. However, for small and inexpensive robots like Sprawlita, utilizing slow adaptation instead of active high-bandwidth control allows the use of simple sensors without concerns that actuator delays, sensor noise or even failures will jeopardize short-term performance. This is an important consideration because many sensors become noisy when mounted on a small hexapod running at 7-10 Hz.

The basic design of the Sprawl family of hexapods consists of a body and legs built up in layers using a rapid-prototyping process called Shape Deposition Manufacturing (Clark et al., 2001)(Bailey, Cham, Cutkosky, Full, 1999). Each leg has two degrees of freedom but only the thrust direction is actuated, using pneumatic pistons embedded into the legs. When running, hip rotations are passive and are accomplished by flexures of viscoelastic material. This design is inspired by the mostly passive trochanter-femur joint in cockroaches. A servomotor at each hip is used only to establish the equilibrium position of the hip joint. Binary contact sensors are attached to the feet. The original Sprawlita design uses two pneumatic valves, one for each tripod, embedded into the body of the robot. A newer design has a valve embedded into each leg. This arrangement results in faster actuator dynamics and more control over the timing of the thrust force at each foot. As will be seen in the

following sections, these are important considerations. Depending on configuration, the robots weigh between 0.25-0.33 Kg and have a length of 10-15cm. Maximum speeds range from 0.5-1.0m/sec with stride periods ranging from 100 to 150ms.

The locomotion parameters that can be varied are the stride period (length of time between activation of each tripod), the duty cycle (length of time that the valves are kept open during each stride) and the equilibrium positions of the compliant hip joints. All of these parameters contribute to running performance and could be subject to adaptation. In this paper, we focus on stride period and duty cycle.

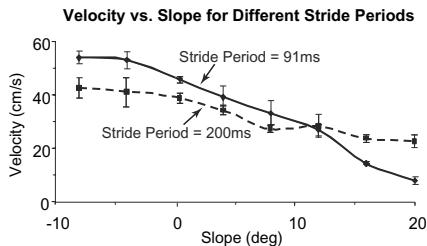


Fig. 3. Ground speed versus stride period on level ground versus slope. As shown, the optimal stride period for maximum speed depends on the slope, which illustrates the need for adaptation

Figure 3 illustrates the value of adjusting the stride period. On level ground the fastest locomotion is obtained with a period of approximately 100ms. On a 20 degree slope, locomotion is considerably faster at 200ms. The optimal stride period also varies somewhat from one robot to the next due to manufacturing tolerances and variations in the material properties of the legs. Consequently, there is motivation to make the robots “self-tuning” over a range of operating conditions. An ideal adaptation strategy would not require adding expensive or complicated sensors to the robot.

For these reasons, we examined the relationship between ground contact time (obtained from binary sensors in the robot’s feet) and the timing parameters (period and duty cycle) of the open-loop motor control. As will be discussed in Section 3, of the various timing quantities that we can examine, the interval between end of thrust (closing of the pneumatic valve) and liftoff of the feet provides a good indicator to use in adjusting the stride period. To understand why this approach works, we first examine a simplified one-legged vertical hopper model.

2 Simplified Model for Open-Loop Locomotion and Adaptation

To understand how monitoring the ground contact time can provide information about the effectiveness of running, we start with a simple vertical hopping model. Although this model cannot tell us about the coupling between vertical and horizontal motion, an important factor in the dynamics of the Sprawl robot family, it does shed light on the relationship between hopping height and actuator timing. It helps us to understand the circumstances under which a stable, steady-state hopping

height is achieved with an open-loop control scheme and how to get the most work out of our actuators using the simplest of sensors.

Variations on this basic model have been examined by Raibert (Raibert, 1986), Koditschek and Buehler (Koditschek, Buehler, 1991), Ringrose (Ringrose, 1997), Berkemeier and Desai (Berkemeier, Desai, 1998), Vakakis et al (Vakakis, Burdick, Caughey, 1991), and Komsuoglu and Koditschek (Komsuoglu, Koditschek, 2000). Despite its apparent simplicity, the one-legged vertical hopper exhibits a rich set of dynamic behaviors including stable and unstable periodic motion.

Raibert's hopper (Raibert, 1986) uses a double-acting pneumatic cylinder and an actuated hip to produce forward motion. The cylinder acts as an air spring when its valves are closed, providing a system similar to the simplified hopper of Fig. 4, although with a $1/r$ spring. Thrust is applied in a closed-loop fashion when the leg is sensed to be at maximum compression. Height is varied by changing the thrust duration. The stability of a simplified one-DOF hopper model based on this closed-loop thrust activation is analyzed by Koditschek and Buehler (Koditschek, Buehler, 1991). Vakakis et al (Vakakis et al., 1991) extended Koditschek and Buehler's analysis for finite, non-zero thrust durations.

Ringrose (Ringrose, 1997) designed a vertical hopper that maintains stable hopping without sensory feedback, using a linear actuator in series with a spring, and a damper in parallel with the actuator/spring combination. For analysis, Ringrose uses a simplified model in which thrust is applied through an impulsive change in leg length, which occurs before maximum compression of the spring.

Berkemeier and Desai (Berkemeier, Desai, 1998) compare Raibert's method of applying thrust at maximum compression, Ringrose's open loop control, and a proposed "adaptive periodic forcing" method that adjusts the period of the open loop control based on the velocity at the time that thrust is applied. In their analysis, they use a hopper with a spring and damper in parallel and an actuator that changes the neutral point of the spring. Their analysis shows that their hopper reaches maximum hopping height when the force is applied at maximum spring compression. They also show that, for low values of damping, the motion is stable, without feedback, when the force is applied prior to the maximum compression of the spring and unstable when the force is applied after the maximum compression.

Komsuoglu and Koditschek (Komsuoglu, Koditschek, 2000) analyze the stability of a similar open-loop one-DOF hopper in which thrust is effected by clock-driven changes in stiffness and find conditions for stability, which include the necessary presence of viscous friction.

The physical implementation of the Sprawl robots requires a review of the minimal damping assumptions made in other hopping models. In insects and in the Sprawl family of robots, viscoelastic materials dissipate substantial amounts of energy per cycle. Typical dimensionless damping ratios are on the order of $\zeta = 0.3$ (Garcia, Kuo, Peattie, Wang, Full, 2000).

We therefore consider a single-legged vertical hopper which can include substantial viscous damping. Figure 4 depicts a sample time history of this hopping model. The massless leg has stiffness, k , damping, b , and an actuator that is able to

provide a thrust force $f(t)$ that is initiated after some time t_f and terminated after a fixed duration, t_{on} , or by liftoff, whichever occurs first. The stride period begins at $t = 0$ as the robot touches down, which occurs when $y = 0$, the spring's neutral length. During the ground contact phase, the ground reaction force is given by:

$$GRF = ky + b\dot{y} - f(t)$$

The equation of motion for the mass is therefore:

$$m\ddot{y} = -ky - b\dot{y} - g - f(t)$$

Liftoff occurs when the ground reaction force is equal to zero, at which point the hopper transitions to an airborne phase, and travels ballistically. In the next two sections, we consider the conditions for optimal hopping height of this model, and conditions for the existence of stable behavior.

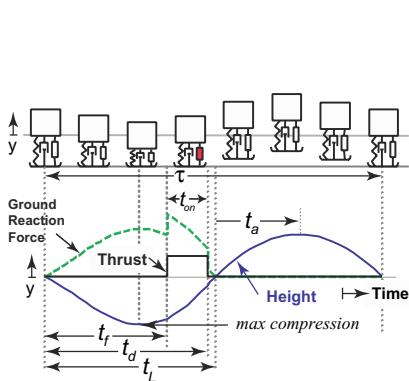


Fig. 4. Time history of a single degree-of-freedom vertical hopper. The mass is attached to a massless leg with stiffness k and damping b . At some time t_f , a thruster in parallel applies a force, F . At some time t_L , the mass lifts off the ground and travels ballistically in the air

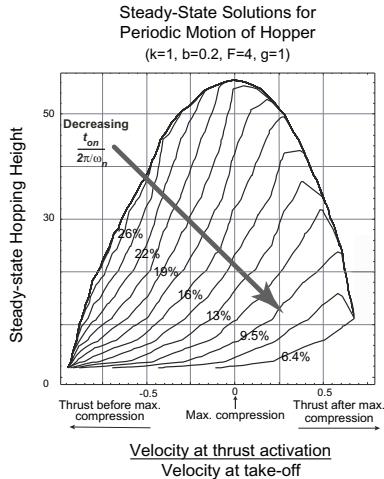


Fig. 5. Hopping height as a function of normalized velocity at thrust application for the one-DOF hopper. Each line represents the solutions to the steady-state constraint equations for a given set of operating parameters and as the thrust duration is varied

2.1 Optimal hopping height

In steady-state, the landing velocity of one cycle must equal the landing velocity of the previous cycle. Simulations were conducted to determine which values of force application delay, t_f , and force duration, t_{on} , meet steady state conditions for given

values of m , k , b , g , and force $f(t) = F$. For example, the steady state hopping height versus the velocity at which thrust is initiated for $m = g = k = 1$, $b = 0.2$, and $F = 4$ are shown in Fig. 5 for a range of thrust durations.

Figure 5 shows that for thrust duration times greater than 20% of the natural period, peak hopping height is achieved when the thrust is applied near the point of maximum spring compression, that is, when the velocity at activation is nearly zero. For shorter thrust durations, however, optimal steady-state hopping heights occur when thrust is initiated after maximum spring compression (velocity at force application is positive).

In evaluating the conditions that determined the maximum hopping heights in Fig. 5, it was found that for a given force level and duration, the maximum hopping height is obtained by maximizing the work added to the system by the actuator:

$$W = \int_{t_f}^{t_f + t_{on}} f(t) \cdot \dot{y}(t) dt$$

If $f(t)$ is constant, it is clear that for a long thrust that is applied until the end of the ground contact phase, this integral is maximized when thrust activation coincides with maximum compression. For a short thrust duration that ends before the end of the ground contact phase, the conditions for the maximization of this integral are more complex, but can be shown to roughly coincide with maximizing the upward velocity at thrust activation (while still achieving end-of-thrust before lift-off).

2.2 Stability and multiple solutions

The plot in Fig. 5 shows an example set of solutions to the steady-state constraint equations. With open-loop control, the system will either converge to one of the steady-state solutions from Fig. 5, converge to more complicated hopping patterns, or assume non-periodic behavior. In practice, as we varied the open-loop timing parameters, we found that the simulations converge to different steady-state behaviors, and often don't converge to periodic motion at all. To study the effects of changing the timing parameters on behavior and stability, we considered the local stability of the steady-state solutions.

The steady-state motion of the hopper model is determined by its return map:

$$X_{n+1} = R(X_n)$$

which relates the state after one cycle, X_{n+1} , to the previous state, X_n , for a given set of operating conditions (Sastry, 1999). Steady-state solutions satisfy:

$$X^* = R(X^*)$$

and are called “fixed points.” Although the equations of motion that govern the ground-contact and airborne phases are linear, the transitions between the two phases make the return map non-linear and, in this case, intractable to solve explicitly. Therefore, fixed points are found numerically. The local stability of each

existing solution is given by the linearized return map, a Jacobian matrix, M , defined as:

$$\delta X_{n+1} = M(X_n)\delta X_n$$

where,

$$M(X_n) = \frac{\partial R(X_n)}{\partial X_n}$$

If the eigenvalues of the matrix M evaluated at the fixed point are within the unit circle, then the fixed point is locally stable, since $M(X^*)$ maps disturbances about the fixed point from one cycle to the next (Sastry, 1999), i.e.:

$$|eig(M(X^*))| < 1$$

The return map and the matrix M were found analytically for our one-DOF model using only the simplifying assumption that lift-off occurs when $y = 0$. The two cases considered in this analysis were: a) when thrust application ends at or after lift-off (termed “Long Thrust”); and b) when thrust application ends before lift-off (termed “Short Thrust”). The analytical equations used in the following results are found in (Cha, Karpick, Clark, Cutkosky, 2002).

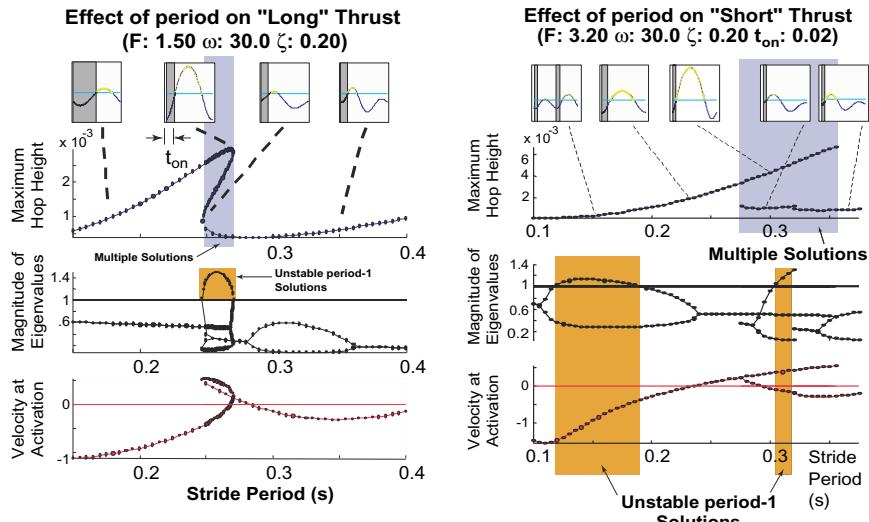


Fig. 6. Effects of changing the stride period on steady-state motion and stability for the one-DOF hopper case in which thrust application ends *at or after* lift-off

Fig. 7. Effects of changing the stride period on steady-state motion and stability for the one-DOF hopper case in which thrust application ends *before* lift-off

Figure 6 shows a typical example of the effects of changing the open-loop stride period for the “Long Thrust” case, for a given constant thrust magnitude, F , natural

frequency, ω_n , and damping ratio, ζ . For short periods, hopping height starts out very small, and thrust application starts well before maximum compression (given by the negative normalized velocity at application). These solutions are termed “Regular Hopping” as they represent a desired mode of hopping behavior. As the period is increased, hopping height increases, and velocity at application approaches zero. Finally, at a certain period (near 275ms), height is maximized when velocity at application is nearly zero (maximum spring compression). However, as the period approaches 275ms the magnitude of the eigenvalues quickly increases and the solutions become unstable. Simulations of the hopper, though, never reach this point. As shown in the figure, the chain of solutions “folds” back with respect to the stride period. As a result, three different hopping behaviors are possible in this range of stride periods. The second set of solutions with intermediate hopping height are characterized by thrust activation after maximum spring compression. These solutions are unstable, as determined by their corresponding eigenvalues. The third set of solutions is termed “hop-settle-fire” as the mass has started to settle before thrust is applied. The hopping height for this solution is much lower, and its corresponding eigenvalues are also much lower, and the simulations converge to this solution.

Figure 7 shows a typical example of the effects of changing the open-loop stride period for the “Short Thrust” case. For short periods, the solutions start out as “Regular Hopping.” However, as period is increased, the eigenvalues start to move outside the unit circle. Simulations for this range are “period-1 unstable” (the state does not repeat after one cycle), but tend to be “period-2 stable” (the state repeats after two cycles). As the period is further increased, the “Regular Hopping” solutions become period-1 stable again. The velocity at thrust activation also increases, and changes from negative (thrusting before maximum compression) to positive (thrusting after maximum compression). Maximum hopping height rises monotonically with stride period until the solutions end. However, another chain of valid steady-state solutions begins near 275ms. This is the “hop-settle-fire” solution of the “Long Thrust” case, where the hopping height is lower. Simulations converged to this solution because of its lower eigenvalues.

A rigorous analysis of the effects of the model’s other parameters on the hopping motion is beyond the scope of this paper, but they are nonetheless stipulated here from experience with the simulations. The onset of the “hop-settle-fire” solutions is determined largely by the system’s natural frequency. These solutions become available when the period of thrust application is long enough that the system is allowed to settle according to its natural period. The addition of damping also makes these solutions possible, since without damping the system would not settle. Decreasing damping and increasing the thrust magnitude and duration all seem to have a destabilizing effect, as they extend the region of unstable solutions.

2.3 One-DOF model conclusions

As discussed in Sect. 2.1, maximum height is achieved when the open-loop stride period is such that the positive work done by the thrust application is maximized. It

is worth noting that initiating thrust at maximum spring compression does not necessarily maximize the hopping height. In general, for both the “Long Thrust” and “Short Thrust” cases, maximum hopping height occurs when the open-loop stride period results in a stable solution in which thrust application starts “as late as possible” and ends shortly before or at lift-off. An adaptation strategy to increase hopping height would set the thrust application period according to these two general guidelines. However, as will be discussed in Sect. 3, optimizing hopping height in a multi-legged robot is not necessarily the same as optimizing ground speed, though similarities may be found.

3 Stride Period Adaptation

3.1 Robot performance tests

The one-DOF model provides insight into the basic behavior of an open-loop hopping system with passive properties in terms of the work performed by the actuator and the resulting performance. In order to develop an adaptation law for the six-legged, multi-DOF robot, we must look at the factors that affect its performance and see whether the same basic mechanisms are evident. Figure 8 shows the performance results of the hexapedal robot as a function of open-loop stride period on flat ground. As shown in Fig. 8a, increasing the stride period causes the amplitude of vertical oscillations, $\Delta height$, to increase, similar to the one-DOF model. Figure 8b shows the difference between the time that a particular tripod is deactivated, t_d , and the time that the middle-foot of that tripod leaves the ground, t_L . This time lag, $(t_d - t_L)$, is a nearly constant small value for periods shorter than 130ms. Above 130ms, this time lag changes slopes and starts to increase, which indicates that thrust application ends near liftoff, here caused by the end of the stroke. Running speed is a maximum near the point where this change in slope occurs. For longer stride periods, this quantity monotonically increases since thrust application, t_{on} , is set as a fixed percentage, or duty cycle, of the stride period.

Figure 8a shows that stride period plays a significant role in determining the ground speed of Sprawlita. Shorter stride periods cause faster running until performance is limited by the capabilities of the actuators. As shown in Fig. 8b, the time lag, $(t_d - t_L)$, approaches zero as the period decreases, indicating that thrust application is being terminated closer to the time that end-of-stroke occurs. Below the optimal stride period the time lag remains nearly constant since ground contact is lost due to leg retraction when the valves are deactivated. This correlation between optimal speed and the change in slope of the time lag, $(t_d - t_L)$, is used as a basis for the simple adaptation law described in the following section.

3.2 Adaptation strategy

The results from the previous sections motivate the robot stride period adaptation strategy described here. As illustrated by the one-DOF model, it is advantageous

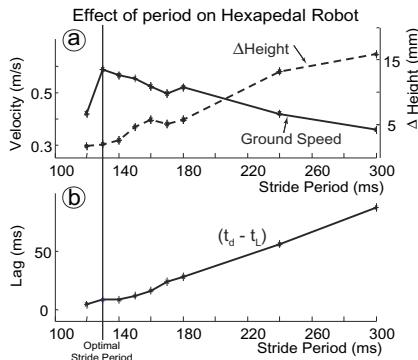


Fig. 8. Performance tests for the hexapedal robot in flat ground as a function of open-loop stride period with constant duty cycle percentages

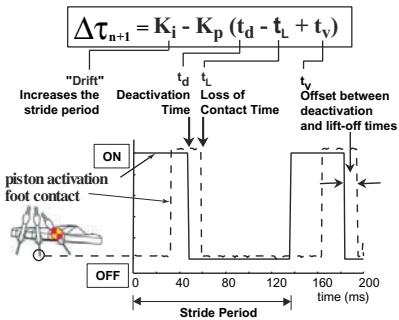


Fig. 9. Simple stride period adaptation law based measured duration between foot contact events. A binary switch in the robot's middle foot provides contact information. This time duration is compared to the actuator valve deactivation time for the adaptation law

to decrease the stride period while trying to have the thrust end “late” in order to maximize work input per stride. Moreover, due to the limitations of the pneumatic system, stride period must be limited in order to maintain full stroke.

A prototype adaptation law for maximizing ground speed that takes these findings into consideration using foot contact information is as follows:

$$\tau_{n+1} = \tau_n + K_i - K_p(t_d + t_v - t_L)$$

Here, K_i and K_p are adaptation gains, t_v is a constant offset parameter and t_d and t_L are the measured valve deactivation and lift off times. Figure 9 illustrates what these quantities represent, where time is measured with respect to the initiation of the gait cycle, which starts when the valve for one of the tripods is activated. The adaptation law is based on a binary switch attached to the middle foot of the same tripod; t_d is determined by the stride period, τ_n , and duty cycle, t_{on} , which in this case is specified as a fixed percentage of the stride period. If there is no measured ground contact information, then the period is not modified.

Intuitively, this simple adaptation law can be described as trying to decrease the stride period as much as possible without terminating the thrust application before end-of-stroke (to maximize available work). Here, K_i is a “drift” term which is trying to increase the stride period at every step, since the difference between t_d and t_L is nearly constant for lower periods. The stride period reaches an equilibrium value when $\Delta\tau$ is zero, which occurs when $(t_d - t_L)$ reaches a nominal value. In this case, t_v is adjusted so that the equilibrium stride period maximizes ground speed.

3.3 Adaptation Results and Discussion

Figure 10 shows test results of the adaptation law implemented in the hexapedal robot running on flat ground. Each data point in Fig. 10a represents the new stride period after one stride cycle, or learning loop, in which t_L (shown in Fig. 10b) was measured. The parameter t_v was chosen such that the equilibrium period coincided with the optimal period for flat ground found in Section 2. The gains K_i and K_p were experimentally chosen to give the adaptation a fast learning rate without causing instabilities. As shown in the figure, for starting periods that are lower than the equilibrium period, the learning rate is approximately linear, due to the predominance of the K_i term. For higher starting periods, however, the K_p term dominates and the learning rate is approximately exponential. Note from Fig. 10b that, although only a simple contact switch was used, the measured values of t_L are still prone to some noise, due to ground imperfections or disturbances to the robot. Despite these variations, the adaptation of the stride period still proceeds smoothly, and does not affect the stabilization performed by the mechanical system throughout a particular stride.

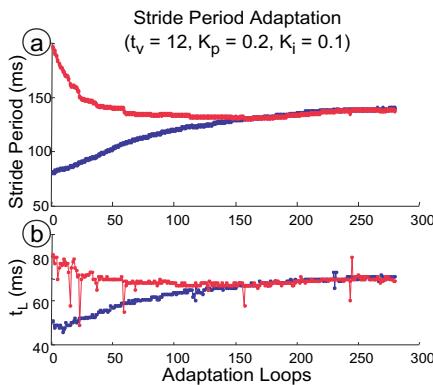


Fig. 10. Adaptation results for stride period. (a) shows the adaptation of the stride period from two initial starting values. (b) shows the corresponding measured values of t_L for each stride period

For an uphill ground slope of 10 degrees, the adaptation strategy also converges to an equilibrium stride period. This new equilibrium period (165ms) is higher than the equilibrium period for flat ground running (130ms) and results in faster uphill running than with the optimal period for flat ground. However, the new equilibrium period is still lower than the period found to be optimal at 10 deg. slope (190ms). This indicates that, although the adaptation strategy works to improve the locomotion, the factors that affect uphill running may need to be re-examined. For example, in seeking to decrease the stride period for optimal ground speed in flat terrain, the prototype adaptation law presented here reduces vertical oscillations, which we postulate will have a significant role in climbing up-hill terrain.

4 Conclusions and Future Work

The analyses and experiments in the previous sections show that for an open-loop running robot, stride period and thrust duration are important parameters that govern hop height and forward speed. The single legged hopper model reveals that optimal hop height is obtained by maximizing the product of thrust force and velocity over the thrust duration. However, this product is subject to both dynamic constraints and hardware limitations. The dynamic constraints include the requirement of a stable, steady-state periodic solution to which the system will converge. Significant passive damping, as found in insects and in robots like Sprawlita, increases the regime of stable, periodic operation with open-loop forcing. The hardware limitations include the speed at which the piston can be filled and exhausted and the maximum thrust force available.

A simple adaptation strategy for the stride period that takes these limitations into account and tries to optimize ground speed was presented in this paper. The adaptation law seeks to minimize the stride period, and obtain the most work from the actuators without exceeding their bandwidth. This adaptation law uses only the sensed duration of ground contact during each stride, and was shown to cause the stride period to converge and to improve the speed even over changes in terrain and despite disturbances to the locomotion. More generally, the adaptation scheme presented in this paper is an example of an approach that is particularly well suited for small, biomimetic robots. It requires no expensive or sophisticated sensing or feedback. Only binary sensors are needed to provide an estimate of ground contact time. The adaptation scheme takes advantage of the passive properties of the robot that allow it to run stably over a range of open-loop stride frequencies and actuator duty cycles. In the event of sensor failure, the performance of the robot degrades only to that of the open-loop system without adaptation. This approach allows the robots to remain simple, inexpensive and robust while also being able to “tune” themselves to accommodate individual variabilities and changes in operating conditions.

Future work will build upon the simple adaptation law tested in this paper to incorporate other simple sensor information (e.g. tilt sensor, contact switches in other feet) in order to increase performance and adaptability. As discussed previously, further understanding of the robot’s dynamic interaction with different types of terrain such as sloped or compliant surfaces will allow us to increase the adaptation’s versatility. Finally, future work will study the effects of such an adaptation law on other types of behavior, such as rapid turning and navigation.

Acknowledgments

We have been fortunate to collaborate with Professor Robert J. Full at the U.C. Berkeley Polypedal Laboratory throughout this work and to benefit from his prodigious knowledge and understanding of the mechanisms underlying insect locomotion. We are also grateful for lengthy discussions on adaptation with Dr. Goran Djordevic at the Johns Hopkins’ Laboratory for Human Motor Learning. Thanks

also to Sean Bailey, Roman Devengenzo, Edward Froehlich and the other members of the Biomimetics team at CDR. Jonathan Karpick is supported by an Alex and Brit d'Arbeloff Stanford Graduate Fellowship. Funding was provided by the Office of Naval Research under grant N00014-98-1-0669.

References

- Ahn AN, Full RJ (1997) A motor and a brake: Similar EMGs in two adjacent leg extensor muscles result in completely different function. *American Zoologist*, 37:107A.
- Bailey SA, Cham JG, Cutkosky MR, Full RJ (1999) Biomimetic mechanisms via shape deposition manufacturing. In Hollerbach J, Koditschek D (eds) *The 9th International Symposium of Robotics Research*, pp 403–410. Springer Verlag.
- Berkemeier MD, Desai KV (1998) A comparison of three approaches for the control of hopping height in legged robots. Submitted to the *International Journal of Robotics Research*.
- Cha JG, Karpick JK, Clark JE, Cutkosky MR (2002) Stride period adaptation for a biomimetic running hexapod. Submitted to the *International Journal of Robotics Research* for review.
- Clark JE, Cham JG, Bailey SA, Froehlich EM, Nahata PK, Full RJ, Cutkosky MR (2001) Biomimetic design and fabrication of a hexapedal running robot. In *IEEE International Conference on Robotics and Automation*, pp 3643–3649.
- Full RJ, Autumn K, Chung JI, Ahn A (1998) Rapid negotiation of rough terrain by the death-head cockroach. *American Zoologist*, 38:81A.
- Garcia M, Kuo A, Peattie AM, Wang PC, Full RJ (2000) Damping and size: Insights and biological inspiration. In *Proc. of the Intl. Symp. on Adaptive Motion of Animals and Machines*.
- Koditschek DE, Buehler M (1991) Analysis of a simplified hopping robot. *International Journal of Robotics Research*, 10:587–605.
- Komsuoglu H, Koditschek DE (2000) Preliminary analysis of a biologically inspired 1-DOF ‘clock’ stabilized hopper. In *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, pp 670–675.
- Kubow TM, Full RJ (1999) The role of the mechanical system in control: A hypothesis of self-stabilization in hexapedal runners. *Phil. Trans. R. Soc. Lond.*, 354:849–861.
- Meijer K, Full RJ (2000) Stabilizing properties of invertebrate skeletal muscle. *American Zoologist*, 39(117a).
- Raibert MH (1986) Legged robots that balance. MIT Press, Cambridge, MA.
- Ringrose R (1997) Self-stabilizing running. In *IEEE International Conference on Robotics and Automation*, pp 487–493. OmniPress.
- Sastry S (1999) Nonlinear systems: Analysis, stability and control. Springer Verlag, New York.
- Vakakis AF, Burdick JW, Caughey TK (1991) An interesting strange attractor in the dynamics of a hopping robot. *International Journal of Robotics Research*, 10:606–618.

Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Using a Neural System Model

Hiroshi Kimura¹, Yasuhiro Fukuoka¹, Yoshiro Hada¹, and Kunikatsu Takase¹

Univ. of Electro-Communications, 1-5-1 Chofu-ga-oka, Chofu, Tokyo 182, JAPAN

Abstract. We have been trying to induce a quadruped robot to walk dynamically on irregular terrain by using a neural system model consisting of a CPG (central pattern generator) and reflexes. In this paper, we define adaptive walking using a neural system model as “coupled-dynamics-based motion generation”, in which a neural system and a mechanical system are coupled and generate motion by interacting with the environment emergently and adaptively. In order to clarify how a CPG is coupled to a mechanical system, we use simulations to investigate the relationship between the parameters of a CPG and the dynamics of the mechanical system. We propose the essential conditions for stable dynamic walking on irregular terrain in general, and we design a neural system by comparing biological knowledge with the essential conditions described in physical terms. We report our experimental results of dynamic walking on terrains of medium degrees of irregularity using a planar quadruped robot and a newly developed three-dimensional quadruped robot. MPEG footage of these experiments can be seen at: <http://www.kimura.is.uec.ac.jp>.

1 Introduction

Many previous studies of legged robots have been performed, including studies on running and dynamic walking on irregular terrain. However, studies of autonomous dynamic adaptation allowing a robot to cope with an infinite variety of terrain irregularities have been started only recently and by only a few research groups. One example is the recent achievement of high-speed mobility of a hexapod over irregular terrain, with appropriate mechanical compliance of the legs (Cham, Bailey, Cutkosky, 2000; Saranli, Buehler, Koditschek, 2000). The purpose of this study is to realize high-speed mobility on irregular terrain using a mammal-like quadruped robot, the dynamic walking of which is less stable than that of hexapod robots, by referring to the marvelous abilities of animals to autonomously adapt to their environment.

As many biological studies of motion control progressed, it has become generally accepted that animals' walking is mainly generated at the spinal cord by a combination of a CPG (central pattern generator) and reflexes receiving adjustment signals from a cerebrum, cerebellum and brain stem (Grillner, 1981). A great deal of the previous research on this attempted to generate walking using a neural system model, including studies on dynamic walking in simulation (Ijspeert, 2001; Taga, 1995; Taga, Yamaguchi, Shimizu, 1991), and real robots (Ilg, Albiez, Jede, Berns, Dillmann, 1999; Kimura et al., 1999a; Lewis, Etienne-Cummings, Cohen, Hartmann, 2000; Tsujita, Tsuchiya et al., 2000). But autonomously adaptable dynamic walking on irregular terrain was rarely realized.

At ISRR99, the authors presented biologically inspired control of a planar quadruped robot called 'Patrush' (Kimura, Fukuoka et al., 1999b). In this paper, we describe our progress in the past two years using Patrush and a newly developed quadruped called 'Tekken,' which contains a mechanism designed for 3D space walking (pitch, roll and yaw planes) on irregular terrain.

In this paper we would like to emphasize three key concepts. First is the coupled-dynamics-based motion generation for autonomous adaptation, described in Section 2.1. Second are the essential conditions for stable dynamic walking on irregular terrain, described in Section 2.4. Third is the sensorimotor interaction between CPGs and reflexes described in Section 2.4, 3.2 and 4 in order to satisfy those essential conditions. Those key concepts are common to both animals and machines in spite of the differences in their mechanisms, actuators, sensors and so on.

2 Adaptive Dynamic Walking Using a Neural System Model

2.1 Coupled-dynamics-based motion generation

Motion generation using a neural system model is illustrated as Fig. 1-(a), where a neural system and a mechanical system each have their own non-linear dynamics. The characteristic of this method is that there is no motion planning. These two dynamic systems are coupled to each other, generating motion by interacting with the environment emergently and adaptively (Taga, 1995; Taga et al., 1991). We call this method "coupled-dynamics-based motion generation."

On the other hand, a control system in conventional robotics, such as impedance control, also constructs a coupled dynamic system with a mechanical system (Fig. 1-(b)). However, the motion planner is outside of the coupled dynamic system. Basic motion generation and motion adaptation based on sensor information should be described as algorithms considering the dynamics of lower systems.

The most important subject in coupled-dynamics-based motion generation is to design and construct a neural system carefully, while taking into account the dynamics of a mechanical system and its interaction with the environment. In this study, we designed a neural system consisting of CPGs and reflexes, while taking into account the characteristics of dynamic walking and utilizing knowledge and concepts in physics, biology, physiology and so on, as described in Section 2.3 and 2.4. The relationship between parameters of CPGs and the mechanical system was previously analyzed to some extent (Kimura, Fukuoka et al., 2001) and is described briefly in Section 3.1. Since the relationship between the reflex parameters and the mechanical system has not yet been investigated, we manually determined reflex parameter values through experiments. The issue of how to construct a neural system suitable for a mechanical system corresponds to the issue of "embodiment" at the lowest level of sensorimotor coordination.

2.2 Biological concepts of legged locomotion

Methods for legged locomotion control are classified into ZMP-based control and limit-cycle-based control (Table. 1). ZMP (zero moment point) is the extension of

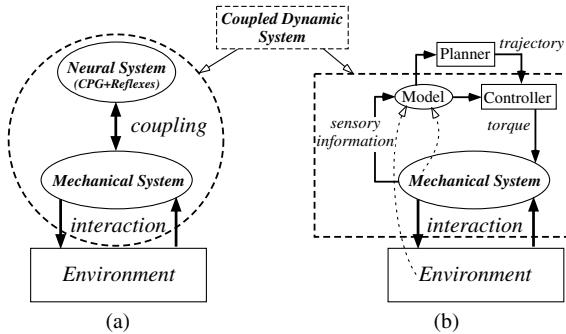


Fig. 1. Illustration of motion generation by a neural system: (a) and a conventional robotic system: (b).

Table 1. Biological concepts of legged locomotion control.

	ZMP based	Limit Cycle based	
		by Neural System (CPG and reflexes)	by Mechanism (spring and damper)
good for control of	posture and low speed walking	medium speed walking	high speed running
main controller	upper neural system acquired by learning	lower neural system (at spinal cord, brain stem, etc.)	musculoskeletal system through self stabilization

the center of gravity considering inertia force and so on. It was shown that ZMP-based control is effective for controlling posture and low-speed walking of a humanoid (Yamaguchi, Kinoshita, Takanishi, Kato, 1996). However, ZMP-based control is not good for medium or high-speed walking from the standpoint of energy consumption, since a body with a large mass needs to be accelerated and decelerated by actuators in every step cycle.

In contrast, motion generated by the limit-cycle-based control has superior energy efficiency. But there exists the upper bound of the period of the walking cycle, in which stable dynamic walking can be realized (Kimura, Shimoyama, Miura, 1990). It should be noted that control by a neural system consisting of CPGs and reflexes is dominant for various kinds of adjustments in medium-speed walking of animals (Grillner, 1981). Full and Koditschek also pointed out that, in high-speed running, kinetic energy is dominant, and self-stabilization by a mechanism with a spring and a damper is more important than adjustments by the neural system (Full, Koditschek, 1990). Our study is aimed at medium-speed walking controlled by CPGs and reflexes (Table. 1).

2.3 Biological concepts of dynamic walking

As a model of a CPG, we used a neural oscillator: N.O. proposed by Matsuoka (Matsuoka, 1987), and applied to the biped simulation by Taga (Taga, 1995; Taga et al., 1991). A single N.O. consists of two mutually inhibiting neurons (Fig. 2-(a)). Each neuron in this model is represented by the following nonlinear differential equations:

$$\begin{aligned} \tau \dot{u}_{\{e,f\}i} &= -u_{\{e,f\}i} + w_{fe}y_{\{f,e\}i} - \beta v_{\{e,f\}i} \\ &\quad + u_0 + Feed_{\{e,f\}i} + \sum_{j=1}^n w_{ij} y_{\{e,f\}j} \\ y_{\{e,f\}i} &= \max(u_{\{e,f\}i}, 0) \\ \tau' \dot{v}_{\{e,f\}i} &= -v_{\{e,f\}i} + y_{\{e,f\}i} \end{aligned} \quad (1)$$

where the suffix e , f , and i mean an extensor neuron, a flexor neuron, and the i -th N.O., respectively. $u_{\{e,f\}i}$ is u_{ei} or u_{fi} , that is, the inner state of an extensor neuron or a flexor neuron of the i -th N.O.; $v_{\{e,f\}i}$ is a variable representing the degree of the self-inhibition effect of the neuron; y_{ei} and y_{fi} are the output of extensor and flexor neurons; u_0 is an external input with a constant rate; $Feed_{\{e,f\}i}$ is a feedback signal from the robot, that is, a joint angle, angular velocity and so on; and β is a constant representing the degree of the self-inhibition influence on the inner state. The quantities τ and τ' are time constants of $u_{\{e,f\}i}$ and $v_{\{e,f\}i}$; w_{fe} is a connecting weight between flexor and extensor neurons; w_{ij} is a connecting weight between neurons of the i -th and j -th N.O..

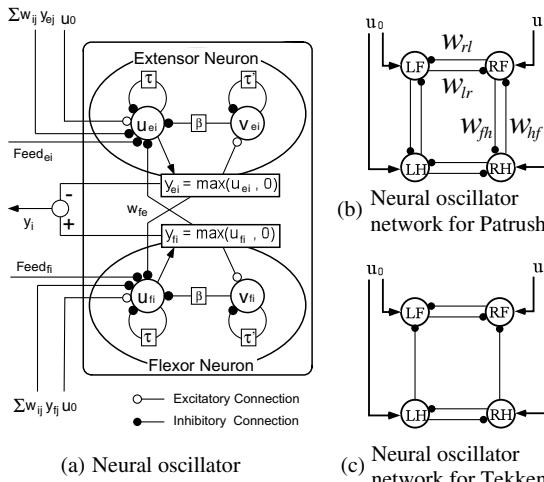


Fig. 2. Neural oscillator as a model of a CPG. The suffix $i, j = 1, 2, 3, 4$ corresponds to LF, RF, RH, L, R, F or H means the left, right, fore or hind leg, respectively.

In Fig. 2-(a), the output of a CPG is a phase signal: y_i .

$$y_i = -y_{ei} + y_{fi} \quad (2)$$

The positive or negative value of y_i corresponds to activity of a flexor or extensor neuron, respectively. We use the following joint angle feedback as a basic sensory input to a CPG in all experiments of this study. This negative feedback makes a CPG be entrained with a rhythmic joint motion.

$$Feed_{e\cdot tsr} = k_{tsr}\theta_{hip}, \quad Feed_{f\cdot tsr} = -k_{tsr}\theta_{hip} \quad (3)$$

where we eliminate the suffix i when we consider a single N.O.

By connecting a CPG of a hip joint of each leg (Fig. 2-(b),(c)), CPGs are mutually entrained and oscillate in the same period and with a fixed phase difference. This mutual entrainment between the CPGs of the legs results in a gait. The gait is a walking pattern, and can be defined by phase differences ($0 \sim 1$) between the legs during their pitching motion. When we use γ for the phase difference from the left hindleg to the left foreleg, the typical symmetric gaits are a trot ($\gamma = 0.5$) and a pace ($\gamma = 0$). Diagonal legs and lateral legs are paired and move together in a trot gait and a pace gait, respectively. A walk gait is the transversal gait between the trot and pace gaits.

Since there isn't much meaning in a gait of planar walking, the gait of Patrush was fixed to a trot gait using the symmetric CPG network: Fig. 2-(b). However, since a gait is one of important subjects in 3D walking using Tekken, the asymmetric CPG network: Fig. 2-(c) was used in order to generate different gaits using a single network configuration.

2.4 Biological concepts of adaptive walking

The essential conditions for stable dynamic walking on irregular terrain can be itemized in physical terms:

- (a) the swinging legs should be free to move forward during the first period of the swinging phase,
- (b) the swinging legs should land reliably on the ground during the second period of the swinging phase,
- (c) the angular velocity of the supporting legs should be kept constant during their pitching motion around the contact points at the moment of landing or leaving, and
- (d) the phase differences between the legs should be maintained regardless of delay in the motion of a leg receiving a disturbance from irregular terrain.

It is well known in physiology that some sensory stimulus modifies CPG activity and that reflexive responses to a sensory stimulus are phase dependent under CPG activity (Matsuoka, 1999). Such interaction between CPG activity and a sensory stimulus is very important for adaptation and corresponds to the essential conditions

described above in physical terms. For example, the reflex to a stimulus on the paw dorsum in the walking of a cat depends on whether flexor or extensor muscles are active (Matsuoka, 1999). That is,

- [a] when flexor muscles are active, the leg is flexed in order to escape from the stimulus,
- [b] when extensor muscles are active, the leg is strongly extended in order to prevent the cat from falling down.

We call [a] and [b] above the 'flexor reflex' and the 'extensor reflex', respectively, and we assume that the phase signal from the CPG of the leg switches such reflexes. The roles of the flexor and extensor reflexes correspond exactly to conditions (a) and (b), respectively.

On the other hand, Pearson (Hiebert, Pearson, 1999) pointed out that increasing the load on a supporting leg strengthens the supporting leg through the excitatory input to an extensor neuron, and also extends the stance phase through the inhibitory input to a flexor neuron. We call this the 'tendon reflex'. It complements the thrusting force on a supporting leg in order for condition (c) to be satisfied, and adjusts the period of a stance phase in response to a delay in the motion of the supporting leg, in order for condition (d) to be satisfied.

In addition, the following biological concepts are known:

- When the vestibule in a head detects an inclination in pitch or roll plane, a downward-inclined leg is extended while an upward-inclined leg is flexed (a 'tonic labyrinthine reflex');
- Asymmetry exists in connections of the CPG network; that is, ascending is strong and descending is weak.

The roles of a tonic labyrinthine reflex correspond to condition (c) for an inclination in the pitch plane and to condition (b) for an inclination in the roll plane.

All those biological concepts are used to design a neural system for adaptive walking in following sections.

3 Planar Walking

In this section, we consider the relationship between the CPGs and the pitching motion of the legs. In the case of planar walking by Patrush, a CPG outputs torque in proportion to the output of extensor and flexor neurons to a DC motor of a joint:

$$trqi = -p_e y_{ei} + p_f y_{fi} \quad (4)$$

A positive or negative value of $trqi$ corresponds to the activity of a flexor or extensor muscle, respectively. Since the output of a CPG is torque, (3) means a stretch reflex generated via a CPG. As a result of this stretch reflex via a CPG, the CPG and the hip joint motion are mutually entrained, and can generate stable walking on flat terrain (Kimura et al., 1999a). This concept of mutual entrainment between the CPG and the mechanical system was proposed by Taga (Taga, 1995; Taga et al., 1991), in which the CPG acts as both a rhythm generator and an α motor neuron in animals.

3.1 Coupling of the dynamics of the CPG with the dynamics of the mechanical system

One of the reasons why such a simple combination of CPGs and a stretch reflex can generate dynamic walking is that the dynamics of the mechanical system is encoded into the parameters of the CPG. In order to clarify how the CPG is coupled with the mechanical system, we investigated the relationship between the CPG parameters and the mechanical system by simulations using the Patrush model (Kimura et al., 2001), mainly from the standpoint of the period of the walking cycle.

As a result, the time constant of CPG: τ in (1) should be determined as the cyclic period of the CPG in free oscillating as being equal to the free cyclic motion period of the pendulum as a model of a swinging leg, for stable walking with a smaller period of the walking cycle and for efficient walking with less acceleration and deceleration.

3.2 Reflexes via CPGs

Based on the biological concepts described in Section 2.4, we employed reflexes generated via CPGs as shown in Fig. 3, where the reflex torque is outputted as part of the CPG torque through the feedback of all sensory information to the CPG. We considered five types of reflexes, shown in Table. 2, in response to the somatic sensations and the vestibular sensations. The meanings of reflexes in Table. 2 are described in 2.4 with being related to conditions for stable dynamic walking on irregular terrain.

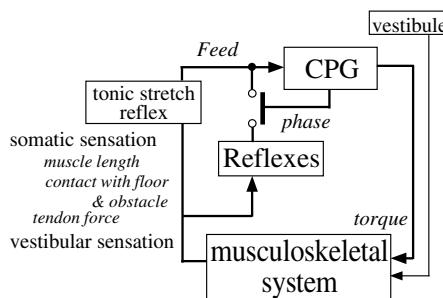


Fig. 3. Concept of reflexes via CPGs.

The joint motion of the leg is delayed as a result of adaptation to irregular terrain through the reflexes. If the joint motion is much delayed, the CPG of the leg should extend its current phase in order to maintain the entrainment with joint motion. When we employ reflexes via CPGs, the positive feedback signal in Table. 2 to an active neuron of a CPG extends the active period of the neuron. That is, the current phase of the CPG is extended, and the entrainment between the CPG and the joint motion can be maintained autonomously.

Table 2. Reflexes via CPGs (Kimura et al., 2001). The stretch reflex acts as negative feedback, and other reflexes act as positive feedback.

Reflexes	switched on	$Feed_{\{e,f\}}$
stretch reflex	always	$Feed_{\{e,f\}\cdot tsr} = \pm k_{tsr} \theta_{hip} : (3)$
tonic labyrinthine reflex for pitching	always	$\theta_{tlrp} = \theta_{hip} - (\text{body pitch angle})$ $Feed_{\{e,f\}\cdot tsr\cdot tlrp} = \pm k_{tsr} \theta_{tlrp}$
tendon reflex	$y_i < 0$	$Feed_{e\cdot tr} = k_{tr}(\bar{\theta} + 1)$, if $\bar{\theta} \geq -1 \text{ rad/s}$
flexor reflex	$y_i > 0$	$Feed_{f\cdot fr} = (k_{fr}/0.12)(0.12 - t)$, if $f_x > 1.5 \text{ Kgf}$ where $t=0$ (the instance a leg stumbles) $\sim 0.12 \text{ s}$
extensor reflex	$y_i < 0$	$Feed_{e\cdot er} = k_{er} \theta_{tlrp}$, if $f_x > 1.5 \text{ Kgf}$ and $\theta_{tlrp} \geq 0$

In addition, during such an extension of the current phase of the CPG, the CPGs of neighboring legs are expected to maintain their phase differences from the CPG by extending their own current phases. When we employ the CPG network in which the neurons of the CPGs of neighboring legs are mutually inhibited (Fig. 2-(b)), the phase differences between CPGs can be maintained autonomously, since the excitation of the neuron of the CPG inhibits the neurons of other CPGs. This means that condition (d), described in Section 2.4, can be satisfied on the CPG network.

3.3 Experiments

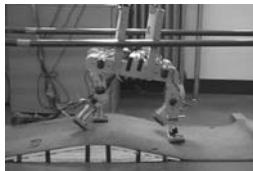
We tried to realize dynamic walking on terrains of medium degrees of irregularity, where a slope, an obstacle and undulations continue in series, with the following feedback to CPGs.

$$\begin{aligned} Feed_e &= Feed_{e\cdot tsr\cdot tlrp} + Feed_{e\cdot tr} + Feed_{e\cdot er} \\ Feed_f &= Feed_{f\cdot tsr\cdot tlrp} + Feed_{f\cdot fr} \end{aligned} \quad (5)$$

By realization of such adaptive walking using the control method expressed by (1) and (5) with fixed values for all parameters, we were able to confirm that the control method proposed in this section enables the robot to adapt to unknown irregular terrain. Photos of the robot walking on such irregular terrain and the experimental results are shown in Fig. 4 and in Fig. 5.

In Fig. 5, stable dynamic walking was realized, while the flexor reflex was activated when the robot stumbled on bumps: (A,B), and the tendon reflex was activated in landing on bumps: (C,D). When the right foreleg stumbled on bumps: (A,B), the flexor reflex instantly increased the neuron torque of the right foreleg. This flexor reflex torque made the active period of the flexor neuron of the right foreleg much longer as an effect of reflexes via CPGs. Correspondingly, the active period of the extensor neuron of the left foreleg was extended: (E,F) as an effect of the CPG network. These results confirmed the ability of a neural system model, consisting of CPGs and reflexes, to autonomously adjust the active periods of the neurons and the phase differences between the CPGs, as described in Section 3.2.

The tendon reflex at Fig. 5-(D) caused additional torque against gravity on the right foreleg and extended the stance phase of the leg. This was similar to what



(a) up and down a slope of 12 degrees.



(b) on terrain undulations 3 cm in height.

Fig. 4. Photos of walking

Pearson (Hiebert, Pearson, 1999) found in cats, and it helped satisfy conditions (c) and (d), as described in Section 2.4.

4 Three-dimensional Walking

In this section, we consider the relation among the CPGs, the pitching motion of the legs, and the rolling motion of the body. A naturally generated rolling motion, gaits, adaptations for 2D irregular terrains (Fig. 6), changing direction, and so on are some of the new issues in 3D dynamic walking. The first three issues listed, related to the rolling motion feedback to CPGs, are the issues we address in this section.

4.1 Mechanical design

We designed Tekken (Fig. 7-(a)) to solve the mechanical problems which occurred in Patrush (Kimura, Fukuoka et al., 1996). The length of the body and a leg in standing are 23 cm and 20 cm. The weight of the whole robot, a whole leg and a lower link under a knee are 3.1 Kg, 0.5 Kg and 0.06 Kg, respectively. Each leg has a hip pitch joint, a hip yaw joint, a knee pitch joint, and an ankle pitch joint. The ankle joint is passive with lock and spring mechanisms (Fig. 7-(b)). Rate gyro sensors for three axes and two inclinometers for pitch and roll axes are mounted on the body in order to measure the body pitch/roll/yaw angles. A contact sensor is attached to the toe of each foot. The direction in which Tekken moves while walking can be changed by using the hip yaw joints.

4.2 Pitching motion control

In order to generate an arbitrary gait from a trot to a pace via a walk with a single network configuration of CPGs in Tekken, we newly propose an asymmetric CPG

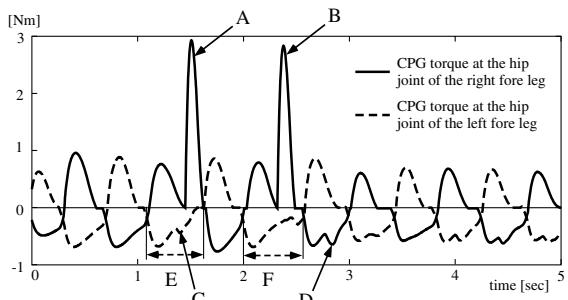


Fig. 5. CPG torque in walking over terrain undulations (Fig. 4-(b))

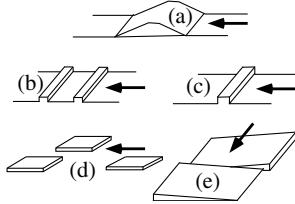
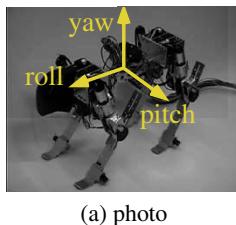
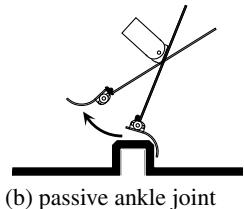


Fig. 6. One-dimensional: (a), (b), (c) and two-dimensional: (d), (e) irregular terrains.



(a) photo



(b) passive ankle joint

Fig. 7. A quadruped robot: Tekken

network shown in Fig. 2-(c). In Fig. 2-(c), a CPG of a foreleg is inhibited by a CPG of a hindleg with a connecting weight: w_{hf} , and a CPG of a hindleg is not inhibited by a CPG of a foreleg: $w_{fh}(=0)$.

The control diagram of the pitch joints is shown in Fig. 8. In Patrush, the output of a CPG was joint torque: (4), and both the amplitude and phase of the CPG output were used for joint control. In Tekken, in order to separate phase information and joint torque via CPGs, only the phase of the CPG output: y_i in (2) was used. Joint torque was determined by the following PD controller, corresponding to a stretch reflex at an α motor neuron in animals (Fig. 8).

$$trq_{jnt} = -K1_{jnt}(\theta_{jnt} - \theta_{jnt}^*) - K2_{jnt}\dot{\theta}_{jnt} \quad (6)$$

where jnt is *hip* or *knee*, θ_{jnt} is the joint angle, θ_{jnt}^* is the desired angle, $K1_{jnt}$ and $K2_{jnt}$ are the feedback gains, and trq_{jnt} is joint torque, respectively. The desired angle of each joint is switched based on the phase of the CPG output: y_i (Fig. 8), where θ_{jnt}^{stance} and θ_{jnt}^{swing} are constant in all experiments.

$$\theta_{jnt}^* = \begin{cases} \theta_{jnt}^{stance}, & \text{if } y_i < 0; \\ \theta_{jnt}^{swing}, & \text{otherwise} \end{cases} \quad (7)$$

As a result of the switching of the joint desired angle in (7) and the feedback signal to the CPG in (3), the CPG and the hip joint motion were mutually entrained.

By using the pitch joint controller described above, and the CPG network shown in Fig. 2-(c) with a constant connecting weight: $w_{rl,lf} = -2$ and various connecting weights: w_{hf} from -1 to 0.1 , we were able to realize dynamic walking on a flat terrain in various gaits, from a trot to a pace via a walk.

4.3 Rolling motion feedback to CPGs

Since Tekken has no joint around the roll axis, CPGs based joint control is applied only to the pitching motion. However, since a dynamic system similar to an inverted pendulum appears in the two-legged stance phase, a rolling motion is naturally generated in most of the gaits as a result. The amplitude of the rolling motion generated

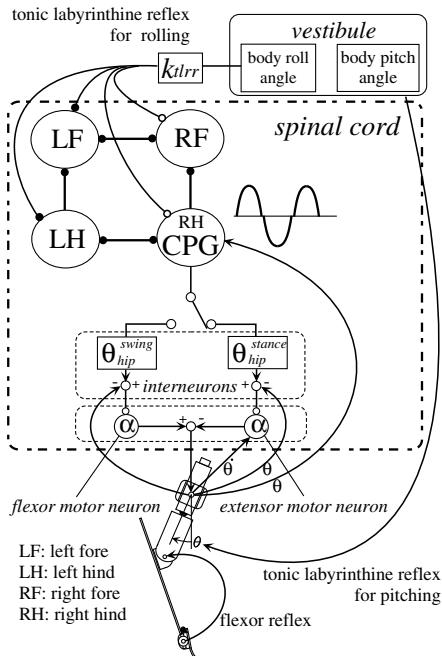


Fig. 8. Control diagram for Tekken.

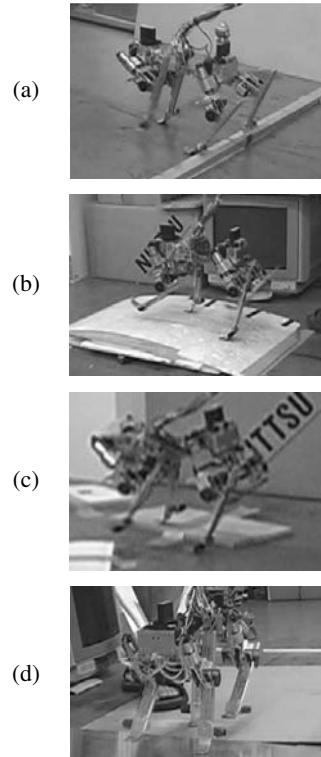


Fig. 9. Walking over irregular terrain shown in Fig. 6.

in walking on flat terrain is mainly determined by the gait, duty factor, and the period of the pitching motion cycle.

In Tekken, we made the body angle around the roll axis be inputted to the CPGs as a feedback signal expressed by (8) (Fig. 8).

$$\begin{aligned} Feed_{e.tlrr} &= \delta(\text{leg}) k_{tlrr} \times (\text{body roll angle}) \\ Feed_{f.tlrr} &= -Feed_{e.tlrr} \end{aligned} \quad (8)$$

$$\delta(\text{leg}) = \begin{cases} 1, & \text{if leg is a right leg;} \\ -1, & \text{otherwise} \end{cases}$$

CPGs, the pitching motion of the legs and the rolling motion of the body, are mutually entrained through the rolling motion feedback to CPGs expressed by (8). This means that the rolling motion can be the standard oscillation for whole oscillations, in order to compensate for the weak connection between the fore and hind legs in the CPG network (Fig. 2-(c)). As a result, the phase difference between the fore and hind legs is fixed, and the gait becomes stable.

4.4 Tonic labyrinthine reflex for rolling

When a leg lands on a bump while walking on irregular terrain, the disturbance of the rolling motion to the pitching motion becomes larger. Therefore, the periods of the current phases of the CPGs in the pitching motion should be adjusted according to the rolling motion, in order to satisfy the condition (b) and (d) described in Section 2.4.

The rolling motion feedback to CPGs: (8) employed in Section 4.3, contributes to an appropriate adjustment of the periods of the stance and swinging phases while walking on irregular terrain (Fig. 10), as a tonic labyrinthine reflex for rolling described in Section 2.4.

In Fig. 10, the right foreleg lands on a bump in a trot gait, and the body is inclined in a roll plane. E or F means the extensor or flexor neuron of a CPG, respectively. '+' or '-' means the inner state of the neuron is increased or decreased by $Feed_{e-tlrr}$, respectively. Extending the stance phase of the left hindleg (E+), and shortening the stance phase of the right foreleg (E-) and the swinging phase of the left foreleg (F-), prevent the body from the excess inclination in a roll plane. Extending the swinging phase of the right hindleg (F+) enables the reliable landing of the leg on the ground.

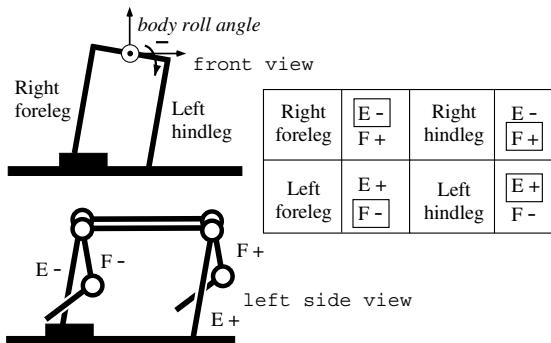


Fig. 10. A tonic labyrinthine reflex for rolling.

4.5 Adaptive walking on irregular terrain

We made Tekken walk on several irregular terrains (Fig. 6) with the following feedback to CPGs. In all experiments, values for all parameters of CPGs, reflexes and joint PD controller were fixed.

$$\begin{aligned} Feed_e &= Feed_{e-tsr-tlp} + Feed_{e-tlrr} \\ Feed_f &= Feed_{f-tsr-tlp} + Feed_{f-tlrr} \end{aligned} \quad (9)$$

where the tonic labyrinthine reflex for pitching: $Feed_{\{e,f\}-tsr-tlp}$ in Table. 2 was also employed (Fig. 8), in order to satisfy condition (c) described in Section 2.4.

When a toe of a swinging leg makes contact with an obstacle, the passive mechanism at the ankle joint (Fig. 7-(b)) quickly prevents the swinging leg from stumbling, and the flexor reflex (Fig. 8) is activated afterward. By using the flexor reflex and the tonic labyrinthine reflex for pitching, Tekken walked over an obstacle 2 cm in height while stumbling (Fig. 9-(a)), and also walked up and down a slope of 10 degrees in the forward direction (Fig. 9-(b)).

Tekken also succeeded in walking over terrains consisting of several boards (Fig. 6-(d)) or slopes in a side direction (Fig. 6-(e)), in a stable trot gait with an appropriate adjustment of periods of the stance and swinging phases by a tonic labyrinthine reflex for rolling (Fig. 9-(c), (d)). Without a tonic labyrinthine reflex for rolling, the gait was greatly disturbed, even if it didn't cause Tekken to fall.

5 Conclusion

In the neural system model proposed in this study, the relationships among CPGs, reflexes and the mechanical system are simply defined, and motion generation and adaptation are emergently induced by the coupled dynamics of a neural system and a mechanical system by interacting with the environment. The coupled dynamic system can induce autonomous adaptation according to its own dynamics, under changes in the environment. Therefore, we can avoid such serious problems in robotics as modeling of mechanical system and environment, autonomous planning, conflict between planned motion and actual motion and so on.

To generate appropriate adaptation, it is necessary to design and construct neural systems carefully. In this study, we designed a neural system consisting of CPGs and reflexes referring to knowledge and concepts in physics, biology, physiology and so on. Concerning a coupling of a neural system and a mechanical system, the relation between parameters of CPGs and the dynamics of a mechanical system was analyzed to some extent. Analysis of the relation between parameters of reflexes and the mechanical system remains as a future problem to be solved.

References

- Cham JG, Bailey SA, Cutkosky MR (2000) Robust dynamic locomotion through feedforward-preflex interaction. In Proc. of ASME IMECE.
- Full RJ, Koditschek DE (1990) Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *J. of Experimental Biology*, 202:3325–3332.
- Grillner S (1981) Control of locomotion in bipeds, tetrapods and fish. *American Physiol. Society*, Bethesda, MD.
- Hiebert G, Pearson K (1999) The contribution of sensory feedback to the generation of extensor activity in the decerebrate cat. *J. Neurophysiology*, 81:758–770.
- Ijspeert AJ (2001) A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biolog. Cybern.*, 84:331–348.
- Ilg W, Albiez J, Jedele H, Berns K, Dillmann R (1999) Adaptive periodic movement control for the four legged walking machine BISAM. In Proc. of ICRA99, pp 2354–2359.

- Kimura H et al. (1999a) Realization of dynamic walking and running of the quadruped using neural oscillator. *Autonomous Robots*, 7(3):247–258.
- Kimura H, Fukuoka Y et al. (1996) 3D adaptive dynamic walking of a quadruped robot by using neural system model. In Proc. of CLAWAR2001, pp 97–104.
- Kimura H, Fukuoka Y et al. (1999b) Biologically inspired adaptive dynamic walking of the quadruped on irregular terrain. *Robotics Research*, 9:329–336.
- Kimura H, Fukuoka Y et al. (2001) Adaptive dynamic walking of a quadruped robot using neural system model. *Advanced Robotics*, 15(8):859–876.
- Kimura H, Shimoyama I, Miura H (1990) Dynamics in the dynamic walk of a quadruped robot. *Advanced Robotics*, 4(3):283–301.
- Lewis MA, Etienne-Cummings R, Cohen AH, Hartmann M (2000) Toward biomimetic control using a custom aVLSI CPG chips. In Proc. of ICRA2000, pp 1289–1295.
- Matsuoka K (1987) Mechanisms of frequency and pattern control in the neural rhythm generators. *Biolog. Cybern.*, 56:345–353.
- Matsuoka K (1999) Sensorimotor interactions during locomotion: principles derived from biological systems. *Autonomous Robots*, 7(3):239–245.
- Saranli U, Buehler M, Koditschek DE (2000) Design, modeling and preliminary control of a compliant hexapod robot. In Proc. of ICRA2000, pp 2589–2596.
- Taga G (1995) A model of the neuro-musculo-skeletal system for human locomotion II. – real-time adaptability under various constraints. *Biolog. Cybern.*, 73:113–121.
- Taga G, Yamaguchi Y, Shimizu H (1991) Self-organized control of bipedal locomotion by neural oscillators. *Biolog. Cybern.*, 65:147–159.
- Tsujita K, Tsuchiya K et al. (2000) Decentralized autonomous control of a quadruped locomotion robot. In Proc. of Adaptive Motion of Animals and Machines (AMAM), pp WeA–I–2.
- Yamaguchi J, Kinoshita N, Takanishi A, Kato I (1996) Development of a dynamic biped walking system for humanoid. In Proc. of ICRA1996.

Design and Applications of Parallel Robots

Karol Miller

Department of Mechanical and Materials Engineering, The University of Western Australia,
Crawley/Perth WA 6009, Australia

Abstract. An optimal kinematic design method suited for parallel manipulators is described. The optimal configuration for a Delta-type three-degree-of-freedom spatial, translational manipulator, known as the New University of Western Australia Robot (NUWAR) is presented, and shown to be advantageous over the Delta configuration in terms of workspace volume. These results led to the construction of a prototype and an Australian Patent application.

The kinematic optimisation process yielding a design, which delivers the best compromise between manipulability and a new performance index: space utilisation, is presented. The process leading to finding an optimal configuration of Linear Delta robot is described.

An example medical application of a parallel robot is discussed. The robot is designed to work inside an open magnetic resonance scanner. A concept of a robot control system, based on biomechanical models of organs the robot operates on, is presented.

1 Introduction

The literature contains much information regarding various types of parallel robots, e.g. (Merlet, 1990; Tsai, 1999).

Most important here, however, is the development of the DELTA by Clavel at the EPFL in Lausanne (Clavel, 1991; Miller, Clavel, 1992). The DELTA is a three-degree-of-freedom translational, spatial manipulator. Parallel manipulators possess a number of advantages when compared to traditional serial arms. They offer generally much higher rigidity and smaller mobile mass than their serial counterparts. These features allow much faster and more precise manipulations. The main drawback of parallel manipulators is their small workspace and also often limited manipulability in certain regions of the workspace.

This contribution summarises work on optimal design and applications of spatial parallel manipulators, conducted in the Department of Mechanical and Materials Engineering of The University of Western Australia.

2 Optimal Design of Parallel Robots

2.1 NUWAR – DELTA Type Robot with Maximum Workspace Volume

Like the Delta, the robots considered here consist of three kinematic chains in parallel, which connect the base to the end-effector. It is of particular interest to determine, for given dimensions of a manipulator, how the size and shape of the robot workspace varies with values of two angles defining the orientations of the motor axes (Fig. 1).

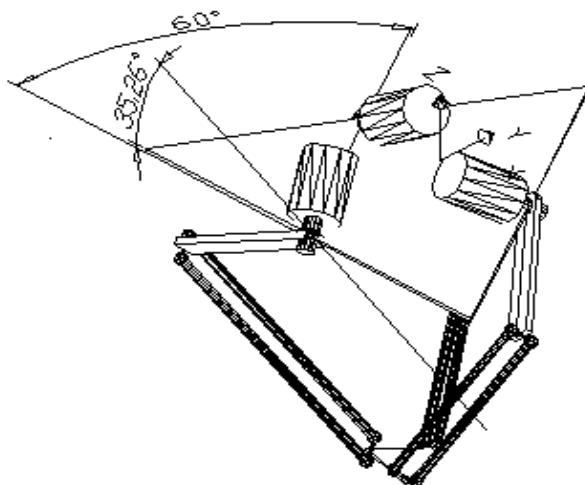


Fig. 1. Layout Layout of the New University of Western Australia Robot. Design variables are angles α ($= 0^\circ$ for DELTA and $= 35.26^\circ$ for NUWAR), and β ($= 0^\circ$ for DELTA and $= 60^\circ$ for NUWAR).

Computer models of the manipulators were written parametrically so that various configurations could easily be analysed. The mechanical simulation package ADAMS (ADAMS, 1993) and AutoCad/AutoLisp (AutoCad, 1997) were used. In addition to the dimensions of each component, two design variables – angles α and β – were introduced: α measures the inclination of each motor to the horizontal (for the Delta robot $\alpha = 0^\circ$), and β – rotation of each motor with respect to the vertical axis (for the Delta robot $\beta = 0^\circ$). As the arm is rigidly attached to its motor axis, the angles α and β control the orientation of the three arm/forearm assemblies. Varying α and β angles leaves the locations of the three motors unchanged.

The choice of α and β resulting in maximum workspace volume was decided by exhaustive search: the volume was calculated for $91 \times 91 = 8281$ combinations of design variable values from the interval $(0^\circ, 90^\circ)$, Table 1. The calculations showed that the Delta configuration is not optimal in terms of the workspace volume. Other feasible configurations occur in the range $\alpha > 30^\circ$ and $\beta > 50^\circ$. The optimal motor axis orientation was given by $\alpha = 35.26^\circ$ and $\beta = 60^\circ$. It is worth noting that for such choice of the design variables the motor axes are orthogonal.

This configuration – known as the New University of Western Australia Robot – NUWAR, Fig 2 – has the workspace 9.4% larger than the workspace of the Delta, assuming all dimensions being the same.

Results of computer simulations described in this Section lead to the construction of the prototype, Fig. 2. Capable of achieving end-effector accelerations of 600 m/s^2 NUWAR is one of the fastest robots in the world.

Table 1. Volume of workspace, with respect to volume of workspace of the Delta robot, $\alpha = 0^\circ$ and $\beta = 0^\circ$

		β									
		0	10	20	30	40	50	60	70	80	90
α	0	1.000	0.965	0.902	0.819	0.733	0.653	0.580	0.525	0.495	0.484
	10	0.930	0.992	1.003	0.969	0.913	0.852	0.782	0.732	0.698	0.688
	20	0.817	0.915	0.995	1.038	1.030	1.000	0.965	0.927	0.898	0.883
	30	0.724	0.849	0.943	1.022	1.071	1.080	1.061	1.056	1.044	1.036
	35.26	0.699	0.830	0.926	1.011	1.058	1.091	1.094	1.106	1.087	1.090
	40	0.690	0.816	0.920	0.999	1.054	1.087	1.105	1.113	1.115	1.117
	50	0.726	0.819	0.910	0.981	1.040	1.077	1.096	1.110	1.111	1.117
	60	0.800	0.869	0.936	0.994	1.049	1.082	1.097	1.114	1.105	1.118
	70	0.905	0.950	0.999	1.037	1.069	1.080	1.117	1.120	1.127	1.137
	80	1.004	1.030	1.051	1.074	1.090	1.113	1.123	1.125	1.130	1.136
	90	1.079	1.079	1.079	1.079	1.079	1.079	1.079	1.079	1.079	1.079

2.2 Kinematic Optimisation of Linear Delta

When designing parallel manipulators one often has to reach a compromise between two conflicting design goals: manipulability and workspace size. Maximisation of the workspace volume alone tends to produce manipulators being singular in all configurations, whilst considering manipulability in isolation may lead to architectures with relatively small workspaces; a clear example of this phenomenon may be found in the results of Stamper, Tsai, Walsh (1997). Accordingly, the objective function considered in this section is a weighted sum of two performance indices.

The first index is based on that of (Gosselin, Angeles, 1989), and measures the inverse of the Jacobian's condition number. In order to overcome the limitations associated with the use of this index in isolation, a new performance index is proposed which measures *space utilisation*, and reflects the ratio of the workspace size to the physical size of the robot's structure.

The method is applied to the Linear Delta Robot, Fig. 3, Fig. 4 and Fig. 5.

Design Variables In order to reduce the number of variables and render the results independent of the scale of each design candidate, several non-dimensional ratios have been chosen as the design variables. These variables, shown in Table 2., have been selected to reflect intuitive measures of the relative proportions of the Linear Delta robot.



Fig. 2. NUWAR Prototype

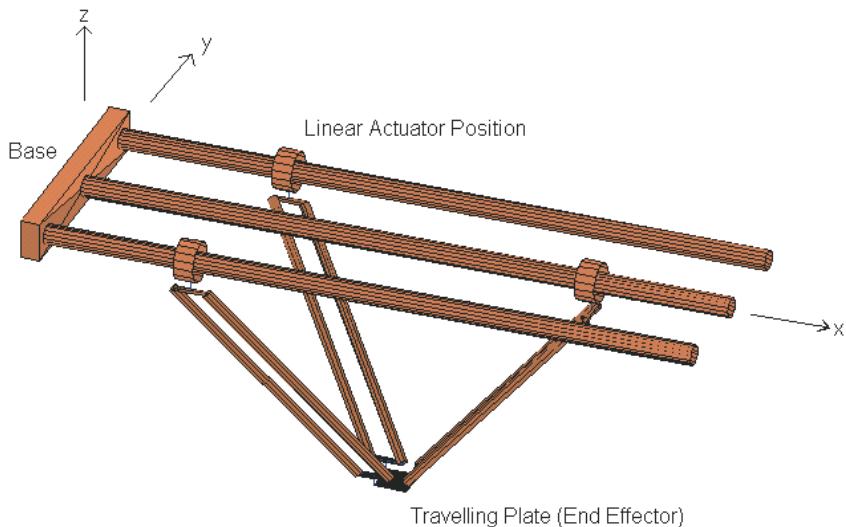


Fig. 3. Render MATLAB representation of the Linear Delta

Performance Indices *Manipulability*: The first performance index, η_1 , measures the average value of the inverse of the condition number of the Jacobian matrix $\frac{1}{K_J}$

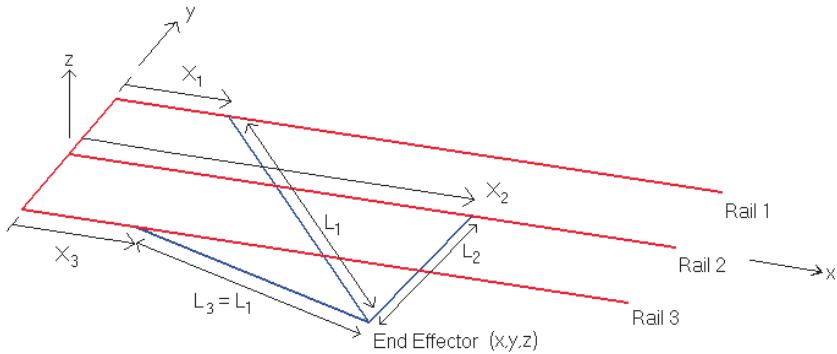


Fig. 4. Wireframe representation of Linear Delta showing nomenclature

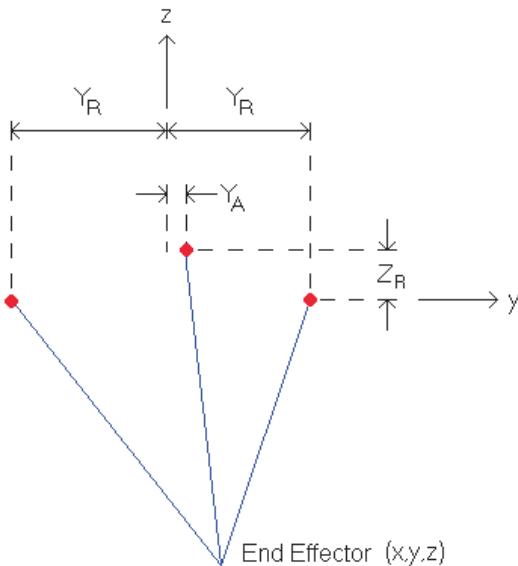


Fig. 5. End view (y-z plane) of wireframe representation showing nomenclature

over the workspace cross section, normalised by the cross sectional area, A_W :

$$\eta_1 = \frac{\int_{A_W} \frac{1}{\mathcal{K}_J} dA_W}{\int_{A_W} dA_W} = \frac{\int_{A_W} \frac{1}{\mathcal{K}_J} dA_W}{A_W} \quad (1)$$

Adapted from (Gosselin, Angeles, 1989), this index possesses several favourable characteristics. The index is normalised by the workspace size, and therefore gives a measure of kinematic performance independent of the differing workspace sizes of design candidates. Furthermore, the reciprocal of \mathcal{K}_J is bounded between 0 and 1, and is more convenient to handle than \mathcal{K}_J , which tends to infinity at singularities;

Table 2. Kinematic design variables

Variable	Description	Restrictions
$\frac{L_1}{Y_R}$	Ratio of outer arm length to actuator separation.	≥ 1
$\frac{Y_A}{Y_R}$	Ratio of centre actuator horizontal eccentricity to actuator separation.	$0 - 1$
$\frac{Z_R}{Y_R}$	Ratio of centre actuator vertical eccentricity to actuator separation.	
$\frac{L_2}{L_1}$	Ratio of centre arm length to outer arm length.	≥ 0

hence, during numerical integration, the number of sample points near singularities has a reduced effect on the result, since $\frac{1}{K_J}$ approaches zero (rather than infinity) at these points. Lastly, the (dimensionless) value of the index lies between zero, for a manipulator singular in all configurations, and unity, for a manipulator perfectly kinematically isotropic in all configurations.

Space Utilisation: This performance index was developed in order to overcome the problems involved in applying simple performance indices, in isolation, to the Linear Delta. For example, attempts to optimise the structure by maximising η_1 alone led to an architecture with a large structural space requirement but a vanishing workspace size, due to the tending of $\frac{L_2}{L_1}$ towards zero; whilst the maximisation of workspace cross sectional area caused the actuators to be collinear (i.e. $Y_R = 0$, since area increases monotonically with decreasing Y_R) resulting in a manipulator which is singular in all configurations. It became clear that a practical optimisation of the Linear Delta would require a utility function comprising multiple performance indices.

The space utilisation performance index is defined as:

$$\eta_2 = \frac{\text{Workspace Cross Sectional Area}}{\text{Bounding Box Area}} = \frac{A_W}{\text{Bounding Box Area}} \quad (2)$$

where the bounding box is defined as the smallest rectangle in the $y - z$ plane, whose sides are parallel to the y and z axes, containing all three actuators and every point within the workspace cross section. The space utilisation value reflects the ratio of workspace size to the physical size of the robot's structure. The index is dimensionless, bounded by the range [0 1], its calculation is simple and inexpensive (c.f. numerical integration of the Jacobian's condition number), and its value is independent of the overall scale of each design that it is applied to. Most importantly, designs requiring large volumes of space, but yielding small workspaces, are penalised. The utility function for maximisation is therefore defined as:

$$\eta = \omega_1 \left(\frac{\int_{A_W} \frac{1}{K_J} dA_W}{A_W} \right) + \omega_2 \left(\frac{A_W}{\text{Bounding Box Area}} \right) = \omega_1 \eta_1 + \omega_2 \eta_2 \quad (3)$$

The function is bounded by the range $[0 \ (w_1 + w_2)]$, and is independent of both the overall scale of the design candidate to which it is applied, and the limits chosen for the design variables (since neither performance index is normalised by its maximum observed value).

Computational Issues: An exhaustive (brute force) search method has been utilised to solve the optimisation problem. Whilst computationally expensive, an exhaustive search is simple, reduces the probability of any local maxima being overlooked, and provides information regarding the behaviour of the utility function over the entire range of allowable values for each of the design variables. The limits selected for the design variables are as follows: $1.4 \leq \frac{L_1}{Y_R} \leq 2.7$; $0 \leq \frac{Y_A}{Y_R} \leq 0.7$; $-0.5 \leq \frac{Z_R}{Y_R} \leq 0.5$; $0.6 \leq \frac{L_2}{L_1} \leq 1.8$. Several of these limits are necessary to ensure that the resulting structure is practical. The remaining limits were chosen heuristically, with alterations made on a trial-and-error basis to yield a reasonably large range, and to ensure the inclusion of any local maxima of the utility function. A notable exception is the case where the manipulability index, η_1 , was considered in isolation (i.e. $w_2 = 0$); in this situation $\frac{L_2}{L_1}$ was allowed to approach zero.

Since there exists no analytical means of calculating the Jacobian's condition number (k_J), and therefore no closed-form expression for the manipulability index, numerical integration is required to determine the value of the utility function. The integral of (1) may be approximated by a discrete sum:

$$\eta_1 = \frac{\int_{A_W} \frac{1}{K_J} dA_W}{A_W} \approx \frac{\sum_{a \in A_W} \frac{1}{K_J}}{N_a} \quad (4)$$

where each a is one of N_a integration points in the $y - z$ plane workspace cross section. These points are generated by examining the boundaries of the cross section and forming a uniformly distributed grid inside it.

In the majority of cases, closed form methods may be utilised to calculate the exact cross sectional area of the workspace (A_W). In some instances, however, the choice of design variables precludes the application of the closed form solutions, and it is necessary to approximate A_W . The following approximation has been utilised:

$$A_W \approx dA_W \cdot N_a \quad (5)$$

where dA_W is the small area associated with each of the N_a integration points. The accuracy of the approximation in (5) increases as N_a increases. Application of this equation to several situations, in which a closed form solution for A_W existed revealed that if dA_W was sufficiently small such that $N_a > 5000$, then the relative error in A_W for these cases was generally less than 0.1%.

The complete four-dimensional optimisation, programmed in MATLAB 5.3 (MATLAB 1999) required about three hours of computations using Pentium II 200 MHz PC.

Optimisation Results Since the weights of the utility function, w_1 and w_2 in (3), are necessarily subjective, results for several combinations of values are presented in Table 2. Visualisation via several surface plots greatly aids the interpretation of these results. In Fig. 6, the values of (a) $\eta = \eta_1 + \eta_2$, (b) η_1 , and (c) η_2 are shown. In order to present the four-dimensional data set, two planes are displayed – each of which passes through the optimum point (emboldened in Table 2).

The tendency for the solution to converge on a zero workspace size architecture when manipulability was optimised alone was the reason for developing the second performance index, space utilisation. Other behaviours of the manipulability index that were noted include the confirmation that optimal value of Y_R is zero (a symmetrical architecture), and that for an arbitrary cross section through the four dimensional data set, manipulability generally exhibits relatively little variation when compared to space utilisation. The peaks and troughs shown in the space utilisation surfaces may be explained intuitively by considering the effects of changes in the design variables to the workspace cross section. For example, the peak in space utilisation at $\frac{L_1}{Y_R} = 2$ corresponds to architectures where the lengths of the outer arms are equal to the separation of the outer actuators – the width of the bounding box is then simultaneously determined by both the workspace and the structure.

Reducing the ratio of outer arm length to actuator separation dramatically reduces η_2 , since the workspace size falls without an accompanying reduction in the width of the bounding box. Increasing the ratio leads to modest reductions in space utilisation, as the increasing size of the bounding box is partially offset by the expanding workspace area.

Table 3. Optimisation results

Weights	Description	$\frac{L_1}{Y_R}, \frac{Y_A}{Y_R}, \frac{Z_R}{Y_R}, \frac{L_2}{L_1}$
$w_1 = 1, w_2 = 0$	Manipulability alone.	(1.55, 0, -0.75, →0)
$w_1 = 1, w_2 = 0$	Manipulability alone, with $Z_R = 0$.	(1.48, 0, 0, 0.56)
$w_1 = 0, w_2 = 1$	Space utilisation alone.	(2.00, 0, 0, 0.60)
$w_1 = 1, w_2 = 1$	Equally weighted utility function.	(2.00, 0, 0, 0.63)

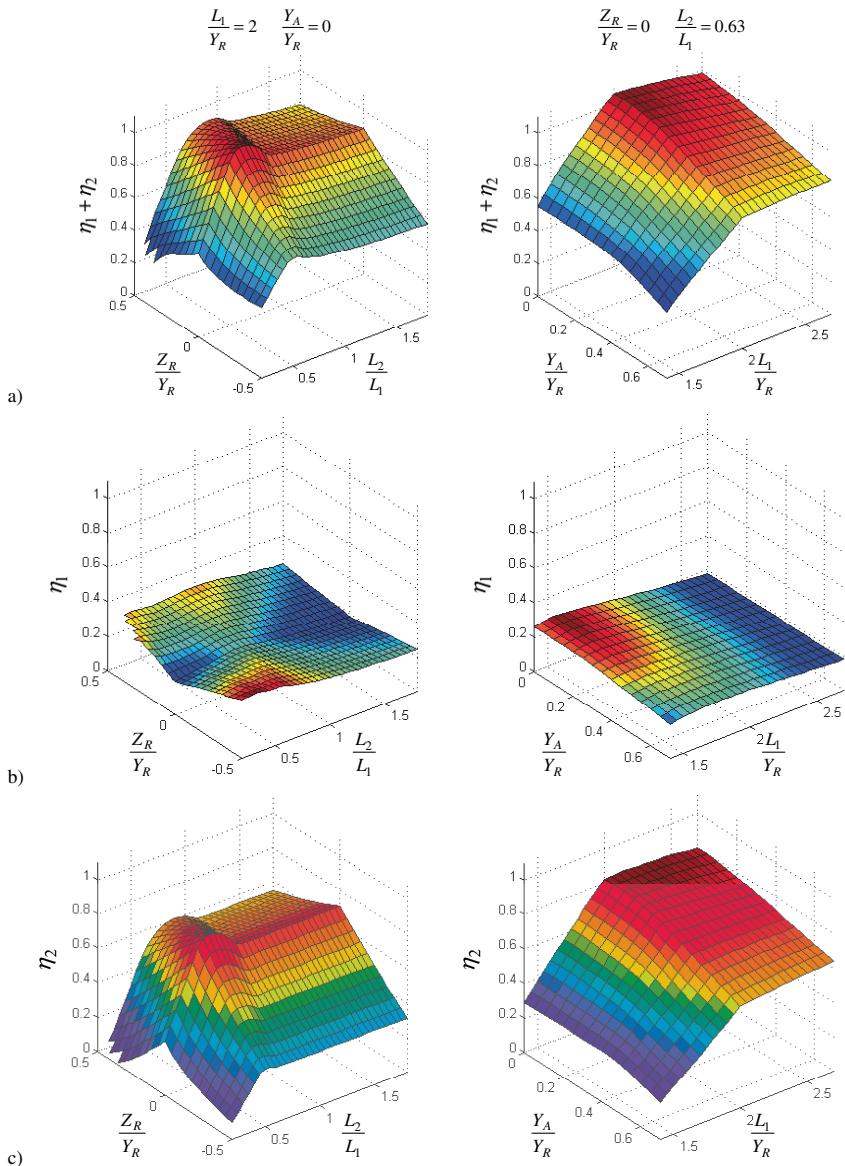


Fig. 6. Optimisation results

The optimum architecture is relatively insensitive to increasing w_1 – large increases in the weighting of manipulability ($w_1 > 3$) are necessary before a considerable change in geometry is noted. Whilst small increases in w_2 affect the results of

the optimisation, the change is limited to the variable $\frac{L_2}{L_1}$, with the greatest possible variation being a decrease of just 0.03.

Currently, a Linear Delta prototype intended for cutting surfboards is under construction.

3 Medical Applications – MRI Compatible Parallel Robot

The advantages of surgical robots and manipulators are well recognised in the clinical and technical community. Precision, accuracy, and the potential for telesurgery are the prime motivators for applying advanced robot technology to surgery. Surgical robots require trajectory planning, which in practice, relies upon preoperative images. If the target organ is deformable the trajectory needs to be updated during the procedure according to the magnitude of the deformation. Here, image-guided surgery, supplemented by a computer simulation of the deformation, is a natural solution.

Magnetic resonance imaging (MRI) provides excellent soft tissue discrimination and a well-defined 3D coordinate reference system. An intra-operative MR scanner shown in Fig. 7 (a) (Signa $\frac{SP}{i}$, GE Medical Systems, Milwaukee, WI, 0.5 Tesla) was specifically designed to bring the power of MRI to the operating theatre. This unit has a pair of parallel facing donut-shaped magnets, separated by an air gap of 560 mm. Two surgeons can stand in the gap to access the patient.

For reliable and safe operation of an MRI compatible robot it is required that all actuators be located outside the gap between magnets and, therefore, outside usable workspace. This requirement cannot be satisfied by serial robots. A parallel kinematic architecture is a preferable solution. Figure 7 demonstrates a unique configuration of a novel magnetic resonance (MR) compatible robotic system for use in MR guided surgery.

The machine was installed in Surgical Planning Laboratory of The Harvard Medical School. The goal of this robot-assist system was to enhance the surgeon's operative performance by providing accurate mechanics and numerical control, rather than undertaking the procedure without human control or intervention. Therefore, the system had to co-exist and co-operate with the surgeon, without its size and physical appearance being perceived as threatening in any way. The system shown in Fig. 7 actively navigates a small tool, such as a catheter needle, with "pin-point" accuracy under intra-operative MR guidance. Intra-operative MR images serve to revise the trajectory as required. Currently the robot can perform positioning of medical instruments and the insertion of needles.

The sketch of the MR compatible robot control system, including a computer simulation of organ deformation module, is shown in Fig. 8 (Chinzei, Miller, 2001). For confident control of a surgical robot the properties of the relevant tissues must be known. It would be helpful to be able to predict interaction forces and tissue displacements. Thus the role and importance of mathematical and computational models of very soft tissue deformation can be appreciated. Although NMRI can provide rich information of tissue deformation tens of seconds are currently required

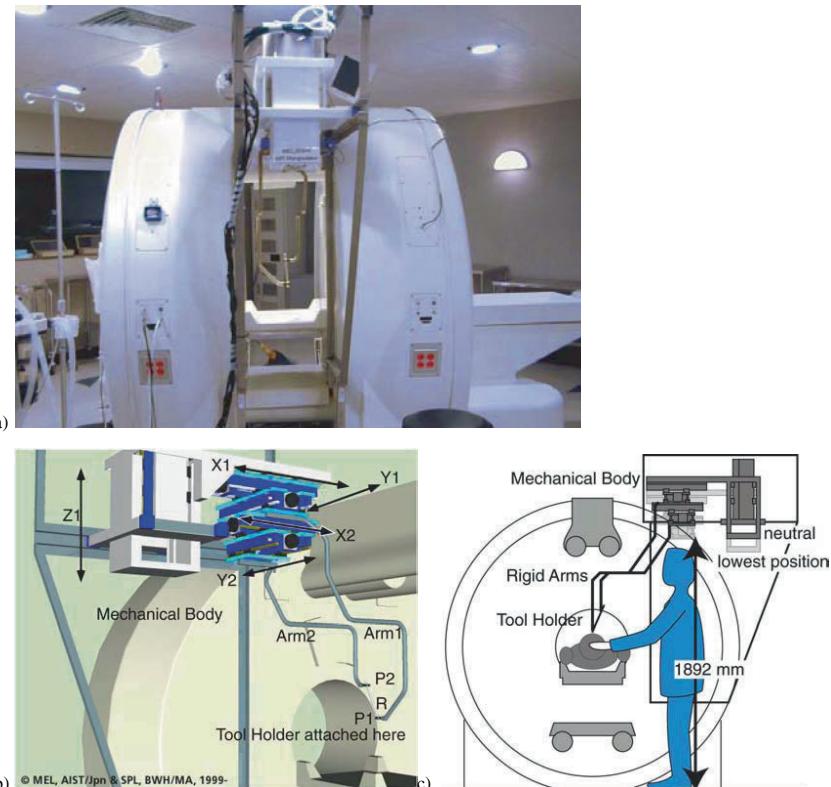


Fig. 7. Magnetic Resonance Compatible robot constructed in Mechanical Engineering Laboratory, Tsukuba, Japan (Chinzei, Miller, 2001), attached to the intra-operative MR scanner (a), the layout of moving parts (b) and the profile of the workspace, (c) The moving part does not obstruct the workspace of the surgeon.

to produce and analyse a new set of images. The practical method of dealing with these delays is for a computer to predict the deformation based on a mathematical model (Miller, Chinzei, Orssengo, Bednarz, 2000).

During operations soft organs deform significantly. Although intra-operative MRI can theoretically track this deformation, the time to produce and analyse the resulting 3D images is at least several seconds, if not tens of seconds. Consequently, it is proposed that the robot control system incorporates a computational model that would pre-calculate the extent of tissue deformation, see Fig. 8. In order to design a robust control system for a surgical robot, the properties of organs on which it operates must be known, and the deformation of the organ caused by the robot movement must be predictable. This requires an understanding of the biomechanical behaviour of soft organs.

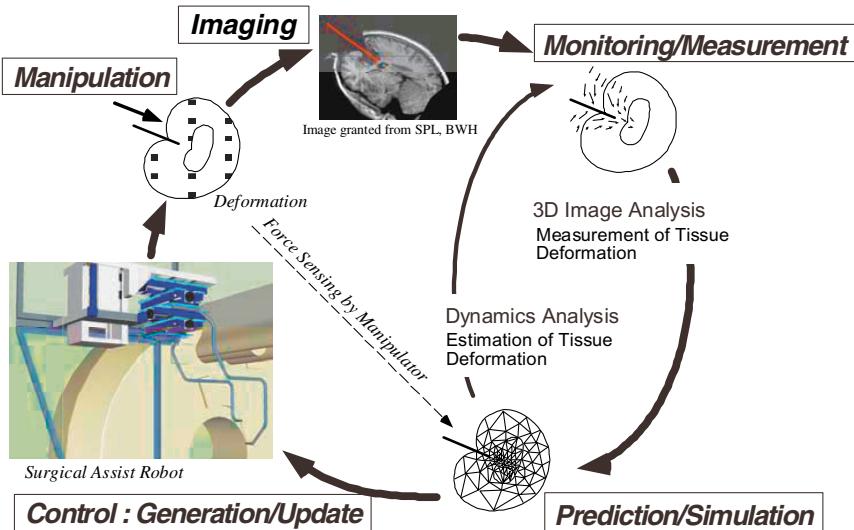


Fig. 8. The concept of computer-based prediction part in the MRI guided robot control system

4 Conclusions and Discussion

Multidimensional optimisation methods developed in this paper were applied to the design of two spatial parallel manipulators. A utility function, incorporating two performance indices, was formulated in order to determine those architectures, which yield an optimum compromise between manipulability and a new performance index, space utilisation. Space utilisation is a concept designed to overcome the problems of small, non-singular workspaces often encountered when a parallel manipulator is optimised with only manipulability in mind.

The exhaustive search algorithm was used in both cases to reliably find all prospective design candidates. The method was shown to work extremely well and, therefore, is a viable competitor to advanced non-linear programming methods.

The numerical results show that NUWAR had maximum workspace volume among Delta-type manipulators, and that the Linear Delta developed gave the optimal compromise between the workspace size and manipulability.

The results presented here, and the suite of MATLAB programs created to perform the optimisation, form a framework, which may be utilised to create and analyse designs that result from selecting weighting factors that are appropriate to a specific application.

An example application of a parallel robot in a very demanding magnetic environment has been presented. The control system, incorporating the prediction part based on the biomechanical model of organs, has been proposed.

Acknowledgements.

I would like to acknowledge the following students for their contributions to this work: NUWAR Project – Shane Parker, Josh Male, Lahlan Roberts, Stephen Le Page, Frances Garland, Nadine Frame, Ben Hawkey; Linear Delta Project – Michael Stock, Robert Davis, Mattew Fleming, Dane Howarth. Contributions of Dr Kiyoyuki Chinzei (currently at Surgical Planning Laboratory, Harvard Medical School) to Medical Robotics project are gratefully acknowledged. This work was funded by the Faculty of Mathematical and Engineering Sciences of The University of Western Australia and The Australian Research Council.

References

- ADAMS (1993) Mechanical dynamics. Michigan, USA. Reference Manual, Version 7.0.
- AutoCad (1997) Release 14 users guide. Autodesk, Inc, USA.
- Chinzei K, Miller K (2001) Towards MRI guided surgical manipulator. *Medical Science Monitor*, 7(1):153–163.
- Clavel R (1991) Conception d'un robot parallele rapide a 4 degres de liberte. PhD thesis, Swiss Federal Institute of Technology (EPFL), Lausanne. No. 925.
- Gosselin C, Angeles J (1989) The optimum kinematic design of a spherical three-degree-of-freedom parallel manipulator. *ASME Journal of Mechanisms, Transmissions and Automation in Design*, 111(2):202–207.
- Merlet JP (1990) Les robots paralleles. Hermes, Paris.
- Miller K, Chinzei K, Orssengo G, Bednarz P (2000) Mechanical properties of brain tissue in-vivo: experiment and computer simulation. *J. Biomechanics*, 33:1369–1376.
- Miller K, Clavel R (1992) The lagrange-based model of delta-4 robot dynamics. In *Robotersysteme 8*, pp 49–54. Springer-Verlag.
- Stamper RE, Tsai LW, Walsh GC (1997) Optimisation of a three dof translational platform for well-conditioned workspace. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, vol 4, pp 3250–3255, IEEE Piscataway, NJ USA.
- Tsai LW (1999) Robot analysis: The mechanics of serial and parallel manipulators. John Wiley & Sons, New York, N.Y.

Part 4

Humanoid Robotics II

Session Summary

Tomomasa Sato

Chair

University of Tokyo

Humanoids have the important roles of human-robot interaction beyond walking. The session covered such scientific aspects of humanoids by the following four papers. The first paper touched upon the psychological aspect of human robot interaction; ‘‘Robovie’ – An interactive Humanoid Robot – Interdisciplinary research approach between cognitive science and robotics’ by H. Ishiguro. Questions and answers were concentrated on the communication channels between human and robot. Facial expression is needed especially for elderly people and handicapped people. The people can touch the skin sensors on the body, arms and shoulders, because the torque is very low. The system is not equipped with powerful hands, because it is designed just for communication purpose. These discussions made clear the new aspect of the robotics research.

The second and the third paper discussed on the learning and imitation aspects through human robot interaction. They were ’Learning and Cooperative Multimodal Humanoid Robots’ by R. Dillmann and ’Real-Time Statistical Learning For Robotics and Human Augmentation’ by S. Schaal. The last paper of ’A New Generation of Compliance Controlled Manipulators with Human Arm Like Properties’ by R. Koeppe was on humanoid manipulator targeted to human arm.

The third talk by S. Schaal described a statistical learning framework that is well suited for real-time learning in complex high-dimensional motor control problems. The essence of the learning methods was to employ piecewise linear function approximation to represent complex nonlinear functions, and to use probabilistic dimensionality reduction techniques to keep the computational complexity of the learning low. Results on learning in a humanoid robot were presented, including real-time inverse dynamics and inverse kinematics learning and learning from imitation. Interactions with the audience emphasized that the presented learning methods are not the universal solution to learning function approximation. It is also necessary to note that the local linear learning approach requires a proper initialization of some initial learning parameters. The presented learning tools should have large applicability in many real-time learning applications in robot and human interaction.

The session time is too limited to discuss the whole matter of human-robot interaction, however, the session clearly showed that the human and humanoid interaction has been forming a new and important area of robotics.

Development of an Interactive Humanoid Robot “Robovie” – An interdisciplinary approach

Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, and Takayuki Kanda

ATR Media Information Science Laboratories & Department of Computer & Communication Sciences, Wakayama University

Abstract. We have developed a humanoid robot called “Robovie”. The task is to communicate with humans and establish relationships by using various sensors and actuators. For designing the robot behavior, we have performed cognitive experiments, implemented the results on the software architecture, and verified the effectiveness in human-robot communication. This paper proposes an interdisciplinary approach between cognitive science and robotics for developing the communicative robot.

1 Introduction

There are two research directions in robotics; one is to develop task-oriented robots that work in limited environments and the other is to develop interaction-oriented robots that collaborate with humans in open environments. Industrial and pet robots are the former ones. They perform particular tasks such as assembling industrial parts, behaving like an animal, and so on. On the other hand, the purpose of the robot that we are developing is not to perform particular tasks. We are trying to develop a robot that exists as our partner in our daily life. The fundamental requirement of humans in our daily life is to communicate and recognize the existence each other. Our robot supports such an aspect of our life and provides rich information to humans by using the communication functions. We consider, the robots existing as our partners will be a new information infrastructure for communication.

For realizing the robot, we are tackling to establish a new collaboration between cognitive science and robotics. Cognitive science, especially on ideas of body properties for communication, helps to design more effective robot-behaviors for interacting with humans. On the other hand, the developed robot can be used for verifying theories of cognitive science. We consider this unique inter-disciplinary relationship enable us to develop a new type of robot.

This paper, first of all, reports the developed robot called Robovie. Then it shows two important cognitive experiments. Based on the experiments, the last section discusses a new robot-architecture for generating episode chains in our daily life.

2 Robovie: An Interactive Humanoid Robot

We have developed a robot called “Robovie” (Ishiguro, Ono, Imai, Maeda, Kanda, Nakatsu, 2001) shown in Fig. 1. The robot that has a human-like appearance is

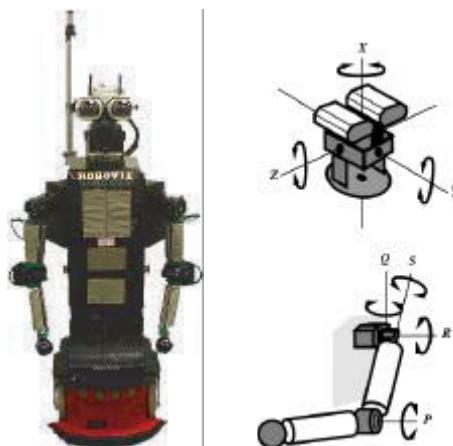


Fig. 1. Robovie

designed for communication with humans. Like a human, it has various sensors, such as vision, sense of touch, audition and so on. With the human-like body and sensors, the robot can perform meaningful interactive-behaviors for humans.

[Hardware] Figure 1 shows the developed robot. It is a humanoid-type robot that moves with two driving wheels. The size is important as an interactive robot. Not to give an awful impression to humans, we have decided the size as 120 cm, which is same as a junior school student. The diameter is 40 cm and the weight is about 40 Kg. The robot has two arms (4*2 DOF), a head (3 DOF), two eyes (2*2 DOF for gaze control), and a mobile platform (2 driving wheels and 1 free wheel). The robot farther has various sensors, skin sensors covering the whole body, 10 tactile sensors around the mobile platform, an omnidirectional vision sensor, two microphones to listen human voices, and 24 ultra-sonic sensors for detecting obstacles. The eye has pan-tilt mechanism with direct-drive motors and they are used for stereo vision and gazing control. The skin sensors are important for realizing interactive behaviors. We have developed a sensitive skin sensors using pressure sensitive conductivity rubber. Another important point in the design is the battery life. This robot can work 4 hours and charges the battery by autonomously looking for battery stations. With the actuators and sensors, the robot can generate almost all behaviors needed for communication with humans.

[Software] Robovie is a self-contained autonomous robot. It has a Pentium III PC on board for processing sensory data and generating gestures. The operating system is Linux. Since the Pentium III PC is sufficiently fast and Robovie does not require precise real-time controls like a legged robot, Linux is the best solution for easy and quick development of Robovie's software modules.

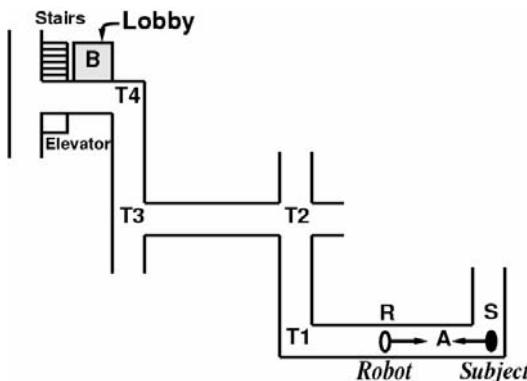


Fig. 2. The structure of the experimental environment

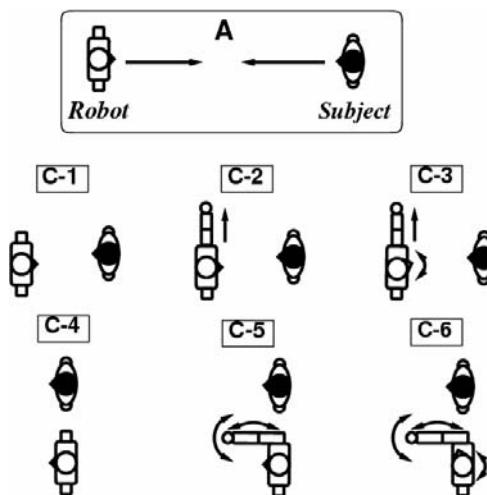


Fig. 3. Six levels of robot's gestures

3 Two Cognitive Experiments

With this robot, we have performed two experiments for human-robot communication in cognitive science. As the results, we have obtained two important ideas: one is importance of physical expressions using the body and the other is effectiveness of robot's autonomy for robot-voice recognition by humans. In other words, the ideas are based on “joint attention” between the robot and a human.

3.1 Mutual Entrained Gestures in Human-Robot Communications

Mutual entrained gestures are important for smooth communications between Robovie and a human. We have performed psychological experiments to en-

sure it Iverson, Goldin-Meadow (1998); McNeill (1987); Morris, Collett, Marsh, O'Shaughnessy (1979); Ono, Imai, Ishiguro (2001); Ono, Imai, Nakatsu (2000). The aim of the experiments was, concretely speaking, to investigate correlations between body movements and utterance understanding in human-robot communications. The detail is summarized as follows.

[Experiments] We focused on the interaction between a subject and a robot while it teaches a route direction to the subject, and investigated the appearance of the subject's gestures and the level of the utterance understanding by using several different gestures in the teaching.

[Subjects] For this experiments, we asked collaboration to thirty undergraduate and graduate students as the subjects, and randomly divide them into six groups. The subjects had not previously visited this experimental environment.

[Environment] Figure 2 shows hallways in our laboratory. Points S and R denote the initial positions of a subject and the robot, respectively. The robot taught a route direction to the lobby B at A.

[Procedure] The experiments consist of the following three phases:

1. The subject and the robot move from S to A and from R to A, respectively.
2. At A, the subject asks a question "Tell me the way to the lobby", and the robot begins to explain the route. The robot says "Go forward, turn right, turn left, turn right, turn left, and then you will arrive at the destination." While speaking, it performs gestures in one of the six levels described in [Condition]. The purpose of this experiment is to investigate relations between the six levels of robot's gestures and emerged human's gestures.
3. The subject tries to go to the lobby. When the subject arrives at the lobby or it gives up by losing the way, the experiment finishes.

[Conditions] As conditions of the experiments, we have prepared the six levels of robot's gestures as shown in Fig. 3.

C-1 The robot does not move.

C-2 The robot raises the left arm leftward when speaking "Go right" and rightward when speaking "Go left".

C-3 In addition to C-2, the robot turns the eyes to the subject while talking.

C-4 The robot stands side by side and directs the body along the hallway.

C-5 In addition to C-4, the robot raises the right arm forward, rightward and leftward when it teaches the directions.

C-6 In addition to C-5, the robot turns the eyes to the subject while talking.

Figure 4 shows the ratio of subjects' body movements under the six levels. We have classified the body movements into three categories: no body movement (Nothing), hand movements (Hand) as shown in the left photo of Fig. 5, and raising hands up to the elbow level (Elbow) as shown in the right photo of Fig. 5. Figure 4 shows a significant changes of subject' gestures against the conditions ($\chi^2 = 25.210, p < 0.01$). As the level changes from 1 to 6, the subjects perform bigger gestures. Moreover, the average numbers of times that the subjects gaze the

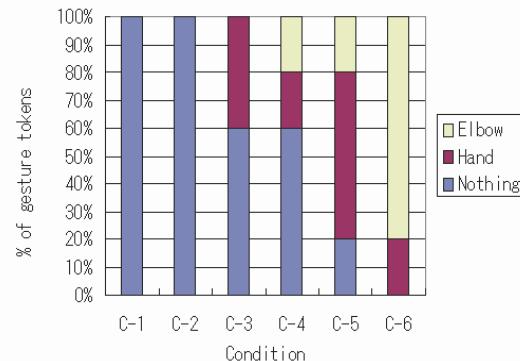


Fig. 5. Scenes of human-robot interactions

robot were as follows: 0.8 (C-1), 1.0 (C-2), 2.0 (C-3), 1.2 (C-4), 1.0 (C-5), and 3.8 (C-6).

Further, we recorded the time that the subjects spent to move from A to B in Fig. 2. Table 1 shows the average time and the number of subjects who did not arrive at B. Regarding the average times, there is no significant difference among the conditions, but the average time in C-6 is shorter than others. A more noteworthy point is that a considerable number of subjects could not arrive at the destination in C-1, C-2, and C-3. The reason found in the questionnaire is that they could not understand robot's utterance. Especially, they confused to understand the meaning of "left" and "right". However, in C-4, C-5, and C-6, there is no subject who could not arrive at the destination. This means that they could obtain a joint viewing point by the robot's gestures.

We conclude these experimental results as follows:

1. Many and various behaviors of the robot induce various human communicative gestures. In other words, the subject's gestures are increased by entrainment and synchronization with the robot and a relationship between the robot and the subject is established from the mutual gestures.

Table 1. Average time to arrive at the destination and the number of failed subjects

	C-1	C-2	C-3	C-4	C-5	C-6
Time to destination	69.5	71.3	67.7	70.2	66.8	65.4
Number of subject not arriving	1	2	3	0	0	0

2. The emerged mutual gestures help to understand robot's utterance.
3. The joint viewpoint represented by the robot gestures allows the subject to understand the utterance.

3.2 Joint Attention in Human-Robot Communication

The experiment shown in the last sub-section clarified the importance to share a joint viewing point in human-robot communication. The results suggest proper robot behaviors in the development of everyday robots. The concept of the joint viewing point can be extended as the concept of joint attention and it gives more proper robot behaviors for interacting with humans Imai, Ono, Ishiguro (2000).

The relevance theory (Sperber, Wilson, 1986) proposes a communication model for recognizing situations and the humans' experiences. It employed a new term named mutual manifestness that represents mental state where two or more humans recognizes the same situation or recall similar experience. The relevance theory regards human's communications as a process of gaining mutual manifestations by passing messages to others. This concept of mutual manifestness is same as that of focus of attention; and it is called "joint attention" in social psychology (Moore, Dunham, 1985) in the case where people frequently focus on the same object while communicating each other.

However, mechanisms of focus of attention proposed so far are insufficient for developing a speech generation system depending on the situations. The following three difficulties have to be overcome in human-robot communication.

1. How to draw a human's attention to the target to which a robot is paying attention.
2. How to make a human realize the intention of the robot.
3. How to utilize the human's attention in a robot mechanism for communication.

Difficulty 1 is attributed to the lack of an expression when the robot pays an attention. Without the expression, the human cannot realize where the robot is paying its attention. That is, the human and the robot are not in a state of mutual manifestness in terms of the relevance theory. The lack of a human's attention in a conversation is a crucial problem. For example, when a guide robot in a museum focuses on an artwork and begins to explain it to a human by using demonstrative pronouns, the human may not be able to understand what the robot is explaining if the human is not focusing on the artwork. This difficulty is overcome by adding attention expression behaviors to the robot. Here, we have implemented two behaviors: gazing head motion to face the target and hand gestures to point at it.

Difficulty 2 comes from Difficulty 1. The relevance theory insists that the occurrence of a state of mutual manifestation depends on the inference of the speaker's communicative intention. For example, when the robot says that "Take this away." In front of a box in order to proceed forward, the human has to use the robot's communicative intention to interpret the robot's utterance. If the human pays its attention to the box, the situation can be recognized as a state of mutual manifestation with the robot. This difficulty is overcome by employing an eye contact behavior. The robot turns the head direction to face the human to promote the relationship with the human. The eye contact inspires the human to guess the robot's intention and to become aware of the robot's attention manifested by the attention expression.

Difficulty 3 is attributed to the joint viewing point problem as discussed in the previous sub-section. By sharing the joint viewing point, a human can easily recognize the robot's utterance even if it omits some concrete words. This effect is not only for sharing the joint viewing point, but also to have a proper positional relation among a robot, a human, and a target. For example, when the robot ask to move a box locating in front away, if they have the proper position relation, it can say just "Move it away". The function to share a joint viewing point and to establish a proper positional relation is effective for smooth communication.

We have verified the effect of the robot's behaviors discussed above. First of all, we have prepared two groups each of which consists of six subjects: one was given Robovie with eye contact, and the other was given Robovie without eye contact. Robovie performed the attention expression for both groups. The target of the attention expression was a poster on a wall. The experiment recorded the number of subjects who looked at the poster according to the attention expression.

The experimental procedure is as follows. At first, the robot passes in front of the subject, and stops in front of the poster, where both the robot and the poster are in the subject's sight. At the location, the robot turns to the subject and points to the poster with its arm while speaking "please look at this". Here, the robot performs eye contact to the subject.

Table 2 shows the results. The results indicate that the subjects with eye contact (the upper photo in Fig. 6) look at the poster (the lower photo in Fig. 6), and the subjects without eye contact look at the robot arm instead of the poster. That is, eye contact is significantly effective for achieving joint attention ($\chi^2 = 8.57, p < 0.01$); and the robot behaviors designed based on the discussions of the difficulties 1–3 are proper for establishing a communicative relationship with a human.

Table 2. Comparison of the number of humans who looked at a poster pointed out by the robot

	Saw a poster	Saw Robovie's hand
With eye-contact	6	0
Without eye-contact	1	5

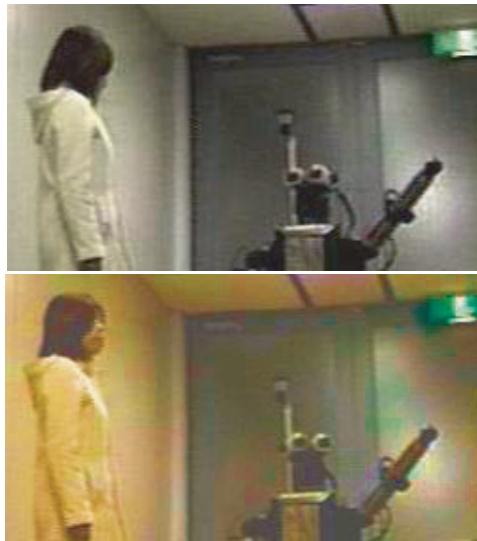


Fig. 6. Eye contact between subjects and the robot

4 A Robot Architecture for Generating Episode Chains

From the psychological experiments discussed in Section 3, we have obtained four ideas as follows:

1. Rich robot's behaviors induce various human communicative gestures that help utterance understanding.
2. Attention expression by the robot guides the human's focus to the robot attention.
3. Eye contact by the robot indicates robot's intention of communication to the human.
4. Sharing of a joint viewing point and a proper positional relation establish a situation where the human can easily understand robot's utterance.

Based on these ideas, we have designed a new architecture of the robot and implemented to the developed robot "Robovie". The basic structure of the architecture is a network of situated behavior modules. Figure 7 shows the meta-structure of Robovie's software. All of the behaviors are classified into four categories; and Robovie performs behaviors belonging to one of them. A unique point is that the category "Play with humans" has two sub-categories of greeting to say "Hello" or "Bye" when switching the category.

The behavior models belonging to the category include elemental behaviors for communications as shown in Fig. 8. The elemental behaviors that implement the above-mentioned ideas are the most important point in this architecture. The robot behaviors developed so far do not have the function to entrain humans into the communication. By combining the elemental behaviors and other task-oriented behav-

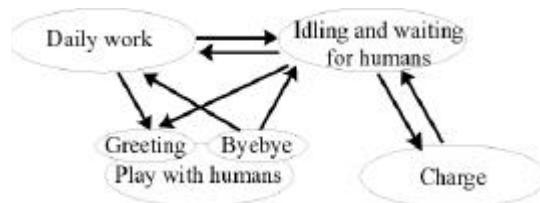


Fig. 7. The meta-structure of the architecture

iors, we can realize various interactive behaviors. Fig. 9 shows an example of the interactive behavior that the robot asks a human to look at a poster.

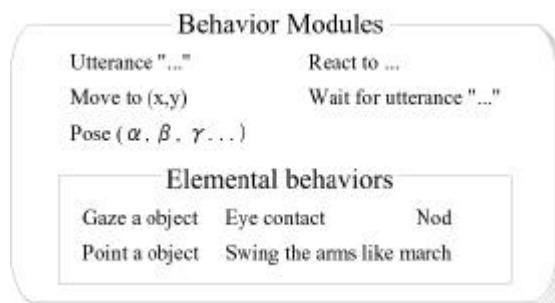


Fig. 8. Behavior modules

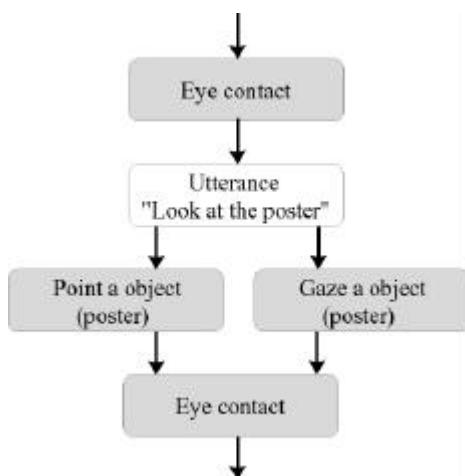


Fig. 9. An example of behavior module

Figure 10 shows the all-over software architecture. Basically, this is an extension of the architecture based on situated modules (Ishiguro, Kanda, Kimoto, Ishida, 1999; Ishiguro et al., 2001). The architecture proposed in our previous work has two merits: easy development of behavior modules and robust execution by dynamic switching of the behavior network. With keeping the merits, we have extended the architecture.

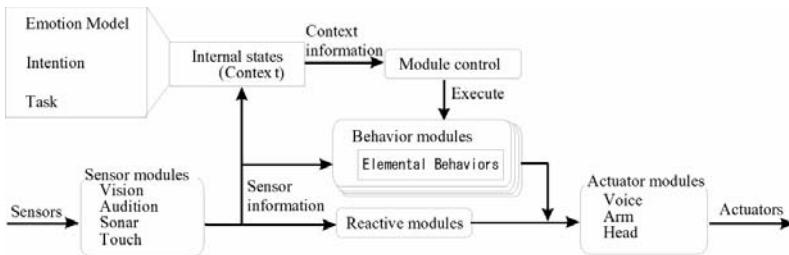


Fig. 10. Software architecture based on behavior modules

Episodes between a robot and humans are emerged through interactive behaviors and contextual chains of the behavior. The behavior modules shown in Fig. 10 form a network based on these execution orders, and the network can generate various sequential orders among the interactive behavior modules. By switching the execution order based on sensory input, the robot can generate various episode chains depending on the situations. This episode chain is not a still behavior sequence. A behavior module in a previous robot is activated based on sensory in-put. On the other hand, our robot controls the behavior sequence based on predefined weak orders even if it does not sufficient sensory data and continuously entrains the human. We consider the episode chains will represent the robot's autonomy.

These two ideas are also important to implement sensory data processing. In previous robotics, robots needed to perform perfect sensor data processing to execute particular tasks. The robot in this paper, however, entrains humans into the interaction loops by the interactive behaviors and it does not require perfect sensory data processing. Humans, rather, adapt to the robot's ability. With the architecture, the robot can continuously generate rich behaviors for communication even if the sensory data processing is not perfect.

Finally, Fig. 11 shows all of developed modules and their relationships. The robot behaviors generated from the various interactive behaviors and the complicated network has given human-like impressions. The typical behaviors of Robovie are: “greeting”, “hand-shake”, “playing the game of ‘paper, stone and scissors’”, “hugging”, “kiss”, “short conversation”, “exercise”, “pointing the poster”, and “saying good bye”. Robovie also takes idling behaviors such as “scratching the head”, “folding the arms”, and so on. Figure 12 shows a scene where two Roboties interact with a chilled. As shown in the figure, the child naturally communicated and played

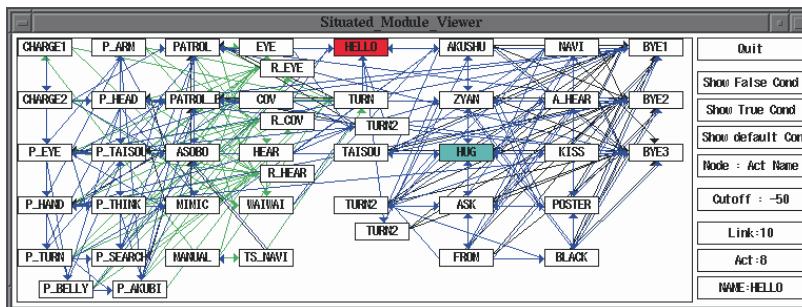


Fig. 11. All behavior modules and their relationships

with Robovie. It is difficult to explain how impressive this scene is, but we have felt strong possibilities of robotic creatures.

5 Conclusion

This paper has reported on a new humanoid robot called “Robovie”. The unique aspect of Robovie is the mechanism designed for communication. Robovie can generate human-like behaviors with the actuators and sensors. In the design, we have performed two psychological experiments and developed the behaviors obtained from them. Our next step is to implement more interactive behaviors to Robovie and try to establish more sophisticated relationships between the robot and humans.

We have started this project on August 1999. After the development of Robovie on July 2000, Robovie has appeared in many robot exhibitions and been reported by almost all major newspapers and several TV programs in Japan. These are not only advertisements but also valuable chances to gather comments from ordinary people. For developing a robot work in our daily life, these activities bring much information in addition to the cognitive experiments. For more detail of this project, please refer to the following WebPages:

<http://www.mic.atr.co.jp/michita/everyday-e/>



Fig. 12. Interactions between a human and Robovie

References

- Imai M, Ono T, Ishiguro H (2000) Attention mechanism for utterance generation. In Proc. 9th IEEE Int. Workshop Robot and Human Communication, pp 1–6.
- Ishiguro H, Kanda T, Kimoto K, Ishida T (1999) A robot architecture based on situated modules. In Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp 1617–1623.
- Ishiguro H, Ono T, Imai M, Maeda T, Kanda T, Nakatsu R (2001) Robovie: A robot generates episode chains in our daily life. In Proc. Int. Symposium Robotics, pp 1365–1361.
- Iverson J, Goldin-Meadow S (1998) Why people gesture when they speak. *Nature*, 396(228).
- McNeill D (1987) Psycholinguistics: A new approach. Harper & Row.
- Moore C, Dunham PJ (1985) Joint attention: Its origins and role in development. Lawrence Erlbaum Associates, Inc.
- Morris D, Collett P, Marsh P, O'Shaughnessy M (1979) Gestures: Their origins and distribution. Stein and Day, New York.
- Ono T, Imai M, Ishiguro H (2001) A model of embodied communications with gestures between humans and robots. In Proc. 23th Annual Meeting of the Cognitive Science Society, pp 732–737.
- Ono T, Imai M, Nakatsu R (2000) Reading a robot's mind: a model of utterance understanding based on the theory of mind mechanism. *Advanced Robotics*, 14(4):311–326.
- Sperber D, Wilson D (1986) Relevance: Communication and cognition. Oxford: Basil Blackwell.

The German Collaborative Research Centre on Humanoid Robots

Rüdiger Dillmann

Institute for Process Control and Robotics, Haid-und-Neu-Str. 7, 76131 Karlsruhe, Germany

Abstract. In this paper a brief overview on the German Collaborative Research Center on Humanoid Robots is given. The research topics focused by the collaborative research center are multimodal man-machine interaction, learning of motion, skills and tasks, man-robot cooperation, perception and action and mechatronics and controls for a human like torus consisting of 2 redundant arms, 2 dexterous 5-finger hands, a head with eyes and ears, a neck and a spine. The project started in July 2001 and is planned in long terms for a period of up to 12 years. Over this period it is envisaged to elaborate and establish basic methodologies and technology for humanoid robots to act and cooperate close with humans in daily living environment. An integrated perspective towards continuous humanoid inter-action gaining multimodality, adaptivity, redundancy and flexibility is proposed.

Programming, cooperation and interaction with a humanoid robot is assumed to be multimodal which means to let the user program the robot simply by speech, gesture or demonstrating a task. The robot observes, interprets and then tries to imitate and to learn the performed user action. On the basis of a redundant humanoid type 2 arm robot system equipped with an active stereo head and a microphone array observation of the demonstrated task is realized as a first prototype using an active vision system. Grasping of objects is detected with the help of data gloves and active vision. The system interprets and stores the observed actions, segments them into meaningful sequences in a given context. Due to sensor errors and the complexity of the intended interaction, the system generates queries concerning intention, manipulation and objects. Motion catch techniques in combination with learning capabilities are combined towards imitation learning and active observational learning strategies.

1 Introduction

In July 2001, the German Collaborative Research Center (GCRC) 588 has been started at the University of Karlsruhe. The interdisciplinary overall project period is expected for about 10 – 12 years, and it addresses a set of basic technical and scientific key problems in the field of humanoid robotics. The main goal of the project is to build an anthropomorphic robot which is able to cooperate with humans and to behave similar to humans in everyday environments like e.g. a household. On the one hand, it should be possible to command the robot in an intuitive natural way using different input channels like speech, gesture or haptics. On the other hand, man-machine-interaction with direct physical contact for cooperative manipulation of objects is also an important aspect. The intention of the project is not to build a robot to replace humans, but to serve them and to share the workspace as flexible, easy-to-use machine with a human-like behaviour. Human-like response requires an extensively integrated humanoid system with an appropriate human like body.

To achieve this goal a common representation and interpretation of the world is required. The context in which we establish the research is focused on continuous adaptive humanoid interaction and cooperation.

As we believe, programming by demonstration and advice (PbD) and imitation learning is a technique that overcomes the drawbacks of classical approaches and could be a natural way to program humanoid robots. Providing multimodal bi-directional interfaces allows both modelling of motion and interaction but also setting up a common representation of the world programmed in an intuitive mode. Thus, human motion can be traced, segmented and interpreted. The observed action sequences can be mapped into a generalized representation and finally adapted and executed by a target robot. Full body motion capture techniques combined with statistical learning techniques (Schaal, 1999) are considered to implement imitation learning of natural human movement by the robot. If more natural motion behaviour can be achieved a higher acceptance rate for robots in various applications may be the result Billard, Mataric (2001).

Several PbD systems and approaches for different applications have been proposed in the past. An overview and classification can be found in (Dillmann, Rogalla, Ehrenmann, Zöllner, Bordegoni, 1999). Learning of skills, action sequences or operation plans requires modelling of cognitive skills. In fact, the purpose of observing the user is to generate an abstract description of the demonstration, reasoning about the user's intention and modelling the problem solution preferably in an optimal way. Likewise, a given problem has to be generalized by identifying for example significant parameters being characteristic for a particular demonstration and by deriving parameters being relevant for the problem concept. Both requires background knowledge about the environment and about the user's behaviour and performance.

The analysis of demonstrations may be based on the observation of dynamic transitions in the workspace manipulated by the user. These dynamic transitions can be described using relational expressions or contact relations (Onda, Hirukawa, Tomita, Suehiro, Takase, 1997). For generalising a single demonstration typically explanation based methods are used (Friedrich, Grossmann, Ehrenmann, Rogalla, Zöllner, Dillmann, 1999; Mitchell, 1986) because they allow an adequate generalisation taken from only one example (One-Shot-Learning).

Because physical demonstration in the real world can be very time-consuming some researchers rely on virtual demonstrations Archibald, Petriu (1993). Augmented reality (AR) techniques and simulation allow some additional degrees of freedom to the user. Hence, in some approaches object poses or trajectories Kang, Ikeuchi (1997); Takahashi (1993); Tung, Kak (1994) and/or object contexts Friedrich, Münch, Dillmann, Bocionek, Sassin (1996); Onda et al. (1997); Segre (1989) are acquired and generalised with the help of such techniques.

Iconic programming as a supplement to AR is based on a more abstract level making use of a pool of already existing or acquired basic skills. These can be retrieved through specific cues (like icons) and embedded in an action sequence. The result of such a demonstration of operation sequences can be generalized in terms of

context, conditions, goal states and perceptive skills and embedded as new knowledge into the system. An example can be found in the skill-oriented programming system SKORP Archibald, Petriu (1993) that helps setting up macro operators out of actuator or cognitive elementary operators interactively.

In addition direct cooperation between the user and the robot has been investigated. Here, user and robot operate in a common working space. The user may direct the robot with the help of a common vocabulary like speech or gestures (Steinhage, Bergener, 1998; Voyles, Khosla, 1999; Zhang, von Collani, Knoll, 1999).

For the humanoid robot project experiments following the PbD paradigm have been undertaken in a stationary environment.

The perceptive capabilities of the robot are essential for the humanoid robot to be able to interact within a daily dynamic environment including humans. An active stereo vision system with steerable cameras and a microphone array integrated into a head combined with a flexible neck is used to identify objects, recognizing people and identifying their focus of attention. The purpose is tracking of people and analyzing their intention and activities. The perceptive and control system steers the robot behaviour. Typical capabilities of the audio- vision system include face recognition, people tracking, lip reading, gesture recognition, identification focus of attention, speech recognition as well as estimation of speaker position Kroschel, Mekhaiel, Crowley, Berard (1995); Stiefelhagen, J.Yang, A.Waibel (1999); Yang, Stiefelhagen, Meier, Waibel (1998).



Fig. 1. The ARMAR System

In the following, an overview on the humanoid robot project and especially on recent work on man-machine interaction and cooperation is given. The GCRC addresses scientific problems in four different fields: mechatronics and controls for humanoid robots, multimodal man-robot interaction and cooperation, perception, real world modelling and skill adaptation and learning.

1.1 Mechatronics and Controls for Humanoid Robot

With the experience from the ARMAR system, see Fig. refFIG:figure1, as a prototype platform we started a new design for a humanoid robot with a holonomic mobile wheeled platform, a flexible spine torso, two redundant 7-axis arms, two 5-finger hands (fluidic technology) and a complex sensor head. The robot is equipped with ultrasonic sensors, laser range sensors and artificial skin for collision avoidance, stereo and colour vision for visual servoing, object recognition and user tracking, microphone arrays and directive microphones for voice recognition and user localization.

The redundant lightweight arms are to be equipped with dexterous 5-finger fluid actuated hands. All drives, computing and sensor components are to be integrated into the humanoid robot system and is one of the main project goals. The coordinated control of the highly redundant system consisting of both arms in combination with the torso (even the platform) and the head is another key research issues. The goal to be achieved is to build a system which is able to observe and to imitate human movements and to map it onto the robot considering kinematic and dynamic constraints. This includes also human-human cooperation strategies to be transferred to the man-machine cooperation level.

The robot supports an extensible skill-interface which will be used by the higher-level planning and cooperation units.

1.2 Multimodal Man-Robot Interaction

To improve intuitive interaction the robot must provide a multimodal user interface. In the GCRC, techniques are focused which enable the user to communicate via speech, via gesture and via direct physical interaction with the robot system.

Speech recognition will allow for user independent, continuous speech even with background noise. An acoustic map of the environment represents sources of sound and noise. From that active filtering algorithms can be derived to increase the speech recognition rate as well as to increase the rate of identifying speakers and their location. In many cases the robot has to clarify via a spoken dialogue uncertain advice or to give status information. Therefore, the robot must be able to synthesize speech to communicate in a natural manner. Dialogues are necessary for all higher level interaction like teaching and learning of tasks or learning of cooperation knowledge. Therefore a considerable amount of effort has to be spent on creating a dialogue engine based on information frames that can be used by all software modules to acquire information in the case of ambiguities. Dynamic vocabularies and grammars are foreseen to allow dialogues on real world structures and events and on tasks or interaction with robots.

Gesture recognition enables the user to control and command the robot in a straightforward way. Additionally, it extends the intuitive speech interface by pointing gestures (e.g. to select the objects to be grasped). This additional information gathered by gesture recognition will be fused with the dialogue context to a multimodal environment context in order to understand focus of interest, what the user

desires and what he is talking about. As an extension of this context the interactive environment modelling (the user explains a new environment structure and situations to the robot) is a very important scientific challenge in that field. This yields to dynamic vocabularies and dynamic grammars.

A part of the multimodal environment context is the actual user state configuration. The robot needs all information about the user (head, arms, hands etc.) that is available. Thus, the gesture recognition will be extended to complete upper body recognition and later on to the entire human to identify his intention and the context of actual situations.

Since in man-machine-cooperation it is often necessary that humans are able to influence during action the robot's motion, tactile sensors are positioned on the robot body skin to detect and to react on touch or on contact. Thus, the robot can be moved via direct haptic contact in zero force control mode. In case of a collision with the user or with the environment a tactile array may trigger a collision avoidance reflex. Which mode to choose depends on the user context.

1.3 Man-Robot Cooperation

The man-machine cooperation problem needs solutions on all levels of abstraction in the robot soft- and hardware. Next to the hardware constraints like artificial skin and force/torque sensors for user safety and zero force control there are many software problems. Starting from adaptive control of the robot kinematics in combination with the tactile sensors, there are high level tasks like recognizing the user intention (by the use of the multimodal context and a database of learned tasks), selecting the appropriate cooperation pattern (learned by programming by demonstration) and selecting the correct competences for a specific instantiation of the pattern.

Initially the robot needs input by the user for recognizing the intention (e.g. to transport an object cooperatively). On this basis, the robot should be able to build up hypotheses of the user intention and to use dialogues in case of ambiguities.

Given the correct user intention and data about the intended discourse, adequate cooperation pattern can be selected from a database of pre-learned patterns (e.g. cooperative manipulation) and can be adapted to the specific task (dedicated objects, constraints, needs,..).

After this instantiation the cooperation patterns can be sequenced into special basic cooperation skills (like shared object manipulation, shared grasping etc.).

1.4 Adaptation and Learning

Various learning strategies may be applied on all levels of abstraction in the cognitive and perceptive architecture of the humanoid robot:

- Learning of natural type basic movements (human-like movement)
- Learning of coordinated movements (human-like movement)

- Learning of tasks (e.g. assembly, fetch and carry, loading, unloading, opening/closing a door, cleaning etc.)
- Learning of cooperative tasks or task-pattern

Learning and mapping of natural basic movements requires adaptive controls for the whole robot body motion system. Learning of coordinated movements includes learning of all control sets for entire-arm-movements, entire-body-movements etc.

Learning of manipulative tasks may be based on the programming by demonstration paradigm. The user demonstrates the solution of a problem to the robot. The robot observes the solution and tries to imitate. Basic skills or task solutions may be generated with the help of a training center which can be a simulator to be accessed via a network.

Simulation allows to correct the generated robot program. Such a program may consist of an operator tree based on a huge set of basic skill operators.

Learning of cooperative tasks is similar to learning autonomous tasks. The robot is incorporated into the process by being commanded via its multimodal interface. The entire user demonstration including its own actions is recorded and post processed. For activating the patterns or execution of them the specification of competences or roles (who does a specific action in a concrete case ?) between the robot and the user is required.

1.5 Programming by demonstration for manipulation tasks

In the following our approach to PbD is discussed. One key issue for the robot is the interpretation of what has been done by the human demonstrator. If the humanoid robot can derive for a given user demonstration a clear and correct hypothesis, it enables the robot system to reuse this observed action in similar or slightly different situations. It can be shown that the process of learning complex problem solving knowledge can be processed by a sequence of well defined and different computing phases.

2 The PbD Process

In Fig. 2, the whole programming process is schematically illustrated. It starts with a user task demonstration which is observed by a sensor system. Within a sequence of 7 processing phases the data derived from the human demonstration is converted into a control program:

1. A sensor systems observes the user's intention, motion, interaction and behaviour. Dynamic changes of object positions and constraints in the environment can be detected. If context information is given as input, the system's sensor processing and tracing performance can be improved considerably.
2. During the next phase relevant operations and environment state changes can be derived based on the acquired sensor data. This process is called segmentation. Segmentation can be performed online during the observation process or

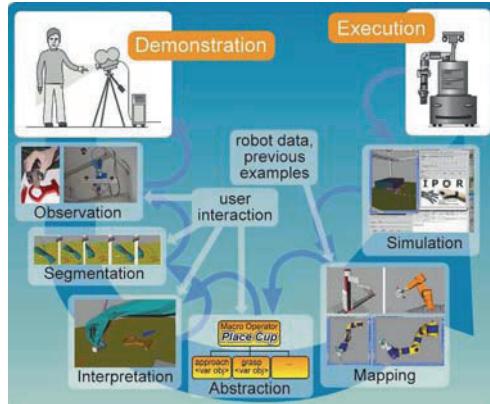


Fig. 2. PbD Process

offline based on recorded data. The segmentation processing performance can be improved significantly by user input concerning context and purpose of the observed interaction to increase the systems hypothesis generation capabilities.

3. Within the interpretation phase the segmented user task demonstration trace is mapped onto a sequence of symbolic descriptions. These symbols contain information about the action (e.g. type of grasp, trajectory type) as well as important data from the sensor traces like forces, used for grasping, etc.
4. Abstraction from the given demonstration for representing the task solution as general as possible is the next processing step. Generalization of the obtained operators may include further advice by the user. Spontaneous and not goal oriented motion and actions may be identified and filtered in this phase. The task knowledge is stored in a form that allows reuse even if the execution constraints and conditions differ slightly from the demonstration.
5. Transfer of the internal symbolic knowledge representation to the target system has to consider physical properties of the robot. Input to the mapping process is the generated task solution knowledge from the previous processing phase. Background knowledge about the kinematic constraints of the target robot system is matched with the abstract task description.
6. In the following process the mapped action sequence is tested with the help of a simulator. In this phase the user can access the system and can confirm and evaluate performance and correctness to avoid ambiguous situations during real task execution.
7. During the execution process success and failure information can be used to evaluate and modify the actual task program but also the mapping process.

The overall process has been implemented and tested for robot manipulation tasks in general.

The PbD technique we have been applied to pick-and-place operations of various household objects like cups, plates, forks and spoons and to other service tasks.

However, the experiments with real household objects yield to learning of different grasp-types for distinct robots. A 2-arm – 2-hand demonstration system is set up actually to demonstrate simple 2 coordinated hand tasks in the above described environment.

3 System Structure

To meet the requirements for each phase of the PbD-process discussed in the previous section we have implemented a system consisting of the components shown in Fig. 3.

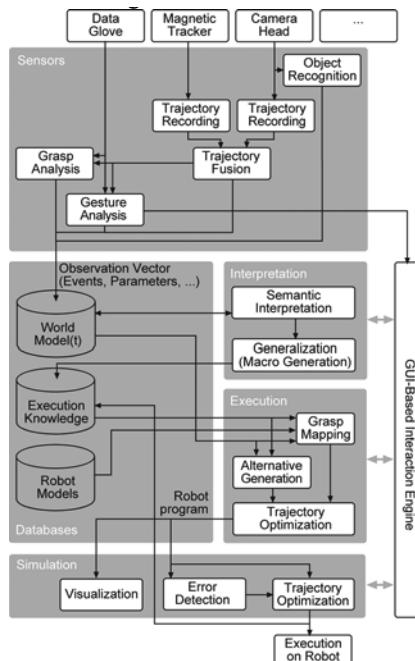


Fig. 3. System Structure

The system integrates four basic modules. The sensor module is responsible for analysis and pre-segmentation of data from the channels connected with the sensor systems. Its output is a vector describing environment states and user actions. This information is stored in a world model database.

The second module operates on the gained observation vectors and associates sequences of observation vectors to a set of predefined symbols. These parameterized symbols represent elementary action sets. During the interpretation phase the symbols are chunked into hierarchical macro operators after replacing specific task-

dependant parameters by variables. The result is stored in a database as generalized execution knowledge.

This knowledge is taken from the execution module which uses specific kinematic robot data for processing. It calculates optimized trajectories for the target system taking into account the actual world model.

Before mapping the generated program to the target system it's correctness is tested by simulation. In case of unforeseen errors or conflicts, the movement of the robot has to be corrected and optimized.

All four components communicate with the user via a graphical user-interface. Additional information can be retrieved or hypotheses can be accepted or rejected. While demonstrating, the user may use gestures for interaction as well.

4 PbD Phases

In the system implementation, every PbD process phase is supported as follows:

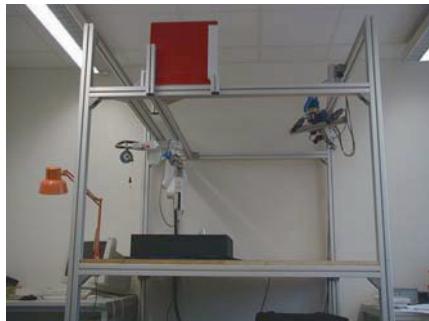


Fig. 4. Training center with sensors

1. During demonstration, the user handles objects in a training center (cf. Fig. 4). This is equipped with sensors like data gloves, magnetic field based tracking systems and an active trinocular camera head. Object recognition is done by computer vision using fast view-based algorithms described in (Ehrenmann, Steinhäus, Dillmann, 1999). With the help of the data glove, the system extracts finger joint movements and hand positions in 3D space. To reduce noise in trajectory information, the user's hand is additionally observed by the camera system. Both measurements are fused using confidence factors, see (Ehrenmann, Zöllner, Knoop, Dillmann, 2001b). Furthermore, pre-trained neural networks classify hand configurations to detect certain events like grasps and gestures. The system is able to distinguish 16 different grasp types and 9 distinct gestures (see (Friedrich et al., 1999)). This information is stored with discrete time stamps in the world model database.

For interpretation, the system applies two types of predefined symbols: first, 16 different grasp types following the Cutkosky hierarchy (see (Cutkosky, 1989)) and second, three types of basic movements (linear, free and spline move). The data from the demonstration (retrieved from the database) is analyzed by an internal heuristic.

The move types are detected by analyzing speed and trajectory type of the user's hand motion. Free moves and spline moves may be selected by thresholds and parameters which can be modified by the user. Result of this phase is a sequence of the proposed associated symbol set including relevant data recorded during demonstration. For instance, finger poses and positions for grasps, particular objects or positions in 3D space.

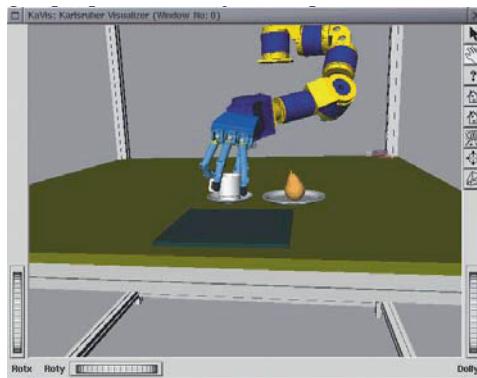


Fig. 5. Simulation of generated program

2. The symbol sequence from the previous processing phase is now chunked into semantically related groups. These are for example: approach phases, grasp and ungrasp phases. This process continues recursively controlled by an internal heuristic. The heuristic is designed by analysis of the grasp and pick-and-place operations. For instance, a macro operator for a simple pick-and-place operation has a tree-like structure starting from the whole operation at the root. The first branches lead to a symbol for a pick and a place. They are subdivided in approach, grasp/ungrasp and disapproach phases. Finally, the process terminates with the elementary operations at the leaves.

Each component in the macro structure has certain context conditions like pre-conditions. They are generated automatically from the demonstration (see (Friedrich, Rogalla, Dillmann, 1998)). Context conditions decide whether a macro operator is suitable for an actual environment situation. It is important to understand that all trajectory parameters stored inside a macro are object-dependant. For instance, during an approach operation all movements are represented relatively to the object of interest. This allows to substitute specific

- objects by variables. If the macro is re-used again, variables become replaced by concrete objects that match context conditions.
3. Up to now, problem solving information stored in the macro operators can not directly used by the target system. Grasps are represented in order to describe human hand configurations and human motion trajectories are not optimal for robot kinematics. Besides, sensor controls for the robot like force control with the help of force/torque sensors can not derived from the demonstration. Thus, additional information like spoken advice may be useful. We have developed a method for automatic mapping of grasp types to robot grippers. Furthermore, stored trajectories in the macro get trimmed for the execution environment and the target system. The system uses generic robot and gripper models described in (Rogalla, Pohl, Dillmann, 2000). This makes it possible to support different gripper and manipulator types for execution. Additionally, we defined a set of logical rules that select sensor constraints depending on the execution context. This is for example to select a force threshold parallel to the movement when approaching an object.
 4. The generated robot program is simulated in a graphical computing environment. Here, the user might modify or reject certain actions and movements (see Fig. 5).

5 Experiments

The demomstration task, we address for the humanoid robot is table laying in a kitchen environment among to demonstrate and evaluate the easibility of our approach. For this purpose, we use a scenario consisting of objects like cups and plates in different forms and colors. The examples were recorded in the training center. For learning basic skills and its execution, the service robot ALBERT was used.



Fig. 6. Service robot Albert

The robot is equipped with a stereo camera head and a 7 DOF modular arm (Amtec, Berlin) (see Fig. 6). The arm weighs about 60 kg and can lift objects of up to 10kg with fully extended modules. A 6 DOF force/torque sensor interconnects the arm with the three finger Barrett hand. The upper body part is mounted on a mobile platform ODETE which was developed at our institute. ODETE is equipped with supersonic sensors and a planar SICK laser scanner for self-localization and obstacle recognition Graf, Weckesser (1998). With it's differential drive, the platform will enable the robot to move freely in a workshop or household environment.

The experiments are to be continued with the humanoid 2-arm-2-hand torso ARMAR.

6 Future Work

For demonstration of various other tasks to the humanoid robot, the multimodal user interface will be enhanced. Several interaction channels will allow intuitive interaction between man and machine:

- Speech recognition will be coupled with face detection and eye tracking.
- Gesture recognition considering static and dynamic gestures. A first approach has already been tested successfully (Ehrenmann, Lütticke, Dillmann, 2001a). Gestures will be used for commanding and commenting a demonstration and for interaction in user dialogues.

Markov representations embedded in a dynamic grammar are planned to describe observed natural motion of humans acquired by motion catch techniques. Cooperation patterns will be modelled to enable the robot to continuously interact with humans in a given context.

6.1 Members of the GCRC

About 40 scientist started working on the project in July 2001. Project partners include the University of Karlsruhe with several institutes belonging to different faculties but also local non-university research institutions take part in the project. The partners are:

- Universität Karlsruhe with 10 institutes
- Forschungszentrum Informatik
- Fraunhofer Gesellschaft (IITB)
- Forschungszentrum Karlsruhe

Acknowledgement.

This work is supported by the Deutsche Forschungsgemeinschaft (DFG) as Collaborative Research Center 588 Humanoid Robots, the DFG project "Programming by Demonstration" and also by the BMBF lead project MORPHA .

References

- Archibald C, Petriu E (1993) Computational paradigm for creating and executing sensor based robot-based programming. In ISIR, pp 401–406, Tokyo, Japan.
- Billard A, Mataric MJ (2001) Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture. *Robotics and Autonomous Systems*, pp 145–160.
- Cutkosky MR (1989) On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279.
- Dillmann R, Rogalla O, Ehrenmann M, Zöllner R, Bordegoni M (1999) Learning robot behaviour and skills based on human demonstration and advice: the machine learning paradigm. In 9th International Symposium of Robotics Research (ISRR '99), pp 229–238, Snowbird, Utah, USA.
- Ehrenmann M, Lütticke T, Dillmann R (2001a) Dynamic gestures as an input device for mobile platforms. to be published.
- Ehrenmann M, Steinhaus P, Dillmann R (1999) A multisensor system for observation of user actions in programing by demonstration. In Proceedings of the IEEE International Conference on Multi Sensor Fusion and Integration (MFI), pp 153–158, Taipeh, Taiwan.
- Ehrenmann M, Zöllner R, Knoop S, Dillmann R (2001b) Sensor fusion approaches for observation of user actions in programing by demonstration. to be published.
- Friedrich H, Grossmann V, Ehrenmann M, Rogalla O, Zöllner R, Dillmann R (1999) Towards cognitive elementary operators: grasp classification using neural network classifiers. In Proceedings of the IASTED International Conference on Intelligent Systems and Control (ISC), Santa Barbara, California, USA.
- Friedrich H, Münch S, Dillmann R, Bocionek S, Sassin M (1996) Robot programming by demonstration: Supporting the induction by human interaction. *Machine Learning*, pp 163–189.
- Friedrich H, Rogalla O, Dillmann R (1998) Interactive robot programming based on human demonstration and advice. In Christensen H (ed) Dagstuhl Proceedings 1998, Dagstuhl, Germany. Springer.
- Graf R, Weckesser P (1998) Roomservice in a hotel. In 3rd IFAC Symposium on Intelligent Autonomous Vehicles – IAV 98, pp 641–647, Madrid, Spain.
- Kang SB, Ikeuchi K (1997) Toward automatic robot instruction from perception – mapping human grasps to manipulator grasps. *IEEE Transactions on Robotics and Automation*, 13 (1).
- Kroschel K, Mekhaiel MS, Crowley JL, Berard F (1995) Localisation of acoustic sources using microphone arrays. In Proc. Of the annual meeting of the IAR, Grenoble.
- Mitchell TM (1986) Explanation-based generalization – a unifying view. *Machine Learning*, 1:47–80.
- Onda H, Hirukawa H, Tomita F, Suehiro T, Takase K (1997) Assembly motion teaching system using position/force simulator – generating control program. In 10th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '97), pp 7–11, Grenoble, Frankreich.
- Rogalla O, Pohl K, Dillmann R (2000) A general approach for modelling robots. In IEEE/RSJ International Conference on Intelligent Robots and Systems, Takamatsu, Japan.
- Schaal S (1999) Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242.
- Segre AM (1989) Machine learning of assembly plans. Kluwer Academic Publishers.

- Steinhage A, Bergener T (1998) Dynamical systems for the behavioral organization of an anthropomorphic mobile robot. In Pfeifer R, Blumert B, Meyer J, Wilson S (eds) Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (SAB 98). MIT Press.
- Stiefelhagen R, J.Yang, A.Waibel (1999) Modeling people's focus of attention. In IEEE Int. Workshop on Modelling People, Corfu.
- Takahashi T (1993) Time normalization and analysis method in robot programming from human demonstration data. In Proceedings of the IEEE International Conference on Robotics and Automation, pp 695–700, Atlanta, Georgia, USA.
- Tung CP, Kak AC (1994) Integrating sensing, task planning and execution. In IEEE International Conference on Robotics and Automation, vol 3, pp 2030–2037.
- Voyles R, Khosla P (1999) Gesture-based programming: A preliminary demonstration. In Proceedings of the IEEE International Conference on Robotics and Automation, pp 708–713, Detroit, Michigan.
- Yang J, Stiefelhagen R, Meier U, Waibel A (1998) Visual tracking for multimodal human computer interaction. In Proc. Of CHI '98, pp 687–694, Hongkong.
- Zhang J, von Collani Y, Knoll A (1999) Interactive assembly by a two-arm-robot agent. In Robotics and Autonomous Systems. Elsevier.

A New Generation of Compliance Controlled Manipulators with Human Arm Like Properties

Ralf Koeppe, Alin Albu-Schäffer, Carsten Preusche, Günter Schreiber,
and Gerd Hirzinger

German Aerospace Center (DLR), Institute of Robotics and Mechatronics, P.O.Box 1116,
82230 Wessling, Germany

Abstract. This paper describes the overall control structure of a new generation of compliance controlled manipulators implemented by the design and control of the DLR light-weight robot. To achieve the compliance property, the manipulator does not have to revert to a human-like, bionic design. A serial link manipulator, control technology, and sensorized joint actuators can generate a kinesthesia similar to the compliance properties of the human arm. The design and control of the new robot generation implement these properties in a very different way than its biological counterpart to enable high fidelity interaction of the robot with its environment.

1 Introduction

The upper limb is the primary organ that humans use to manipulate objects. It must retain its stability and keep its performance while the physical properties that govern its interaction may vary within a wide range.

In contrast, standard industrial robot manipulators are not capable of controlling their interaction dynamics. They are pure stiff positioning devices which are programmed to execute a task with high speed. With force-torque sensing at the robot wrist and compliant motion control, these robots have been able to physically interact with the environment in a limited way. But the execution speed of a compliant motion task comparable to that of the human has not yet been achieved within a general manipulator setup.

Researchers from various fields attempt to analyze the biomechanics and the underlying principles of movement organization (Winters, Woo, 1990). Especially the study of robot manipulation has lead to an increased understanding of musculo-tendinous actuation principles. Hannaford has built an anthropomorphic robotic arm based on braided pneumatic actuators with the primary goal to investigate human neuro-musculos control principles Hannaford (1995). The braided “artificial muscle” actuator consists of an internal bladder surrounded by a braided mesh that is attached to some “tendon-like” structure. Such an actuator system can adequately mimic the basic properties of a human muscle. Furthermore, it has the same physical size, the same range of forces, stiffness and velocities as the skeletal muscle (Hannaford, Winters, 1990). But increasing design complexity as required for multiple link actuation and low closed-loop bandwidth Caldwell, Medrano-Cerda, Goodwin (1994) make this *bionic human-like design* approach, using current technology, not suitable for a new generation of dexterous manipulators.

Direct-drive robots are known to supply a high bandwidth but at the cost of high weight and large power dissipation. Hence this drive principle cannot be used on mobile or humanoid platforms that aim at a load to weight ratio comparable to the human.

In this paper we contrast the design principle of the human arm (Section 2) to the novel concept of an *advanced robot* (Section 3). The latter implements *human-like dynamic characteristics* and high load to robot weight ratios based on the principle of mono-articulation. Each joint is actuated by a dedicated geared actuation system, which is common in manipulator design. The DLR light-weight robot is an example implementation of such a new generation of advanced robot manipulators. The robot has a weight of 17 kg and can handle 8 kg within the range of human-like dynamics.

2 The Human Arm System

2.1 The Neuro-Musculoskeletal System

The motion of the upper limb is caused by contraction of the agonist and dilatation of the antagonist. Therefore a minimum of two single muscles are needed to actuate motion around one joint. Figure 1 depicts parts of the neuro-muscular control system. Under the assumption of small hand motions around an initial position, as for the case of commanding a joystick with second order linear dynamics, the control system can be represented by a simplified linear single-input single-output model.

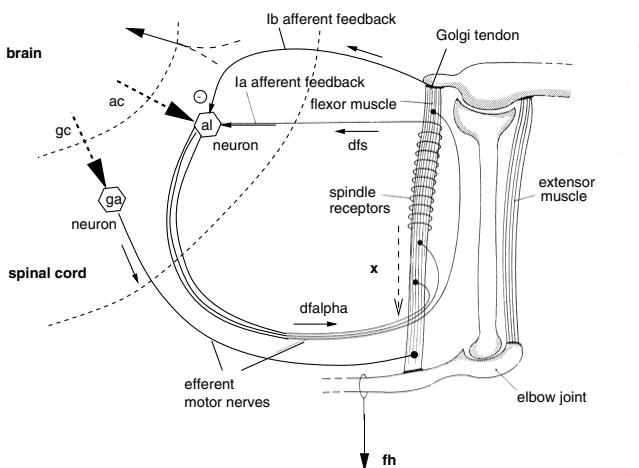


Fig. 1. Simplified model of the upper limb control system

The muscle actuator is represented by the transfer function

$$X(s) = G_{m\alpha}(s)\Delta f_\alpha(s) + G_{mF}(s)F(s). \quad (1)$$

The internal length X of the muscle is a function of the change of efferent nerve excitation Δf_α from the α -neuron and the force F acting on the muscle. The transfer function $G_{m\alpha}$ and G_{mF} depend on the muscle stiffness, the muscle damping, an elastic component connected in serial to the muscle, the combined mass of the limb and interface element, and the spring constant of the interface.

2.2 Impedance Modulation and Passivity

The important advantage of actuation by muscle pairs is the ability to generate a trajectory of *limb position* x and *limb impedance* Z_{op} at the same time, where Z_{op} describes the limb force/velocity relation $F = Z_{op}(V)$. The joint impedance is increased by “tensing” opposing muscles simultaneously, whereas the joint position is changed due to a moment as a result of the difference in muscle tension. The impedance parameters are set to values which allow robust handling of perturbations experienced during the manipulation and robust coupling to an object with arbitrary dynamics.

Colgate and Hogan Colgate, Hogan (1988) investigated the necessary and sufficient condition for stability of a system while interacting with any passive environment. They showed that the system itself has to be of passive nature. The muscular actuator driven by α -efference is certainly an active system since, as any actuator, the muscle can supply energy. In contrast, experiments clearly suggest that the impedance of the arm shows passive behavior Hogan (1989). Hence the driving point impedance of the human arm G_{mF} in (1) appears to be that of a passive system with spring-like behavior, although neural feedback may modify muscle impedance.

The passivity property of the human upper limb implies that the operator model can be designed as a mass, damper, spring system expressed by the parameters m_{op} , b_{op} , and c_{op} , respectively. Hogan Hogan (1989) has presented results from measurements of human arm stiffness and damping. Human subjects were instructed to hold the handle of a manipulandum stationary at a location in workspace. The manipulandum was destabilized at a frequency of approximately 4 Hz and an increasing amplitude up to 5 cm. Stiffness parameters vary within at least two orders of magnitude. A minimum incremental elbow stiffness of 2 Nm/rad and a maximum of 400 Nm/rad were measured. The incremental damping shows variations as well but in a smaller range. From measurements an effective viscosity was estimated to be 5.5 Ns/m. This indicates that the human arm is lightly damped.

Important properties of the human arm can be derived from its multi-joint design. An essential property of the arm in handling contact tasks, is the ability to modulate its internal stiffness in task coordinates. Depending on the task, the arm is made stiff for directions in which the disturbance due to external forces needs to be minimized, for example in case of a disturbance force due to friction in motion direction. The arm is made compliant in directions normal to the contact surface to ensure kinematic constraints.

3 The Advanced Robot: Design and Control Principles

Current advances in mechatronic design enable to build a new generation of advanced, light-weight manipulators with highly integrated sensorized joint drive systems Hirzinger, Albu-Schäffer, Hähnle, Schaefer, Sporer (2001). Each joint is equipped with motor position, joint torque, and joint position sensing. In the sequel, the overall control concept of the DLR light-weight manipulator is devised (Fig. 2 and 3).

Four layers can be identified. The first, top layer represents the complete specification of a consistent skill strategy that is required to execute a compliant motion task. The second layer defines the Cartesian controller consisting of the force controller and the velocity feedforward component. The third layer represents the projection of the Cartesian to the joint space, resulting in the specification of the joint velocity, stiffness and damping. The fourth layer provides the synthesis of the joint controllers and the control action taking into account the physical model of the robot, including the coupling terms of the motion and impedance dynamics. In the sequel, the components of the control concept are discussed.

3.1 Compliant Motion Specification

This skill strategy can be implicitly acquired in an intuitive way from human demonstration Koeppe, Hirzinger (2000) or explicitly synthesized by well known hybrid control methods. The compliant motion is specified by the desired velocity $\dot{x}_d(t)$ and force trajectory $f_d(t)$ describing the external task dynamics. The execution of a compliant motion task with a desired velocity implies a certain force tracking capability of the robot De Schutter, Van Brussel (1988), resulting in a proper choice of the Cartesian internal impedance of the robot's end-effector. The impedance is specified in terms of the Cartesian stiffness and damping matrices, $K_c(t)$ and $D_c(t)$, respectively. The null-space criteria, defining the behavior of the redundant joints, completes the specification for the robot execution.

3.2 Force Control

By closing the force control loop around an inner position/velocity loop, a Cartesian velocity command \dot{x}_c is synthesized from any compliant motion control structure, resulting in two parts

$$\dot{x}_c = \dot{x}_{c,\text{motion}} + \dot{x}_{c,\text{force}}, \quad (2)$$

where $\dot{x}_{c,\text{motion}} = \dot{x}_d$ is the the vector of the desired velocity and $\dot{x}_{c,\text{force}}$ the output of the force controller.

In force control we have developed efficient procedures to automatically synthesize the control parameters based on well known technology and process parameters Cortesão, Koeppe, Nunes, Hirzinger (2000); Natale, Koeppe, Hirzinger (2000).

For the design of the Cartesian force controller a PT_1 behavior with additional dead time of the robot in each degree of freedom (DoF) is assumed:

$$G_{R,i}(s) = P_R \frac{1}{1 + T_R s} e^{-T_{DR}s}, \quad (3)$$

where i denotes the corresponding degree-of-freedom, T_{DR} the dead time, T_R the robot rise time and P_R the resulting proportional gain in the Cartesian space. P_R and T_R are dependent on the internal impedance of the robot, which can be controlled in case of the advanced robot.

The total open loop gain is then

$$P_{total} = P_{fc}(K_c)P_R(K_c), \quad (4)$$

where P_{fc} , the force control gain, is inverse and P_R is directly proportional to the Cartesian stiffness of the robot.

This means that the gain of the force controller can be chosen either in favor of the force controller itself (case 1) or in favor of the joint impedance controller (case 2). The overall bandwidth of the system remains the same as shown in Table 1.

Table 1. Assigning gains to force and impedance controller by choosing a different robot stiffness K_C

Case	K_C	P_R	P_{fc}	P_{total}
1	500 N/m	500	40	20000
2	4000 N/m	4000	5	20000

3.3 Velocity Decomposition

The desired joint velocity is computed using the inverse kinematic relation of a redundant robot arm Sciavicco, Siciliano (1996); Yoshikawa (1990)

$$\dot{\mathbf{q}}_d = \mathbf{J}^{W+}(\mathbf{q})\dot{\mathbf{x}}_c + [\mathbf{I} - \mathbf{J}^{W+}(\mathbf{q})\mathbf{J}(\mathbf{q})] \dot{\mathbf{q}}_a \quad (5)$$

with \mathbf{J}^{W+} the weighted pseudo inverse of the Jacobian \mathbf{J} , and \mathbf{I} the Identity matrix. The inverse kinematics consists of two solutions: $\mathbf{J}^{W+}\dot{\mathbf{x}}_c$ describes the primary kinematic constraint and characterizes the end-effector behavior. $(\mathbf{I} - \mathbf{J}^{W+}\mathbf{J})\dot{\mathbf{q}}_a$ is an additional constraint of secondary priority which can be specified via $\dot{\mathbf{q}}_a$. This term describes the null-space of the manipulator which means that in case of $\dot{\mathbf{x}} = 0$ the manipulator is still capable of performing internal motions. The secondary constraint is typically chosen according to an objective function which is to be maximized. Examples for such an objective function are the manipulability measure, the distance from mechanical joint limits, or the distance from an obstacle.

In practical operation of a manipulator, one has to ensure that joint speed and acceleration limits are not exceeded and that the robot can be controlled despite singularities. For this reason we treat the velocity decomposition as a constraint optimization problem (Schreiber, Ott, Hirzinger, 2001; Schreiber, Otter, Hirzinger, 1999).

3.4 Robot Dynamics

The joint space dynamics of a general redundant robot arm is described by

$$\tau = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})\mathbf{f}_{\text{contact}} \quad (6)$$

where \mathbf{q} is the vector of n joint coordinates, τ is the vector of joint torques, $\mathbf{M}(\mathbf{q})$ is the $n \times n$ inertia matrix, $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ is the vector of centrifugal and Coriolis joint forces, $\mathbf{g}(\mathbf{q})$ is the vector of gravitational forces, and $\mathbf{J}^T(\mathbf{q})\mathbf{f}_{\text{contact}}$ is the vector of joint forces due to external forces applied to the end-effector.

3.5 Torque Decomposition

Another way of expressing the joint torques τ is achieved by the following decomposition

$$\tau = \bar{\mathbf{J}}^T(\mathbf{q})\mathbf{F} + [\mathbf{I} - \mathbf{J}^T(\mathbf{q})\bar{\mathbf{J}}^T(\mathbf{q})] \tau_0, \quad (7)$$

with

$$\bar{\mathbf{J}}^T(\mathbf{q}) = \mathbf{M}^{-1}(\mathbf{q})\mathbf{J}^T(\mathbf{q}) [\mathbf{J}(\mathbf{q})\mathbf{M}^{-1}(\mathbf{q})\mathbf{J}^T(\mathbf{q})]^{-1}, \quad (8)$$

the dynamic consistent generalized inverse Khatib (1990). The joint generalized force τ is decomposed into two dynamically decoupled control vectors, $\bar{\mathbf{J}}^T(\mathbf{q})\mathbf{F}$ the joint forces representing the vector of fictive end-effector force that is required to realize a certain Cartesian motion and force trajectory, and

$$[\mathbf{I} - \mathbf{J}^T(\mathbf{q})\bar{\mathbf{J}}^T(\mathbf{q})]\tau_0$$

the vector of joint forces that only effect the internal motion of the manipulator, i.e., the dynamically consistent null-space of the robot. Analogous to $\dot{\mathbf{q}}_a$ in (5), τ_0 is chosen to satisfy a secondary constraint. The objective function which is to be maximized is the same as in the velocity decomposition case.

Equations (5) and (7) define a force/position duality for redundant manipulators Khatib (1993). To formalize the duality in a consistent way, the weighted pseudo inverse \mathbf{J}^{W+} has to be equal to the dynamic consistent generalized inverse $\bar{\mathbf{J}}^T(\mathbf{q})$, i.e., the weight matrix of the pseudo inverse, in (5), is $\mathbf{W} = \hat{\mathbf{M}}^{-1}(\mathbf{q})$. $\hat{\mathbf{M}}$ is the mass matrix computed from known physical parameters of the robot.

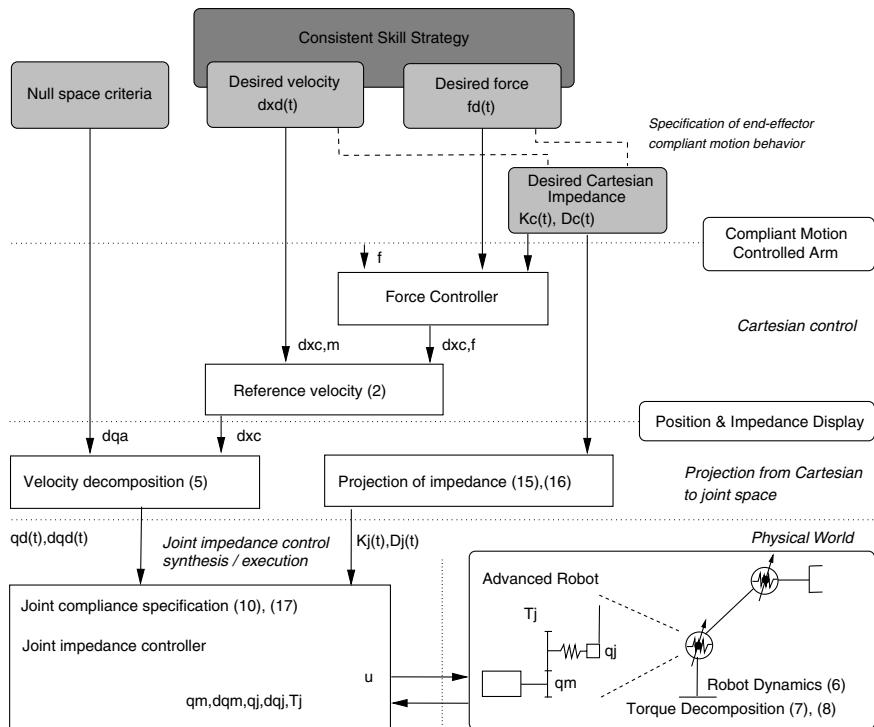


Fig. 2. Advanced robot compliant motion controller – Computation and control scheme

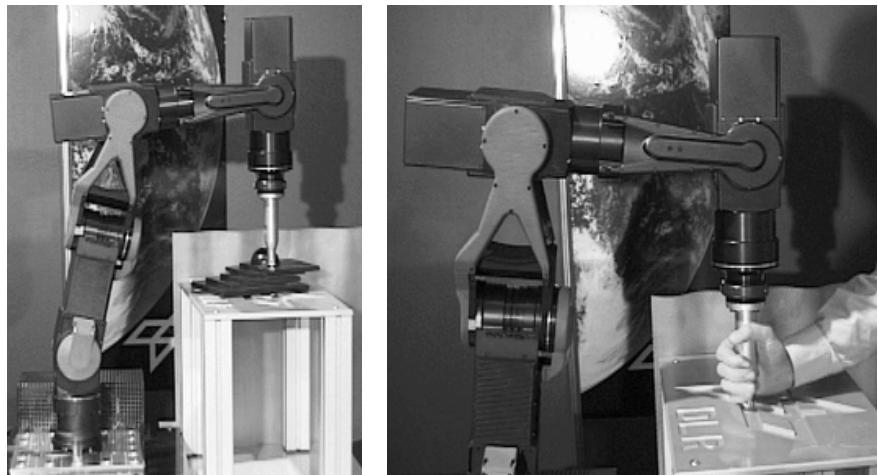


Fig. 3. The DLR light-weight robot as a position and impedance display. High-fidelity interaction with the environment and a human operator

3.6 Joint Control and Design

Industrial robots are controlled to display stiff joint behavior with the goal that the joint trajectory follows the desired one as close as possible. With joint torque sensing the robot can be controlled to display zero stiffness, i.e., a zero torque control mode including gravity compensation that allows to move the robot manually by touching any link.

High stiffness position control and zero stiffness torque control are two complementary concepts. To achieve a compliance characteristics similar to the human-arm a third concept, joint compliance control, has to be considered. This can be achieved by introducing a 4th order elastic joint model Spong (1996).

For the control of the robot, a flexible joint model is assumed. Fast and reliable methods for the identification of the joint model parameters (joint stiffness, damping and friction) were developed, while the rigid body parameters were directly generated from the mechanical CAD programs Albu-Schäffer, Hirzinger (2001). This leads to an accurate simulation of the robot dynamics, so that the controller structures can be developed and tested directly in the simulation.

The first stage in the controller development was a joint state feedback controller with compensation of gravity and friction. The state vector $x = \{q_m, \dot{q}_m, \tau, \dot{\tau}\}$ contains the motor position, the joint torques as well as their derivatives. The conditions for the passivity of this structure were derived in (Albu-Schäffer, Hirzinger, 2000). Under these conditions, the global asymptotic stability of the controller can be proven.

By the appropriate parameterization of the feedback gains, the controller structure can be used to implement position, torque or impedance control. In the last case, the gains of the controller are computed in every Cartesian cycle, based on the desired joint stiffness and damping, as well as taking into account the actual value of the inertia matrix.

The joint torque which produces the Cartesian force f is:

$$\begin{aligned} \tau_c &= J^T(q)f \\ &= \tau - [M(q)\ddot{q} + b(q, \dot{q}) + g(q)], \end{aligned} \quad (9)$$

The aim of the joint impedance controller is then to establish the following relationship between the torque τ_c and the link side position and velocity:

$$\tau_c = K_j(q_d - q) + D_j(\dot{q}_d - \dot{q}), \quad (10)$$

with K_j the joint stiffness matrix and D_j the joint damping matrix.

Since the torque $\tau = \tau_{\text{sensor}}$ can be directly measured by the torque sensor, it is possible to obtain an estimate of τ_c using the model for the rigid body dynamics of the robot:

$$\begin{aligned} \tau_c &\approx \tau_{\text{sensor}} - \tau_d = \\ &\tau_{\text{sensor}} - [\hat{M}(q)\ddot{q} + \hat{b}(q, \dot{q}) + \hat{g}(q)], \end{aligned} \quad (11)$$

with $\hat{M}(\mathbf{q})$, $\hat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}})$, and $\hat{\mathbf{g}}(\mathbf{q})$, the inertia matrix, the vector of centrifugal and Coriolis joint forces, and the vector of gravitational forces, computed from known model parameters of the robot, respectively. According to the flexible joint model, the transfer function $H(s)$ for the joint i from the torque τ_{ci} to the link position has a fourth order denominator $Den_i(s)$:

$$H_i(s) = \frac{\tau_{ci}(s)}{q_i(s)} = \frac{Nom_i(s)}{Den_i(s)} \quad (12)$$

Hence, the best approximation of the first order dynamics (10) by the fourth order system (12) has to be found. A reasonable choice is to define the stiffness of the controlled joint as:

$$K_{jii} = \left. \frac{\tau_{ci}(s)}{q_i(s)} \right|_{s=0} \quad (13)$$

and to design the feedback gains such that the characteristic polynomial in (12) is

$$Den_i(s) = (s^2 + 2\xi_1\omega_1 s + \omega_1^2)(s^2 + 2\xi_2\omega_2 s + \omega_2^2) \quad (14)$$

The desired damping D_{ji} can then be directly mapped to the parameters ξ_1 and ξ_2 . For the estimate of the torque τ_c in (12), the measurement of the link acceleration is required. In many cases this is not feasible, and a common solution is to use the desired acceleration \ddot{q}_d instead. This is possible in the case of free movement. In contact with unknown environments, the real acceleration may substantially differ from the desired one. An alternative solution is to include the term $m_{ii}(q)\ddot{q}_i$ in the transfer function $H(s)$. In this case, $H(s) = H(s, \theta, m_{ii}, K)$ is a function of the physical parameters of the plant θ , of the actual inertia m_{ii} and of the control gain vector K . The values of the parameters ω_1 and ω_2 , and implicitly the control gains K are then chosen to simultaneously satisfy the following criteria:

- To implement the desired joint stiffness (13) and the desired damping.
- To maximize the bandwidth of the joint control, for the given instantaneous values of the inertia m_{ii} .
- To provide, in case of $D_{ji} = 1$ an active vibration damping of the flexible joint structure.

With maximum and zero stiffness the joint compliance control implements pure position and pure torque control in a unified manner. The maximum joint stiffness and the minimum joint damping represent the bandwidth limits for the joint position and joint torque control loop, respectively. A realized maximum joint stiffness of 10000 Nm/rad for stiff position control and a minimal residual Cartesian damping of 15 Ns/m for zero torque control suggest that the range of human-like dynamics is well covered by the DLR light-weight robot system.

3.7 Cartesian Impedance Specification

The desired Cartesian stiffness is specified according to the task geometry and task speed. In directions of motion, the robot is required to display stiff behavior. Consequently the bandwidth of the joints contributing to the motion is required to be high.

In directions of forces, the robot has to display compliant behavior quantifiable as a function of the task execution speed. High task execution speeds require more compliance in force directions to keep a certain limit on the force tracking error. Hence the internal impedance has to be specified according to the desired task dynamics and therefore becomes an integral part of the compliant motion task specification.

3.8 Projection of Impedance

The joint stiffness matrix can be computed from the Cartesian stiffness matrix \mathbf{K}_c , the Jacobian matrix $\mathbf{J}(\mathbf{q})$, and the vector of interaction forces $\mathbf{f} = \mathbf{f}_{\text{contact}}$

$$\mathbf{K}_j = \frac{d\mathbf{J}^T(\mathbf{q})}{d\mathbf{q}} \mathbf{f} + \mathbf{J}^T(\mathbf{q}) \mathbf{K}_c \mathbf{J}(\mathbf{q}). \quad (15)$$

The first term describes the contribution to the joint stiffness due to the variation of the moment arm with the configuration. For zero forces or forces which only lead to small deflections the variation of the Jacobian is small and the first term can be neglected. For larger deflections in case of a compliant system this additional stiffness component has to be considered.

Similar to the derivation of the joint stiffness matrix, the joint damping matrix is computed from the 6×6 diagonal Cartesian damping matrix \mathbf{D}_c to

$$\mathbf{D}_j = \mathbf{J}^T(\mathbf{q}) \mathbf{D}_c \mathbf{J}(\mathbf{q}). \quad (16)$$

3.9 Implementation Issues

The implementation of the above devised control system entails some important consequences. The joint stiffness and joint damping matrix are dependent on the robot configuration. In addition, the joint stiffness matrix varies for large deflections proportional to the external force. Consequently, Equations (15) and (16) have to be computed on-line at a certain minimum rate depending on the task dynamics.

Note that the matrices \mathbf{K}_j and \mathbf{D}_j have diagonal and off-diagonal terms. The diagonal terms specify the desired joint impedance and the off-diagonal terms represent stiffness and damping coupling terms. In the current, decentralized joint controller structure, only the diagonal terms of the stiffness and damping matrices can be implemented using the joint impedance controller. The torques produced by the off-diagonal elements are computed on the central computer in a joint level task (at the bus sampling rate of 1ms) and are added to the desired torques. The desired joint torque then becomes

$$\begin{aligned} \tau_d = & \left[\hat{\mathbf{M}}(\mathbf{q}) \ddot{\mathbf{q}} + \hat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}) + \hat{\mathbf{g}}(\mathbf{q}) \right] \\ & + \mathbf{K}'_j(\mathbf{q}_d - \mathbf{q}) + \mathbf{D}'_j(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}), \end{aligned} \quad (17)$$

with \mathbf{K}'_j and \mathbf{D}'_j containing only the off-diagonal terms of \mathbf{K}_j and \mathbf{D}_j .

As we explain in the following section, the joint impedance control loop including the coupling terms have to be implemented as a high bandwidth control loop.

Hence, the concept of joint control as a pure decentralized control scheme has to be extended taking into account the non-diagonal coupling terms which results in an increase of communication bandwidth between the centralized controller and the joint controller.

4 Conclusion

In this paper we have presented design and control principles of a new generation of advanced light-weight manipulators that can achieve human-like dynamic with a none bionic design.

The control concept introduced in this paper, requires special attention to guarantee properties like system stability and passivity. Furthermore, the robot pose has to be properly chosen to achieve a well conditioned internal impedance range. The presentation of these considerations is beyond the scope of this paper. The important lesson learned, is that robot coordination, commonly considered as a pure movement problem, has to be extended to the modulation of internal multi-joint impedance when compliant motion tasks are performed. With mechatronic technologies as developed for the DLR light-weight robot, human-like dynamic behavior can be achieved. The dynamic and power-to-weight ratio similarity of an advanced robot and a human arm have to be assessed and quantified as a measure of performance.

References

- Albu-Schäffer A, Hirzinger G (2000) State feedback controller for flexible joint robots: A globally stable approach implemented on the DLR-robots. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp 1087–1093.
- Albu-Schäffer A, Hirzinger G (2001) Parameter identification and passivity based joint control for a 7DOF torque controlled light weight robot. In Proc. of the IEEE Int. Conf. on Robotics and Automation, pp 2852–2858.
- Caldwell D, Medrano-Cerda G, Goodwin M (1994) Characteristics and adaptive control of pneumatic muscle actuators for a robotic elbow. In Proc. of the IEEE Int. Conf. on Robotics and Automation, pp 3558–3563.
- Colgate J, Hogan N (1988) Robust control of dynamically interacting systems. Int. J. Control., 48(1):65–88.
- Cortesão R, Koeppe R, Nunes U, Hirzinger G (2000) Explicit force control for manipulators with active observers. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS, pp 1075–1080.
- De Schutter J, Van Brussel H (1988) Compliant robot motion, ii. a control approach based on external control loops. The International Journal of Robotics Research, 7(4):18–33.
- Hannaford B (1995) The anthropomorphic arm: a system for the study of spinal circuits. Annals of Biomedical Engineering: L.Stark Special Issue.
- Hannaford B, Winters J (1990) Actuator properties and movement control: Biological and technological models. In Multiple Muscle Systems, Biomechanics and Movement Organization. Springer-Verlag.

- Hirzinger G, Albu-Schäffer A, Hähnle M, Schaefer I, Sporer N (2001) A new generation of torque controlled light-weight robots. In Proc. of the IEEE Int. Conf. on Robotics and Automation.
- Hogan N (1989) Controlling impedance at the man/machine interface. In Proc. of the IEEE Int. Conf. on Robotics and Automation.
- Khatib O (1990) Reduced effective inertia in macro/mini-manipulator systems. In Robotics Research: The Fifth International Symposium, pp 279–284. Cambridge: MIT Press.
- Khatib O (1993) Lecture notes in advanced robotic manipulation. Stanford University.
- Koeppe R, Hirzinger G (2000) Sensorimotor skill transfer of compliant motion. In Hollerbach J, Koditschek D (eds) Robotics Research: The Ninth International Symposium, pp 239–246. Springer-Verlag.
- Natale C, Koeppe R, Hirzinger G (2000) A systematic design procedure of force controllers for industrial robots. *IEEE/ASME Transactions on Mechatronics*, 5(2):122–131.
- Schreiber G, Ott C, Hirzinger G (2001) Interactive redundant robotics: Control of the inverted pendulum with nullspace motion. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS.
- Schreiber G, Otter M, Hirzinger G (1999) Solving the singularity problem of non-redundant manipulator by constraint optimization. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS, Kyongju, Korea, pp 1482–1488.
- Sciavicco L, Siciliano B (1996) Modeling and control of robot manipulators. The McGraw-Hill Companies, Inc.
- Spong M (1996) Motion control of robot manipulators. In Handbook of Control, pp 1339–1350. CRC Press, W. Levine, editor.
- Winters J, Woo SY (eds) (1990) Multiple muscle systems, biomechanics and movement organization. Springer-Verlag.
- Yoshikawa T (1990) Foundations of robotics: analysis and control. MIT Press.

Part 5

Human Centred Robotics

Session Summary

James Trevelyan

Chair

Department of Mechanical & Materials Engineering, University of Western Australia

Visitors to Australia often like to ask local people about food that is truly Australian. It is a hard question to answer: people from over 160 countries now live in Australia and most of them have brought their culinary tastes with them. However, there is one particular delicacy enjoyed by aboriginal Australians: the wicheety grub. This creature is a white caterpillar about 6 to 10 cm long often found in dead and decaying wood if you know where to look for them. It bears a remarkably accurate resemblance to the creature depicted in figure 1 of Allison Okamura's paper entitled Uniting Haptic Exploration and Display. In fact one can only truly appreciate the wicheety grub with both the sense of taste and touch, bringing us to the world of haptics which two papers in this session explore.

Allison Okamura's paper describes a series of experiments in which a small robot equipped with a touch sensing finger is used to explore features on a smooth surface. The paper is particularly valuable because of an extensive reference list: there are many references and comparisons to earlier work which make this a particularly useful paper for researchers in this field. She also takes us through a formal definition of surface features and methods that can be used to explore them. She then goes on to explain how the shape of the feature can be extracted from feature points discovered from exploration. Finally she describes how the same small manipulator can be converted into a haptic rendering and display device to display the feature.

Features on the surface that can be explored with this device are described in terms of differential geometry. Given the radius of curvature of the finger, and the path traced by the finger as it traverses the surface, one can compute differential geometric properties of the surface. This process is limited: if the surface has the feature with a negative radius of curvature less than that of the finger tip there will be small unobservable regions. These aspects are clearly explained in the paper.

The author points out that there is much more to do in this field. While she has demonstrated that it is possible to generate realistic haptic models automatically she points out that this work is really only a starting point. Haptics has many advantages in situations where other kinds of sensing are not feasible, for example, in underwater situations where vision is impractical due to turbidity. She suggests that rolling fingertips need to be developed to reduce friction problems where sliding is impractical.

Oussama Khatib presents an elegant paper that brings robot control, task planning, posture control, corporative manipulation and obstacle avoidance together to provide a computationally efficient method for interacting with a virtual world full of complex objects using a Phantom haptic manipulator. The paper is elegant because it shows how other methods developed by the author over several years can be neatly combined to solve this problem. The paper can only be properly appreciated

if what has the chance to see the author's video which he showed at the conference meeting.

The author explains that the key problem to solve is to make the computation feasible for typical computers available today. He does this for systems with up to 500 degrees of freedom: the systems can interact and collide with each other, have articulated joints, and users can feel all forces as if they were actually present in the artificial world.

The final paper of the session takes us to a different subject: how to bring together the cognitive strengths of human beings and autonomous robots. The aim is to solve the typical problems that make the robots unable to operate without occasional help from the human masters. The authors introduce a concept that they label as "collaborative control". They define this in terms of a human and a robot working as partners to achieve common goals. Whenever the robot needs assistance it sends a message to the human based on its knowledge of the human's expertise. The robot knows that a novice human will not be able to reliably answer all the questions it might ask. Therefore, the robot treats the human has just another source of information alongside its many senses. It weights the information from the human appropriately. If the human does not answer the question, either because he or she is too busy or because the question is too difficult, the robot will give up after a while and do its best to solve the problem that it confronts without the additional information.

These ideas are explored in a hypothetical situation: a rock finding robot exploring a planet, looking for green rocks. The vision system of the robot cannot distinguish different kinds of rocks but it can find green things. Therefore, when it finds a green thing it asks the human operator "is this a green rock?".

The authors have implemented this system using a hand-held organizer for the human interface: they address the problems of limited screen resolution and the need for simple menu driven responses.

During the discussion following the paper presentation many possibilities were explored. Like the other papers, this one created a great deal of interest because it offers an effective work-around for many of the problems facing today's generation of autonomous vehicles. A member of the audience pointed out that many control systems used for process plant automation interact with plant operators using computer-generated displays and text messages. It is quite possible that some of the techniques could provide a valuable source of ideas for the robotics community.

The author opened the presentation by asking the audience "what kind of conversation would a robot like to have with human?" He suggested that perhaps the robot would like to ask for help every now and again, particularly when confronted with difficult perceptual problems. "What you think this is? Is it a real cat? Is it safe to drive over it?"

The unifying theme of this session is the interactions that are possible between robots and human beings: a theme that was revisited in many other papers in this conference. Robotics science is coming to terms with the need for human beings and robots to interact usefully and safely. While human-computer interaction (HCI)

has been carefully studied for many years, it is clear that the field of human-robot interaction (HRI – a new acronym?) is wide open and is only just beginning to be explored. These three papers provide a valuable starting point for anyone interested in further work in this area.

Uniting Haptic Exploration and Display

Allison M. Okamura

The Johns Hopkins University, Baltimore MD 21218, USA

Abstract. This work develops a methodology for building haptic reality-based modeling systems by exploiting the complementary goals of haptic exploration and display. While the generally unstructured nature of haptic exploration makes it difficult to develop and control autonomous robotic fingers, simultaneous consideration of exploration and display provides a way to characterize the required sensing, control, finger geometry, and haptic modeling components. Both the exploring robot and haptic display must complement an appropriate virtual model, which can accurately recreate remote or hazardous environments. We present an example of haptic exploration, modeling and display for surface features explored using a three-degree-of-freedom spherical robotic fingertip with a tactile sensor.

1 Introduction

Haptic exploration is a mechanism by which humans learn about the surface properties of unknown objects. Through the sense of touch, we are able to learn about attributes such as object shape, surface texture, inertia, and stiffness. Robots can also use cutaneous and kinesthetic sensing to learn about unknown environments. Unlike vision or audition, robotic haptic exploration involves direct interaction with the object being explored, presenting significant challenges in both control and sensing. However, recent advances in tactile sensing and dexterous manipulation have made it possible for multi-fingered robotic hands to better manipulate and therefore explore unknown objects.

Haptic interfaces are also related to the human ability of haptic exploration. These devices allow users to feel virtual environments for visualization, design, training, and entertainment. Ideally, these haptic virtual environments feel identical to real environments, although limitations in hardware and modeling prevent them from feeling perfectly realistic. Reality-based modeling attempts to generate realistic virtual environments based on measurements of actual objects.

By combining these two areas of haptics, exploration and display, we hope to gain insight into the following questions:

- How can robots perform autonomous haptic exploration? What goals should define exploratory procedures and how are they translated into control systems, path planning, and system design?
- How can virtual environments be made to feel as realistic as possible? What kind of data should be obtained and how should it be modeled and rendered?

This research builds on the author's previous work in dexterous manipulation and haptic exploration of surface features Okamura, Cutkosky (2001a,b). In that

work, the exploration problem was restricted to a specific goal to enable straightforward motion planning, tactile data interpretation, and finger control. The challenge in this paper is the development of a “design science” for building autonomous haptic reality-based modeling systems, which require a combination of haptic exploration, computation, and display. A logical system design (as shown in Fig. 1) takes all three steps into account. While we present the specific goal of feature exploration as an experimental example, the method can be applied to any type of haptic information.

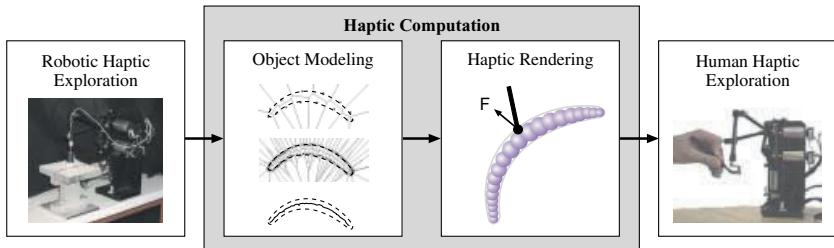


Fig. 1. The connection between robotic exploration of real environments and human exploration of haptic virtual environments is provided through modeling and haptic rendering.

1.1 Previous Work

A number of investigators have addressed the problem of using robotic fingers in exploratory procedures. Examples include Allen, Michelman (1990); Stansfield (1988), which focus on the sensing of global object shapes and fitting shapes to object models. The integration of tactile sensing and dexterous manipulation with rolling or sliding (Li, Qin, Jiang, Han, 1998; Maekawa, Tanie, Komoriya, 1995) and tracking of surfaces and edges Bay, Hemami (1990); Hemami, Bay, Goddard (1988) have also been accomplished in recent work. Overviews of dexterous manipulation and tactile sensing are provided in (Nicholls, Lee, 1989; Okamura, Smaby, Cutkosky, 2000).

Reality-based modeling for display in virtual environments has recently become possible using visual, aural, and tactile systems. The most advanced work in this area has been performed by Pai, *et al.*, who developed an active measurement facility Pai, Lang, Lloyd, Woodham (2000); Richmond, Pai (2000) for obtaining multi-modal data. In 3D vision, researchers have developed surface and volumetric methods for digitizing the external shape of physical objects Curless, Levoy (1996); Lim, Turkiyyah, Ganter, Storti (1995). Friction and texture have recently been modeled using several different techniques Hayward, Armstrong (2000); Richard, Cutkosky, MacLean (1999). Others (Debus, Dupont, Howe, 2000) have created environment models by analyzing motions and constraints during teleoperation. In the medical domain, models of tissue and organs have been populated with actual data and displayed on a haptic interface (d’Aulignac, Balaniuk, Laugier, 2000). While a few

researchers have examined the modeling of haptic information for haptic display, no general method has been developed to create autonomous haptic exploratory procedures for this goal.

2 The Correspondence Between Exploration and Display

In haptic exploration, it is generally desired to capture as much information about the object as possible. Yet, the goal of haptic display provides natural limitations on the type of data that is useful. There is no need to take data that cannot be used to populate the virtual model, and the virtual model is limited by the type of haptic display. Table 1 shows the major components of haptic exploration and display, and reveals a correspondence between these systems.

Autonomous exploration is inherently difficult because the problem is not well defined. Haptic exploration is particularly difficult because it is intrinsically an active activity; unless the object to be examined is massive or fixtured, exploration is coupled with manipulation. No general approach has been developed for the design of haptic exploration systems. In this section, we examine how the synergy between haptic exploration and display can be used in bilateral system design.

Table 1. Corresponding components of haptic exploration and display systems

Haptic Exploration (Robot)	Haptic Display (Interface)
Sensors (force, tactile, acceleration)	Output (cutaneous, kinesthetic, vibration)
“Fingertip” Geometry	Physical/Virtual Tool Shape
Degrees of Freedom	Degrees of Freedom
Control	Control
Path Planning/Exploration AI	Rendering/Human Exploration Method

2.1 System Components

We will now examine the reality-based modeling system components in detail. For each component, we describe how limitations in haptic display place bounds on the type and amount of information that must be collected by the robot.

Sensing: The exploring robot will need to sense the environment for both exploration and data acquisition. A key point is that not all sensors are necessarily used for both purposes. Data types include position, force, tactile pressure (intrinsic or extrinsic), acceleration, etc. The robot’s sensing ability will depend on both the type and accuracy of the target haptic virtual environment/display. For example, if the haptic interface has a cutaneous display, the robot must record local shape information either through extrinsic tactile sensors or moving point contact. Limitations

in the haptic display, such as spatial resolution, can be applied to robot sensing requirements. Temporal resolution is also relevant in the sensing and display of high frequency vibrations resulting from impact (Okamura, Dennerlein, Howe, 1998).

Geometry: The geometry of a robot finger is influenced directly by the sensing system (some sensors are only available in certain geometries) and the desired components of the virtual model. Given a resolved-force haptic device, the virtual model may include object properties such as shape, friction, and (resolved-force) texture. The selection of appropriate finger geometry will affect the workspace and enable or enhance detection of certain object properties. One can also consider the geometry of the haptic interface, especially when the interaction is tool-based. For some applications, such as surgical modeling, the robot's sensing tool should record forces through an appropriate geometry, e.g., forceps or scissors.

Degrees of Freedom: The necessary sensing and actuation can also be limited in degrees of freedom (DOF). For example, it is often sufficient to represent objects with 3 DOF of force feedback, although the robot may require additional degrees of freedom in order to collect the data.¹ The correspondence in degrees of freedom will depend on the application and desired degree of realism.

Control: The low-level control method is determined by many factors, including sensor availability, desired accuracy in specific parameters, and robot/haptic device design. Two primary control types exist for both the robot and the haptic device: admittance and impedance control (Hogan, 1985). While the robot and haptic device do not need to use the same control system, the virtual model may be more easily designed and rendered if similar control laws are used in data acquisition and display. Otherwise, the system may require unnatural mappings between the robot control and rendering techniques.

Path Planning: Path planning for haptic exploration requires a high-level “intelligence” that is based on a continuously updated object model. The virtual object model, which is influenced by the haptic display, requires the robot to take appropriate data. For example, if the model needs to be populated using points on the boundary of a feature, then the robot finger path must be constrained to outlining the feature. When data are required for friction modeling, the robot finger will need to press on the object with a specified force while sliding over the surface. In addition, observations of how humans explore can be used to create robot exploratory procedures. Human hand motions have been mapped to object properties such as texture, hardness, temperature, weight, and shape (Klatzky, Lederman, 1990). The way that a human is likely to explore the virtual environment can determine the necessary model, which in turn specifies the robotic exploratory procedure.

2.2 Selecting a Basis for System Design

The reality-based modeling system can be designed based on components of the robot, virtual model, haptic display, or a combination of the three. Which should be

¹ In this case, the virtual environment will be limited in realism; the lack of torque display can create a “phantom” effect where the user can physically pass through the volume occupied by a virtual object, yet the operational point is constrained (Massie, Salisbury, 1994).

used as the basis for designing a system from scratch? We argue that the desired virtual model should be the design base, as that will most directly specify the requirements of the robot and haptic display. This model can be created to meet the needs of a particular application, such as remote planetary geology, undersea artifact exploration, or surgical simulation.

Often, both the robot and haptic interface exist from the onset, denying us the luxury of designing an ideal complement to an existing robot or interface. Thus, a common challenge is designing virtual models that can (1) be rendered for given haptic interface and (2) be efficiently populated with data from the exploring robot. Robot exploration and haptic rendering methods should be chosen carefully to create a cohesive reality-based modeling system, allowing natural mappings between the system components.

3 Exploration, Modeling, and Display of Surface Features

The section provides a detailed example of a reality-based modeling system. While the original system was created only for haptic exploration and modeling purposes, the integration of rendering and haptic display generated a complete reality-based modeling system.

3.1 Experimental System Design

This work concentrates on one particular exploratory procedure: the detection and identification of fine surface features. Features are a useful way to organize the information obtained while exploring an object. A number of investigators including Ellis (Ellis, 1984) and Stansfield (Stansfield, 1988) consider features as intrinsically relevant for robotic haptic sensing. Features can also be used to segment object data for registration procedures in pattern matching applications.

In designing the reality-based modeling system, a primary concern was to obtain a feature definition (which partially specifies how it is modeled) that was scalable and general. Knowing that we wanted a robot that could explore in 3D (so we could detect and model features on a curved surface), we chose the 3GM from Immersion Corporation, which was originally designed as a haptic interface. This selection had a distinct advantage in that the same robot could be used for haptic exploration and display. It eliminated the need for complicated mappings between the robot, model, and haptic device.

It was determined that an intrinsic tactile sensor was necessary for robot control, in order to maintain an appropriate contact force direction during exploration. As will be shown, tactile sensor data were not necessary for feature detection. Once a feature detection scheme was implemented, simple path planning was added for active exploration of features. We then formulated a method for modeling of “snake-like” ridge features using shape skeletons. Finally, this model was haptically rendered and displayed on the 3GM haptic interface.

3.2 Feature Definition and Detection

We now present a feature definition and detection method that is particularly suited for haptic exploration of objects with small surface features such as bumps, ridges, and grooves using a round fingertip. A more detailed description is available in Okamura, Cutkosky (2001a,b).

This work takes a differential geometry-based approach to surface feature definition, using principal curvatures and their associated principal directions. Given a smoothly curving surface (at least C^2 continuous), one can define the *first and second principal curvatures*, and their corresponding *first and second principal directions*. These quantities are calculated from the *Second Fundamental Form*. Montana (Montana, 1988) defines a *Gauss map* for a surface patch and a normalized *Gauss frame* at a point on the surface. Consider the axes of the Gauss frame to be an orthogonal frame $(x, y, z) = (e_1, e_2, e_3)$. The Second Fundamental Form is a matrix of partial derivatives, where the elements ω_{ij} are computed as the dot product of e_j and the first derivative of the gradient in the e_i direction. It is possible to extract the principal curvatures from the Second Fundamental Form by creating a diagonalized matrix whose elements k_1 and k_2 are the principal curvatures where $|k_1| \geq |k_2|$ Interrante (1997).

$$\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} = [\mathbf{p}_1 \mathbf{p}_2]^{-1} \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \end{bmatrix} [\mathbf{p}_1 \mathbf{p}_2] \quad (1)$$

The principal directions are the eigenvectors \mathbf{p}_1 and \mathbf{p}_2 .

Now consider a spherical robot finger rolling and sliding over the surface of an object with point contact at all times. The center point of the finger generates an *offset surface* Rossignac, Requicha (1986). The concept of a *parallel surface* can be used to develop a description of the offset surface traced by the fingertip's center of curvature. A parallel surface is constructed by drawing a normal line (determined by the z-axis of the Gauss frame, $\mathbf{z}(u)$) with length r_f at each point on the surface. The locus of the end points of these normal lines is the parallel surface. $s_p(u) \in S_p$ maps U to \mathcal{R}^3 :

$$s_p(u) = f(u) + r_f \mathbf{z}(u), \quad (2)$$

where $\mathbf{z}(u)$ is the unit vector in the z -direction of the Gauss frame, r_f is the radius of the fingertip, and u parameterizes the surface. On the parallel surface, discontinuities called *caustic points* (Koenderink, 1990) occur when the curvature of the object is less than $-\frac{1}{r_f}$. Caustic points are associated with features having high local curvature. The *traced surface* is defined as the envelope of spheres with centers on the original surface. As the finger moves along a traced curve, the curve will have discontinuities at *interference points*, where the parallel curve intersects itself, surrounding the caustic points. These interference points are determined analytically by finding the location where the traced curve intersects itself.

Using tactile sensor data and a model of the robot finger, one can compute a trajectory of the finger/object contact point in space and build a model of the surface. This method works for any fingertip shape, as long as the contact point can

accurately be sensed. This *estimated surface* can also be built by taking the inverse parallel surface of the traced surface. These surfaces and associated points of interest are shown in Fig. 2.

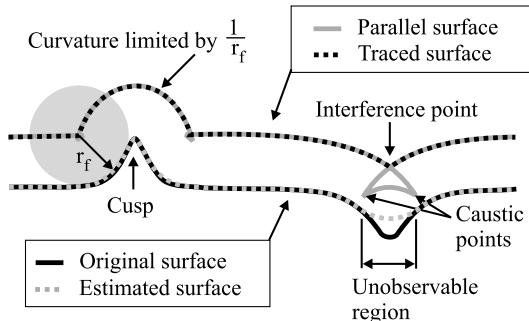


Fig. 2. 2-D slice of a spherical fingertip with original, parallel, traced, and estimated surfaces. Interference and caustic points are identified.

The concept of a feature in the context of haptic exploration with robotic fingers is not only dependent on the surface of the object, but also on the size and shape of the finger. (In this work, we assume a spherical fingertip.) Suppose that an object can be locally fit with a surface S with principal curvatures $k_1(S)$ and $k_2(S)$. A *curvature feature*, as detected by a spherical robotic fingertip with radius r_f tracing over a surface S with defined outward normal and curvature, is a compact region where at least one of the principal curvatures satisfies $k_i > \frac{1}{r_f}$ or $k_i < -\frac{1}{r_f}$. These are *positive curvature* (convex) and *negative curvature* (concave) features, respectively. A *macro feature* is a compact region on surface S containing one or more curvature features. A macro feature is formed from multiple curvature features if each curvature feature is near another curvature feature within a distance r_f .

Using the basic curvature feature definition, an infinite number of macro features may be defined. While the minimum number of curvature features required for a macro feature is one (creating a concave or convex cusp), additional curvature features can be used to define more complicated macro features (Fig. 3). Features are detected by measuring the curvature of either the estimated surface or the traced surface and comparing it to the curvature of the fingertip. When the traced surface is used, the position of the center of the fingertip, not the tactile sensor, provides the feature data. As shown in Fig. 2, the interference point in the traced surface creates an easily detected discontinuity in the traced surface.

3.3 Feature-Guided Haptic Exploration

We will consider the situation where a robotic finger has sufficient degrees of freedom to actively explore a surface in three dimensions. The robotic finger used was the 3GM haptic interface from Immersion Corporation; the software for controlling



Fig. 3. Two possible macro features. On the right, in a plan view of the surface, the gray regions indicate curvature features, with the necessary principal curvatures labeled. + indicates a positive curvature feature, - indicates a negative curvature feature, and ϵ is no curvature feature ($|k_i| < \frac{1}{r_f}$), where k_1 and k_2 are the principal curvatures of the surface and r_f is the radius of the fingertip. It is assumed that $|k_1| \geq |k_2|$. The white regions between the curvature feature regions have a maximum width of r_f .

it was easily modified to make it into a robotic finger. Because the 3GM is backdrivable and lightweight, open-loop force control and gravity compensation can be used (Goldenberg, Wies, Martin, Hassler, 1998). The tactile sensor used was the Optical Waveguide Tactile Sensor from Maekawa, *et al.* Maekawa, Tanie, Komoriya (1993). Because this PSD-based sensor is analog and fast, data can be obtained at the 1kHz rate used in finger control. We used a hybrid control law with impedance control parallel to the surface and a specified normal force perpendicular to the surface. Figure 1 shows this robotic finger and tactile sensor exploring a flat surface with a ridge feature.

This exploratory procedure is termed “feature-guided” exploration because the discovery of a feature leads to changes in the fingertip path. Features may be detected when a finger’s path simply moves the contact point over the feature, however, complete feature type and shape identification may require further maneuvering.

The first step in a feature encounter is identification of the feature type, which can be detected from a pattern of curvature features. Next, the finger traces the boundary of the feature to obtain appropriate data to populate the model. Figure 4 shows the contact point data from a finger that has partially explored a feature and is beginning the tracing step. While performing this tracing, it is possible for the finger to lose contact with the feature. At intervals, the finger can re-approach the feature and find the curvature feature regions again. When the finger moves away from the feature in order to relocate a curvature feature, the extra contact data should be excluded from the set used for feature modeling. In this work, non-boundary data taken during a tracing phase were removed manually. However, this removal can be performed automatically by noting the increase in impedance force as the finger travels over a feature and allowing time for overshoot before the finger returns to tracing the feature.

3.4 Modeling

For each macro feature, a shape skeleton, or 2D medial axis Blum (1967), may be created in a coordinate system fixed on the surface of the object. This is called the *feature skeleton*. Shape skeletons are geometric abstractions of curves, surfaces, or solids that are useful as lower-dimensional representations. The shape skeleton is

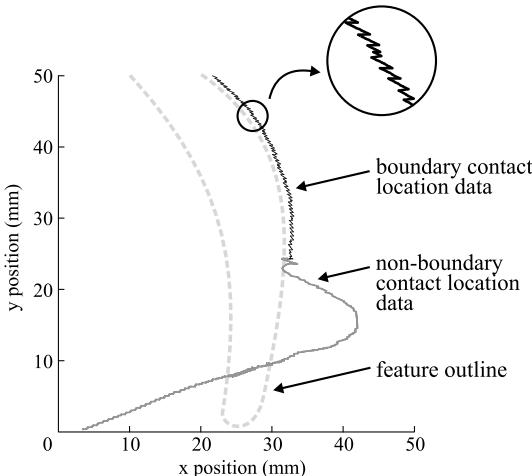


Fig. 4. Actual contact location data obtained as the finger approached the feature, traveled over it (overshooting in order to complete feature detection), and began tracing around the boundary, with a velocity command of 10 mm/s. Data that are not part of the feature boundary is excluded from the feature modeling process.

chosen to represent features to create a structured model that can be easily rendered and used in pattern matching.

The skeletonization method used to extract the medial axis transform of feature data is taken from medial axis algorithms for 2D image data (Brandt, Algazi, 1992). Traditionally, the boundaries are extracted from the image by edge detection and skeletons are generated directly from the boundary data. Much of the overhead of the boundary approach is in the detection of a boundary of the image and removal of excessive small branches due to noisy data. However, our data already contains only feature boundary information because the data were obtained by tracing the feature boundary. Because the portion of local exploration, which is identified as a “tracing” phase, is well defined, the data taken during that phase can be separated from the rest of the tactile or position data. This creates a clean group of boundary samples.

Brandt and Algazi (Brandt, Algazi, 1992) and Ogniewicz (Ogniewicz, 1994) have used approaches in which boundary points are used to find Voronoi diagrams in order to generate a discrete medial axis. The algorithm begins with the creation of a Voronoi diagram from the boundary data points. The Voronoi diagram is the partitioning of a plane with n points into n convex polygons. Each polygon of the Voronoi diagram (also known as a Voronoi polygon) contains exactly one data point and every point in a polygon is closer to this data point than to any other data point. Ogniewicz calls the Voronoi diagram of a discrete set of boundary points the *discrete Voronoi medial axis (DVMA)*. It can be shown formally that the DVMA approaches the continuous Voronoi diagram as the number of boundary samples increases (Fig. 5).

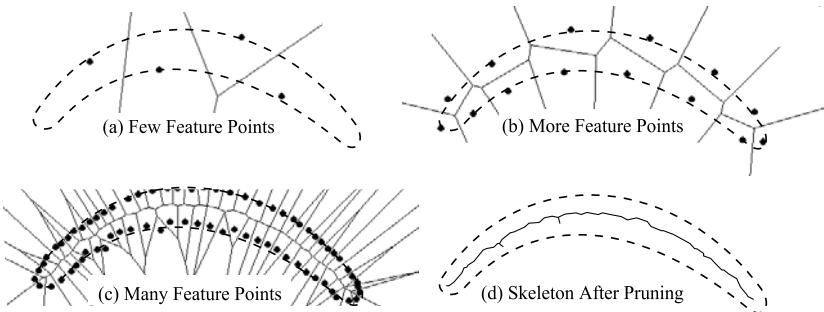


Fig. 5. As the number of boundary samples on a feature increases, the discrete Voronoi medial axis (DVMA) approaches the continuous Voronoi diagram. The DVMA is then pruned to get a more accurate object skeleton. In this example, the boundary samples are noisy and cause unwanted arcs in the final skeleton. These can be removed using a length threshold or connectivity rules.

Once the DVMA has been found, the skeleton must be extracted by pruning away undesirable Voronoi polygon edges. Brandt Brandt, Algazi (1992) presents a method where polygon segments are deleted based on the absolute regeneration error and an empirically-determined threshold. For feature detection based on tactile and position data, however, there is a simpler pruning algorithm based on the length of the Voronoi polygon edges. Given the velocity of the robotic fingertip performing the exploration and the rate at which data are sampled, there will be a minimum distance between samples at the boundary of the feature. The results of pruning using this method are shown in Fig. 5. One can see that where there are data points that are particularly noisy and far away from the main feature shape, unwanted skeleton arcs appear. These could also be pruned away, provided that the features under consideration do not branch. Pruning is accomplished by removing the skeleton arcs that are smaller than a threshold length, defined using the noise present in the data. Some researchers (Patrikalakis, Gursoy, 1990) use the medial axis itself to determine if an object “branches,” so the threshold choice depends on the application.

Because the features will be rendered in a haptic virtual environment, we store both the points on the feature skeleton and the radius of the locally maximal ball at each point. While some skeletonization algorithms retain this information automatically, the DVMA approach does not. Thus, at each skeleton point, we post-process the data to determine the distance from each skeleton point to the nearest boundary data point. While there are certainly limitations to this approach (e.g., the height of the feature is assumed to be the same as its width), this successfully captures the general shape and width of the feature. For some applications, where the height of the feature (orthogonal to the surface) is necessary, additional exploratory procedures that trace along the highest point on the feature can be implemented.

3.5 Haptic Rendering and Display

The final step in the reality-based modeling process is to render the model and display it on a haptic interface. As mentioned, the 3GM was used not only as the robot, but also as the haptic interface. This resolved-force, stylus-based device is appropriate because the model obtained was a shape description to be rendered using a “stiffness” surface model, where forces felt by the user are proportional to and in the opposite direction of penetration into the surface. Surface stiffness data were not obtained by the robot, thus the model uses the maximum stiffness that can be stably displayed by the haptic interface. (This is an example of influence of the haptic device on the model.)

The model contains several types of information: global/nominal shape, skeleton points in the coordinate frame of the nominal surface, their corresponding radii, and the surface stiffness. Considering only singly-connected skeletons (with no branches), the skeleton points are arranged in order of connectivity. In our exploratory procedure, the nominal surface was assumed to be a plane, however, additional exploratory procedures could be performed to determine the “global” surface shape.

In our haptic rendering algorithm, which is described in more detail in (Corso, Chhugani, Okamura, 2002), objects are defined by a discretized Medial Axis Transform. The skeleton and physical properties of the object, including the radii of the spheres centered on the skeleton and material properties, are encapsulated in a single high dimensional parametric surface (e.g., NURBS) that fits the data points. Intersection computations are performed by (1) using a binary search over the knot value range to determine the closest point on the skeleton to the haptic device position, (2) obtaining the radius of the MAT disc at that point, and (3) comparing the haptic device position to that radius. To compute the force upon collision, we use a linear spring model that takes into account both normal and shear surface forces. (Deformable surfaces can also be rendered by attaching springs and dampers to the control points of the surface splines.) The implementation attains real time 3D haptic and graphic rendering rates making it appropriate to model deformation in complex haptic virtual environments. This method is appealing because it takes advantage of single point haptic interaction to render efficiently while maintaining a very low memory footprint. Future work on haptic rendering using the MAT will address bifurcating surfaces, analysis of area/volume preservation, and methods to increase computational efficiency.

4 Conclusions and Future Work

This work investigated techniques that allow robots to acquire and interpret tactile information in order to automatically produce realistic virtual environments. By uniting the goals of haptic exploration and display, we obtain a better understanding of how exploratory procedures can be described and applied using various control laws, and how to model and interpret tactile information. Improvements in haptic exploration bring us closer to robotic systems that can use the sense of

touch to accomplish tasks in unstructured environments. This work also addressed a fundamental problem in virtual environment research: the automatic generation of realistic haptic models. Without intelligent, autonomous exploratory procedures, reality-based modeling requires tedious data acquisition and simulation programming. This work demonstrates that haptic virtual environments can be created from data obtained with an autonomous robotic finger. Applications include remote planetary exploration, robot-assisted and simulated surgery, and undersea salvage and discovery. Reality-based haptic models will be useful in any situation where the user may not have direct access, yet desires (or requires) cutaneous or kinesthetic information.

In future work, a new automated system for exploration and display will be designed. Software will be created to generate and evaluate models real-time during exploration. There are many possible surface and volumetric models that can be used for haptic display, other than the shape skeletons considered here. We will explore additional object properties including global shape, texture and roughness, and hardness/stiffness. New robotic fingers, specially designed for exploration, are also under development. Rolling fingertips will facilitate data collection from tactile sensors that function poorly under sliding conditions. In addition, when multiple fingers are used for combined manipulation and exploration, the design will improve object workspace without regrasping. This will allow simultaneous exploration and modeling of global and local shape.

Acknowledgments.

The author acknowledges Dr. M. Cutkosky and Dr. B. Roth for their feedback. Dr. H. Maekawa of AIST (Japan) provided the optical waveguide tactile sensor. J. Corso, J. Chhugani and S. Khor programmed the haptic rendering algorithms. This research was supported by an NSF Dissertation Enhancement Award and an Army STRICOM Grant through Immersion Corp.

References

- Allen PK, Michelman P (1990) Acquisition and interpretation of 3-d sensor data from touch. *IEEE Trans. on Robotics and Automation*, 6(4):397–404.
- Bay JS, Hemami H (1990) Dynamics of a learning controller for surface tracking robots on unknown surfaces. *Proc. of the IEEE Int'l. Conf. on Robotics and Automation*, pp 1910–1915.
- Blum H (1967) A transformation for extracting new descriptors of shape. *Models for Perception of Speech and Visual Form*, pp 362–380.
- Brandt JW, Algazi VR (1992) Continuous skeleton computation by voronoi diagram. *CVGIP: Image Understanding*, 55(3):329–338.
- Corso J, Chhugani J, Okamura A (2002) Interactive haptic rendering of deformable surfaces based on the medial axis transform. *Proc. Eurohaptics*.
- Curless B, Levoy M (1996) A volumetric method for building complex models from range images. *Computer Graphics*, 30:303–312.

- d'Aulignac D, Balaniuk R, Laugier C (2000) A haptic interface for a virtual exam of the human thigh. Proc. IEEE Int'l. Conf. on Robotics and Automation, 3:2452–2457.
- Debus T, Dupont P, Howe R (2000) Automatic identification of local geometric properties during teleoperation. Proc. IEEE Int'l. Conf. on Robotics and Automation, pp 3428–3434.
- Ellis RE (1984) Extraction of tactile features by passive and active sensing. Intelligent Robots and Computer Vision, Proc. of SPIE, 521:289–295.
- Goldenberg AS, Wies EF, Martin K, Hasser CJ (1998) Next-generation 3d haptic feedback system. Poster, American Soc. of Mechanical Engineers Haptics Symposium.
- Hayward V, Armstrong B (2000) A new computational model of friction applied to haptic rendering. In Corke P, Trevelyan J (eds) Exp. Robotics VI, pp 403–412. Springer-Verlag.
- Hemami H, Bay JS, Goddard RE (1988) A conceptual framework of tactually guided exploration and shape perception. IEEE Trans. on Biomedical Engineering, 35(2):99–109.
- Hogan N (1985) Impedance control: an approach to manipulation: parts i, ii, and iii. ASME Jnl. of Dynamic Systems, Measurement, and Control, 107:1–24.
- Interrante V (1997) Illustrating surface shape in volume data via principal direction-driven 3d line integral convolution. Proc. 24th Int'l. Conf. on Computer Graphics and Interactive Techniques, pp 109–116.
- Klatzky RL, Lederman S (1990) Intelligent exploration by the human hand. In Venkataraman ST, Iberall T (eds) Dextrous Robot Hands, pp 66–81. Springer-Verlag.
- Koenderink JJ (1990) Solid shape. The MIT Press.
- Li ZX, Qin Z, Jiang S, Han L (1998) Coordinated motion generation and real-time grasping force control for multifingered manipulation. Proc. IEEE Int'l. Conf. on Robotics and Automation, pp 3631–3638.
- Lim CT, Turkiyyah GM, Ganter MA, Storti DW (1995) Implicit reconstruction of solids from cloud point sets. ACM SIGGRAPH Symposium on Solid Modeling and Applications, pp 393–402.
- Maekawa H, Tanie K, Komoriya K (1993) A finger-shaped tactile sensor using an optical waveguide. Proc. IEEE Int'l. Conf. on Systems, Man and Cybernetics, pp 403–408.
- Maekawa H, Tanie K, Komoriya K (1995) Tactile sensor based manipulation of an unknown object by a multifingered hand with rolling contact. Proc. of the IEEE Int'l. Conf. on Robotics and Automation, pp 743–750.
- Massie TH, Salisbury JK (1994) The phantom haptic interface: A device for probing virtual objects. Proc. ASME Dynamic Systems and Control Div., pp 295–299.
- Montana DJ (1988) The kinematics of contact and grasp. Int'l. Jnl. of Robotics Research, 7 (3):17–32.
- Nicholls HR, Lee MH (1989) Survey of robot tactile sensing technology. Int'l. Jnl. of Robotics Research, 8(3):3–30.
- Ogniewicz RL (1994) Skeleton-space: a multiscale shape description combining region and boundary information. Proc. IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition, pp 746–751.
- Okamura AM, Cutkosky MR (2001a) Feature detection for haptic exploration with robotic fingers. Int'l. Jnl. of Robotics Research, 20(12):925–938.
- Okamura AM, Cutkosky MR (2001b) Feature-guided exploration with a robotic finger. Proc. IEEE Int'l. Conf. on Robotics and Automation, pp 589–596.
- Okamura AM, Dennerlein JT, Howe RD (1998) Vibration feedback models for virtual environments. Proc. of the IEEE Int'l. Conf. on Robotics and Automation, 1:674–679.
- Okamura AM, Smaby N, Cutkosky MR (2000) An overview of dexterous manipulation. Proc. IEEE Int'l. Conf. on Robotics and Automation, pp 255–262.

- Pai DK, Lang J, Lloyd JE, Woodham RJ (2000) Acme, a telerobotic active measurement facility. In Corke P, Trevelyan J (eds) Experimental Robotics VI, pp 391–400. Springer-Verlag.
- Patrikalakis NM, Gurusoy HN (1990) Shape interrogation by medial axis transform. Proc. ASME DETC, Design Eng. Div., 3(1):77–88.
- Richard C, Cutkosky MR, MacLean K (1999) Friction identification for haptic display. Proc. ASME Dynamic Systems and Control Div., pp 327–334.
- Richmond JL, Pai DK (2000) Active measurement and modeling of contact sounds. Proc. IEEE Int'l. Conf. on Robotics and Automation, pp 2146–2152.
- Rossignac JR, Requicha AAG (1986) Offsetting operations in solid modeling. Computer Aided Geometric Design, 3(2):129–148.
- Stansfield S (1988) A robotic perceptual system utilizing passive vision and active touch. Int'l. Jnl. of Robotics Research, 7(6):138–161.

Human-Centered Robotics and Interactive Haptic Simulation

O. Khatib, O. Brock, K.C. Chang, D. Ruspini, L. Sentis, and S. Viji

Department of Computer Science
Stanford University, Stanford, California 94305

Abstract. A new field of robotics is emerging. Robots are today moving towards applications beyond the structured environment of a manufacturing plant. They are making their way into the everyday world that people inhabit. The paper focuses on models, strategies, and algorithms associated with the autonomous behaviors needed for robots to work, assist, and cooperate with humans. In addition to the new capabilities they bring to the physical robot, these models and algorithms and more generally the body of developments in robotics is having a significant impact on the virtual world. Haptic interaction with an accurate dynamic simulation provides unique insights into the real-world behaviors of physical systems. The potential applications of this emerging technology include virtual prototyping, animation, surgery, robotics, cooperative design, and education among many others. Haptics is one area where the computational requirement associated with the resolution in real-time of the dynamics and contact forces of the virtual environment is particularly challenging. The paper describes various methodologies and algorithms that address the computational challenges associated with interactive simulations involving multiple contacts and impacts between human-like structures.

1 Introduction

The successful introduction of robotics into human environments will rely on the development of competent and practical systems that are dependable, safe, and easy to use. To work, cooperate, assist, and interact with humans, the new generation of robot must have mechanical structures that accommodate the interaction with the human and adequately fit in his unstructured and sizable environment. Human-compatible robotic structures must integrate mobility (legged or wheeled) and manipulation (preferably bi-manual), while providing the needed access to perception and monitoring (head camera) (Asfour, Berns, Schelling, Dillmann, 1999; Hirai, Hirose, Haikawa, Takenaka, 1998; Khatib, Yokoi, Brock, Chang, Casal, 1999; Nishiwaki, Sugihara, Kagami, Kanehiro, Inaba, Inoue, 2000; Takanishi, Hirano, Sato, 1998). These requirements imply robots with branching structures - tree-like topology involving much larger numbers of degrees of freedom than those usually found in conventional industrial robots. The substantial increase in the dimensions of the corresponding configuration spaces of these robots renders the set of fundamental problems associated with their modeling, programming, planning, and control much more challenging.

The first of these challenges is the whole-robot modeling, motion coordination, and dynamic control. For robots with human-like structures, tasks are not limited to

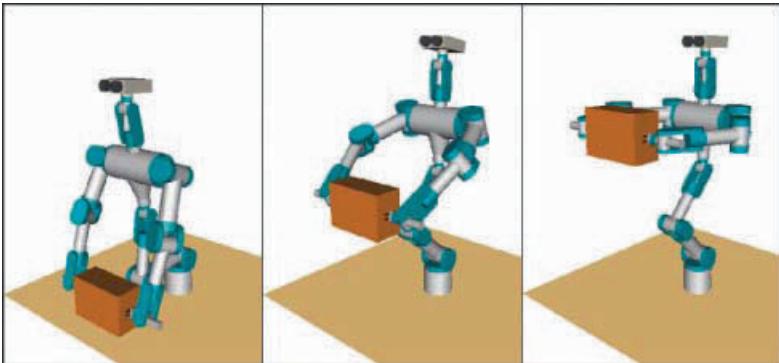


Fig. 1. Manipulation and Posture Behaviors: a sequence of three snapshots from the dynamic simulation of a 24-degree-of-freedom humanoid system, whose task is generated from simple manipulation and posture behaviors.

the specification of the position and orientation of a single effector. For these robots, task descriptions may involve combinations of coordinates associated with one or both arms, the head-camera, and/or the torso among others. The remaining freedom of motion is assigned to various criteria related to the robot posture and its internal and environmental constraints.

There is a large body of work devoted to the study of motion coordination in the context of kinematic redundancy. In recent years, algorithms developed for redundant manipulators have been extended to mobile manipulation robots (Cameron, MacKenzie, Ward, Arkin, Book, 1993; Papadopoulos, Dubowsky, 1991; Ullman, Cannon, 1989; Umetani, Yoshida, 1989). Typical approaches to motion coordination of redundant systems rely on the use of pseudo or generalized inverses to solve an under-constrained or degenerate system of linear equations, while optimizing some given criterion. These algorithms are essentially driven by kinematic considerations and the dynamic interaction between the end effector and the robot's self motions are ignored.

Our effort in this area has resulted in a *task-oriented* framework for whole-robot dynamic coordination and control (Khatib, Yokoi, Chang, Ruspini, Holmberg, Casal, 1996). The dynamic coordination strategy we developed is based on two models concerned with the task dynamics (Khatib, Yokoi, Brock, Chang, Casal, 1987) and the robot posture behavior. The *task dynamic behavior* model is obtained by a projection of the robot dynamics into the space associated with the task, while the *posture behavior* is characterized by the complement of this projection. To control these two behaviors, a consistent control structure is required. The paper discusses these models and presents a unique control structure that guarantees *dynamic consistency* and decoupled posture control (Khatib, 1995), while providing optimal responsiveness for the task. The paper also presents recently developed recursive algorithms which efficiently address the computational challenges associated with branching mechanisms. Dynamic simulation of virtual environments is

another important area of applications of these algorithms. The paper also discusses our ongoing effort for the development of a general framework for interactive haptic simulation that addresses the problem of contact resolution.

A robotic system must be capable of sufficient level of competence to avoid obstacles during motion. Even when a path is provided by a human or other intelligent planner, sensor uncertainties and unexpected obstacles can make the motion impossible to complete. Our research on the artificial potential field method (Khatib, 1986) has addressed this problem at the control level to provide efficient real-time collision avoidance. Due to their local nature, however, reactive methods (Arkin, 1987; Khatib, 1986; Krogh, 1984; Latombe, 1991) are limited in their ability to deal with complex environments. Using navigation functions (Koditschek, 1987) the problems arising from the locality of the potential field approach can be overcome. These approaches, however, do not extend well to robots with many degrees of freedom, such as mobile manipulators (Carriker, Khosla, Krogh, 1989; Seraji, 1993; Yamamoto, Yun, 1995). Our investigation of a framework to integrate real-time collision avoidance capabilities with a global collision-free path has resulted in the *elastic band* approach (Quinlan, Khatib, 1993), which combines the benefits of global planning and reactive systems in the execution of motion tasks. The concept of elastic bands was also extended to nonholonomic robots (Khatib, Jaouni, Chatila, Laumond, 1997). The paper discusses our ongoing work in this area and presents extensions to the *elastic strip* approach (Brock, Khatib, 1997), which enable real-time obstacle avoidance in a task-consistent manner. Task behavior can be suspended and resumed in response to changes in the environment to ensure collision avoidance under all circumstances.

2 Whole-Robot Control: Task and Posture

Human-like structures share many of the characteristics of macro/mini structures (Khatib, 1995): coarse and slow dynamic responses of the mobility system (the macro mechanism), and the relatively fast responses and higher accuracy of the arms (the mini device). Inspired by these properties of macro/mini structures, we have developed a framework for the coordination and control of robots with human-like structures. This framework provides a unique control structure for decoupled manipulation and posture control, while achieving optimal responsiveness for the task. This control structure is based on two models concerned with the task dynamic behavior and the robot posture behavior. The *task behavior* model is obtained by a projection of the robot dynamics into the space associated with the effector task, and the *posture behavior* model is characterized by the complement of this projection. We first present the basic models associated with the task. In a subsequent section we present the whole-robot coordination strategy and posture control behavior.

2.1 Task Dynamic Behavior

The joint space dynamics of a manipulator are described by

$$A(\mathbf{q})\ddot{\mathbf{q}} + b(\mathbf{q}, \dot{\mathbf{q}}) + g(\mathbf{q}) = \Gamma \quad (1)$$

where \mathbf{q} is the n joint coordinates, $A(\mathbf{q})$ is the $n \times n$ kinetic energy matrix, $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ is the vector of centrifugal and Coriolis joint forces, $\mathbf{g}(\mathbf{q})$ is the vector of gravity, and Γ is the vector of generalized joint forces.

The *operational space formulation* (Khatib, 1987) provides an effective framework for dynamic modeling and control of branching mechanisms (Russakow, Khatib, Rock, 1995), with multiple operational points. The generalized torque/force relationship (Khatib, 1987, 1995) provides the decomposition of the total torque, Γ (1) into two dynamically decoupled command torque vectors: the torque corresponding to the task behavior command vector and the torque that only affects posture behavior in the null space:

$$\Gamma = \Gamma_{task} + \Gamma_{posture} \quad (2)$$

For a robot with a branching structure of m effectors or operational points, the task is represented by the $6m \times 1$ vector, \mathbf{x} , and the $6m \times n$ Jacobian matrix is $J(\mathbf{q})$. This Jacobian matrix is formed by vertically concatenating the m $6 \times n$ Jacobian associated with the m effectors.

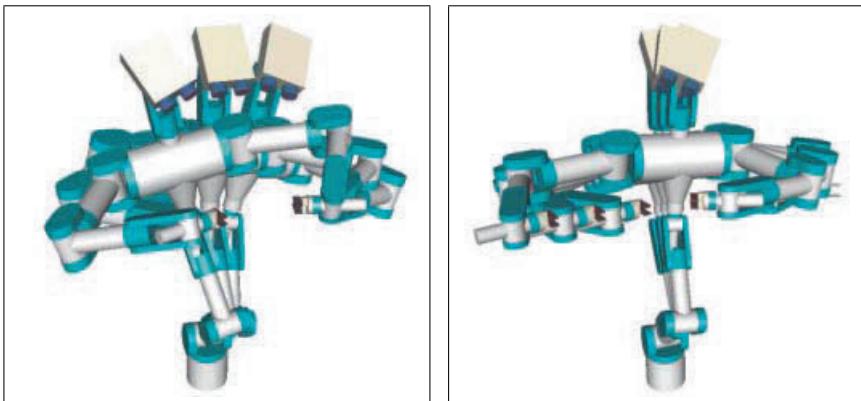


Fig. 2. Dynamic Consistency and Posture Behaviors: a sequence of snapshots from the dynamic simulation of a 24-degree-of-freedom humanoid system. On the left, the task is to maintain a constant position for the two hands, while achieving hand-eye coordination. The posture motion has no effect on the task. On the right, the task also involves hand-eye coordination and motion of the common hand position. This position is interactively driven by the user. The posture is to maintain the robot total center-of-mass along the z -axis.

The task dynamic behavior is described by the operational space equations of motion (Khatib, 1995)

$$\Lambda(\mathbf{x})\ddot{\mathbf{x}} + \mu(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{p}(\mathbf{x}) = \mathbf{F} \quad (3)$$

where \mathbf{x} , is the vector of the $6m$ operational coordinates describing the position and orientation of the m effectors, $\Lambda(\mathbf{x})$ is the $6m \times 6m$ kinetic energy matrix associated

with the operational space. $\mu(\mathbf{x}, \dot{\mathbf{x}})$, $\mathbf{p}(\mathbf{x})$, and \mathbf{F} are respectively the centrifugal and Coriolis force vector, gravity force vector, and generalized force vector acting in operational space.

The joint torque corresponding to the task command vector \mathbf{F} , acting in the operational space is

$$\Gamma_{\text{task}} = J^T(\mathbf{q})\mathbf{F} \quad (4)$$

The task dynamic decoupling and control is achieved using the control structure

$$\mathbf{F}_{\text{task}} = \widehat{\Lambda}(\mathbf{x})\mathbf{F}_{\text{motion}}^* + \widehat{\mu}(\mathbf{x}, \dot{\mathbf{x}}) + \widehat{\mathbf{p}}(\mathbf{x}) \quad (5)$$

where, $\mathbf{F}_{\text{task}}^*$ represents the inputs to the decoupled system, and $\widehat{\cdot}$ represents estimates of the model parameters.

2.2 Posture Behavior

An important consideration in the development of posture behaviors is the interactions between the posture and the task. It is critical for the task to maintain its responsiveness and to be dynamically decoupled from the posture behavior. The posture can then be treated separately from the task, allowing intuitive task and posture specifications and effective whole-robot control. The overall control structure for task and posture is

$$\Gamma = \Gamma_{\text{task}} + \Gamma_{\text{posture}} \quad (6)$$

where

$$\Gamma_{\text{posture}} = N^T(\mathbf{q})\Gamma_{\text{desired-posture}} \quad (7)$$

with

$$N(\mathbf{q}) = [I - \overline{J}(\mathbf{q})J(\mathbf{q})] \quad (8)$$

where $\overline{J}(\mathbf{q})$ is the *dynamically consistent generalized inverse* (Khatib, 1995), which minimizes the robot kinetic energy

$$\overline{J}(\mathbf{q}) = A^{-1}(\mathbf{q})J^T(\mathbf{q})\Lambda(\mathbf{q}) \quad (9)$$

and

$$\Lambda(\mathbf{q}) = [J(\mathbf{q})A^{-1}(\mathbf{q})J^T(\mathbf{q})]^{-1} \quad (10)$$

This relationship provides a decomposition of joint forces into two control vectors: joint forces corresponding to forces acting at the task, $J^T\mathbf{F}$, and joint forces that only affect the robot posture, $N^T\Gamma_{\text{posture}}$. For a given task this control structure

produces joint motions that minimize the robot's instantaneous kinetic energy. As a result, a task will be carried out by the combined action of the set of joints that reflect the smallest effective inertial properties.

To control the robot for a desired posture, the vector $\Gamma_{\text{desired-posture}}$ will be selected as the gradient of a potential function constructed to meet the desired posture specifications. The interference of this gradient with the task dynamics is avoided by projecting it into the dynamically consistent null space of $J^T(\mathbf{q})$, i.e. $N^T(\mathbf{q})\Gamma_{\text{desired-posture}}$.

Dynamic consistency is the essential property for the task behavior to maintain its responsiveness and to be dynamically decoupled from the posture behavior since it guarantees not to produce any coupling acceleration in the operational space given any τ_{null} . In Fig. 2 (left), the robot (a 24-degree-of-freedom humanoid system) was commanded to keep the position of both hands constant (task behavior) while moving its left and right in the null space (posture behavior). Notice that dynamic consistency enables task behavior and posture behavior to be specified independently of each other, providing an intuitive control of complex systems.

For instance, the robot posture can be controlled to maintain the robot total center-of-mass aligned along the z_0 axis of the reference frame. This posture can be simply implemented with a posture energy function

$$V_{\text{posture-energy}}(\mathbf{q}) = \frac{1}{2}k(x_{\text{CoM}}^2 + y_{\text{CoM}}^2) \quad (11)$$

where k is a constant gain, x_{CoM} , and y_{CoM} are the x and y coordinates of the center of mass. The gradient of this function

$$\Gamma_{\text{desired-posture}} = J_{\text{CoM}}^T(-\nabla V_{\text{posture-energy}}) \quad (12)$$

provides the required attraction to the z axis of the robot center of mass. This is illustrated in the simulation shown in Fig. 2 (right), whose task involved three operational point associated with the arms and head.

Collision avoidance can be also integrated in the posture control as discussed in section 4. With this posture behavior, the explicit specification of the associated motions is avoided, since desired behaviors are simply encoded into specialized potential functions for various types of operations.

More complex posture behaviors can be obtained by combining various posture energies. We are currently exploring the generation of human-like natural motion from motion capture of human and the extraction of motion characteristics using human biomechanical models.

2.3 Cooperative Manipulation

The development of effective cooperation strategies for multiple robots platforms is an important issue for both the operations in human environments and the interaction with humans. Human guided motions may involve tightly constrained cooperation performed through compliant motion actions or less restricted tasks executed



Fig. 3. Cooperative Manipulation with the Stanford Robotic Platforms

through simpler free-space motion commands. Several cooperative robots, for instance, may support a load while being guided by the human to an attachment, or visually following the guide to a destination.

Our approach is based on the integration of two basic concepts: The *augmented object* (Khatib, 1988) and the *virtual linkage* (Williams, Khatib, 1993). The *virtual linkage* characterizes internal forces, while the *augmented object* describes the system's closed-chain dynamics. For systems of a mobile nature, a *decentralized* control structure is needed to address the difficulty of achieving high-rate communication between platforms. In the decentralized control structure, the object level specifications of the task are transformed into individual tasks for each of the cooperative robots. Local feedback control loops are then developed at each grasp point. The task transformation and the design of the local controllers are accomplished in consistency with the *augmented object* and *virtual linkage* models (Khatib, 1988; Williams, Khatib, 1993). This approach has been successfully implemented on the Stanford robotic platforms (see Fig. 3) for cooperative manipulation and human-guided motions.

2.4 Efficient Operational Space Algorithms

Early work on efficient operational space dynamic algorithms has focused on open-chain robotic mechanisms. An efficient O(n) recursive algorithm was developed using the spatial operator algebra (Kreutz-Delgado, Jain, Rodriguez, 1991; Rodriguez,

Kreutz, Jain, 1989) and the articulated-body inertias (Featherstone, 1987). A different approach that avoided the extra computation of articulated inertias also resulted in an $O(n)$ recursive algorithm for the operational space dynamics (Lilly, 1992; Lilly, Orin, 1993) Building on these early developments, our effort was aimed at algorithms for robotic mechanisms with branching structures that also address the issue of redundancy and dynamics in the null space.

The most computationally expensive element in the operational space whole-body control structure (6) is the posture control, which involves the explicit inversion operation of the $n \times n$ joint space inertia matrix A of (9), which requires $O(n^3)$. We have developed a computationally more efficient operational space control structure that eliminates the explicit computation of the joint space inertia matrix and its inverse. This elimination was achieved by combining the dynamically consistent null space control and the operational space control in a computationally more efficient dynamic control structure.

Using this control structure, we have developed a recursive algorithm for computing the operational space dynamics of an n -joint branching redundant articulated robotic mechanism with m operational points (Chang, Khatib, 2000). The computational complexity of this algorithm is $O(nm + m^3)$, while existing symbolic methods require $O(n^3 + m^3)$. Since m can be considered as a small constant in practice, this algorithm attains a linear time $O(n)$ as the number of links increases. This work was extended for the dynamics of closed-chain branching mechanisms with an efficient $O(nm + m^3)$ algorithm (Chang, Holmberg, Khatib, 2000).

3 Interactive Haptic Simulation

Beyond their immediate application to physical robots, these efficient dynamic algorithms are making a significant impact on the simulation and interaction with the virtual world. The computational requirements associated with the haptic interaction with complex dynamic environments are quite challenging. In addition to the need for real-time free-motion simulation of multi-body systems, contact and impact resolution and constrained motion simulation are also needed.

Building on the operational space formulation, we developed a general framework (Ruspini, Khatib, 1999) for the resolution of multi-contact between articulated multi-body systems. A contact point is treated as an operational point and a contact space is defined. Similarly to the operational space inertia matrix, a contact space inertia matrix Λ is introduced to provide the effective masses seen at all the contact points and to characterize the dynamic relationships between them. Computing the contact space inertia matrices Λ for a number of m contact point on a branching mechanism is achieved with an efficient $O(nm + m^3)$ recursive algorithm.

The contact space representation allows the interaction between groups of dynamic systems to be described easily without having to examine the complex equations of motion of each individual system. As such, a collision model can be developed with the same ease as if one was considering interaction only between simple

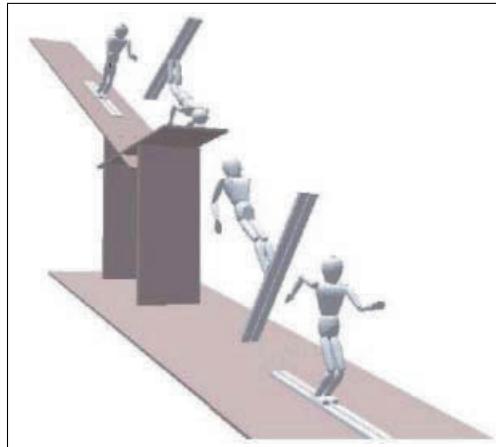


Fig. 4. A frame of an animation showing the dynamic interaction of multiple articulated/rigid bodies (left). A similar sequence in which direct haptic interaction is permitted between the user and the objects in the environment (right).

bodies. Impact and contact forces between interacting bodies can then be efficiently solved to prevent penetration between all the objects in the environment.

This framework was integrated with our haptic rendering system (Ruspini, Kollarov, Khatib, 1997) to provide a general environment for interactive haptic dynamic simulation. Figure 4 illustrates some of the virtual environments that have been modeled with this system. Figure 4(left) is one frame from an animation consisting of two puma560 manipulators (6 d.o.f. each) on which a rain of large blocks is allowed to fall. A total of 366 d.o.f. are modeled.

In Fig. 4(right) a similar environment where two puma560 manipulators and two rigid bodies (16 d.o.f.) is modeled. Direct haptic interaction is permitted via a 3 d.o.f PHANTOM haptic manipulator. The user is allowed to push and attach oneself to any of the objects in the environment and feel the force and impact created by their interaction.

4 Task-Consistent Elastic Plans

The control methods presented in Section 2 allow the consistent control of task and posture for robots with complex mechanical structures, such as human-like robots. To perform or assist in the execution of complex actions, however, these control structures have to be linked with motion generated by a planner. Furthermore, since unstructured environments can be highly dynamic, such an integration has to accommodate unforeseen obstacle motion in real time, while conforming to constraints imposed by the task. We have developed algorithms that perform task-consistent, real-time path modification to address this issue.

4.1 Real-Time Path Modification

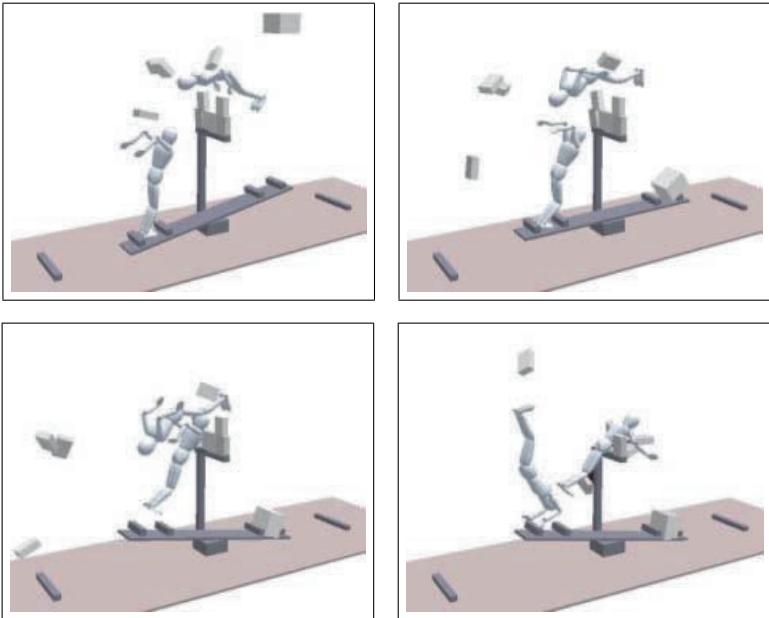


Fig. 5. Top images show from left to right: real-time obstacle avoidance without task consistency, with task consistency, and transitioning between task-consistent and non task-consistent behavior. Lines indicate trajectories of the base, elbow, and end effector. The graphs show end effector error and base deviation from the task for the respective experiment.

Motion planners generally perform a global search in configuration space to determine a collision-free motion accomplishing a given task. Due to the high dimensionality of the configuration space of the class of robots we are concerned with in this paper, planning operations are too computationally complex to be performed in real time. As a consequence, motion in dynamic environments cannot adequately be generated by those planners. The elastic band framework (Quinlan, Khatib, 1993) was developed to allow real-time modification of a previously planned path, effectively avoiding a costly planning operation in reaction to changes in the environment. More recently, this framework was complemented by the elastic strip framework (Brock, Khatib, 1997).

The elastic strip framework augments the representation of a path computed by a planner with a description of free space around that path. Collision avoidance can be guaranteed, if the work space volume swept by the robot along its path is contained within the free space. Real-time path modification is implemented by subjecting the entire path to an artificial potential field (Khatib, 1986), keeping the

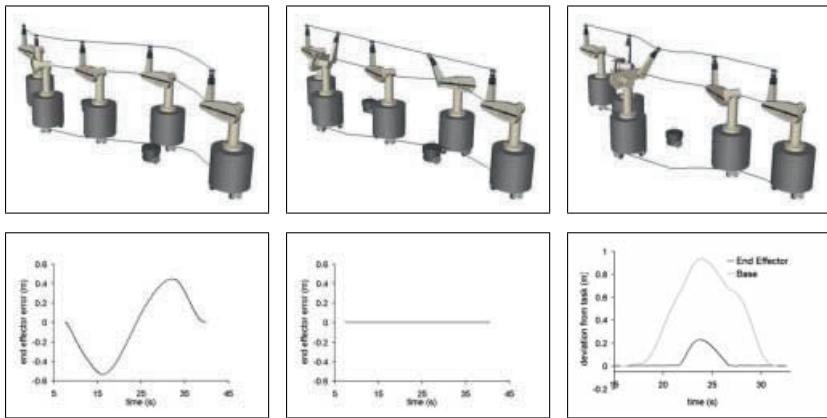


Fig. 6. Task-consistent, real-time obstacle avoidance: despite the base changing its trajectory to avoid the moving obstacle, the end effector performs a straight-line trajectory

path at a safe distance from obstacles. The modification of the path in accordance with those potentials is performed while ensuring that the volume swept by the robot along the path is always contained within the representation of local free space. This results in “elastic” paths, which deform in reaction to approaching obstacles, while maintaining the global properties of the path.

In addition to the repulsive, external potential, we also apply internal forces to consecutive configurations of the robot along the path. This shortens and smoothes the path. The overall behavior of a path represented in the elastic strip framework can be compared to a string of elastic material: as obstacles approach, the path is locally modified or “stretched” by repulsive forces; once the obstacle moves further away, internal forces shorten and smoothen the path. Such behavior is shown in the top left image of Fig. 5. The end effector of the robot is commanded to move on a straight line. The original path generated by the planner is a simple straight-line motion, as no obstacles are present. The image shows the path after two mobile obstacles move into the path.

The elastic strip framework scales to robots with many degrees of freedom and with many operational points, as it avoids a costly search for collision-free motion in configuration space. Instead, it employs simple work space-based potential fields in conjunction with aforementioned control structures to modify a previously planned motion in real time.

4.2 Task-Consistent Path Modification

In dynamic environments it is desirable to integrate reactive obstacle avoidance with task behavior. To accomplish this we extend the overall control structure for task and posture behavior (2) by adding torques $\Gamma_{obstacle}$ representing desired obstacle

avoidance behavior:

$$\Gamma = \Gamma_{task} + \Gamma_{posture} + \Gamma_{obstacle}$$

Both, $\Gamma_{posture}$ and $\Gamma_{obstacle}$ have been mapped into the nullspace of the task, as shown in (7). Using this control structure, the task-consistent obstacle avoidance behavior shown in the second image of Fig. 5 is achieved. Note how without task-consistency the end effector deviates significantly from the required straight-line trajectory. Using task-consistent obstacle avoidance, the end effector only deviates minimally from the task, as can be seen in the graphs shown in Fig. 5. A task-consistent motion execution on the Stanford Assistant Manipulator can be seen in Fig. 6.

This approach of integrating task and obstacle avoidance behavior can fail, however, when the torques resulting from mapping $\Gamma_{obstacle}$ into the nullspace yield insufficient motion to ensure obstacle avoidance. In such a situation it would be desirable to suspend task execution and to realize obstacle avoidance with all degrees of freedom of the robot.

Let $N^T(J(\mathbf{q})) = [I - J^T(\mathbf{q}) \bar{J}^T(\mathbf{q})]$ be the dynamically consistent nullspace mapping of the Jacobian $J(\mathbf{q})$ associated with the task. The coefficient

$$c = \frac{\|N^T(J(\mathbf{q})) \Gamma_{obstacle}\|}{\|\Gamma_{obstacle}\|}$$

corresponds to the ratio of the magnitude of the torque vector $\Gamma_{obstacle}$ mapped into that nullspace to its unmapped magnitude. This coefficient is an indication of how well the behavior represented by $\Gamma_{obstacle}$ can be performed inside the nullspace of the task. We experimentally determine a value c_s at which it is desirable to suspend task execution in favor of the behavior previously mapped into the nullspace. Once the coefficient c assumes a value $c < c_s$, a transition is initiated. During this transition task behavior is gradually suspended and previous nullspace behavior is performed using all degrees of freedom of the manipulator. The motion of the manipulator is now generated using the equation

$$\begin{aligned} \Gamma = & \alpha J^T(\mathbf{q}) \mathbf{F} + \\ & \alpha [I - J^T(\mathbf{q}) \bar{J}^T(\mathbf{q})] \Gamma_{obstacle} + \\ & \bar{\alpha} \Gamma_{obstacle} \end{aligned}$$

where $\alpha \in [0..1]$ is a time-based transition variable, transitioning between 1 and 0 during task suspension and between 0 and 1 during resumption of the task, and $\bar{\alpha} = (1 - \alpha)$ is defined as the complement of α .

The experimental results, performed on the Stanford Assistant Manipulator, for such transitioning behavior can be seen in Fig. 5. The image on the top right shows how despite task-consistent obstacle avoidance the task has to be suspended to ensure obstacle avoidance. Below, the graph shows how the base deviates significantly from the straight line in response to the obstacle. The end effector, however, maintains the task until it has to be suspended. The graph also shows that the task is resumed in a smooth manner, after the base has passed the obstacle

5 Conclusion

Advances toward the challenge of robotics in human environments depend on the development of the basic capabilities needed for both autonomous operations and human/robot interaction. In this article, we have presented methodologies for whole-robot coordination and control, cooperation between multiple robots, interactive haptic simulation with contact, and the real-time modification of collision-free path to accommodate changes in the environment.

For the whole-robot coordination and control, we presented a framework which provides the user with two basic task-oriented control primitives: task control and posture control. The major characteristic of this control structure is the dynamic consistency it provides in implementing these two primitives: the robot posture behavior has no impact on the end-effector dynamic behavior. While ensuring dynamic decoupling and improved performance, this control structure provides the user with a higher level of abstraction in dealing with task specifications and control.

Addressing the computational challenges of human-like robotic structures, we presented efficient $O(nm + m^3)$ recursive algorithms for the operational space dynamics of mechanisms involving branching structures and closed chains. Building on the operational space formulation, we also developed a framework for the resolution of multi-contact between articulated multi-body systems. The computational efficiency of the dynamic algorithms developed for physical robots provided the interactivity needed for haptic simulation of complex virtual environments.

The elastic strip framework allows the seamless integration of reactive, real-time obstacle avoidance and the task-oriented control structure. It provides for real-time motion generation that combines obstacle avoidance and task execution. When kinematic or external constraints imposed by obstacles make it impossible to maintain the task, task-consistent obstacle avoidance is suspended and all degrees of freedom are relaxed, the task is resumed in a smooth manner. Using the elastic strip framework, motion for complex kinematic structures can be generated very efficiently, as the required computations are mostly performed in work space and as a result are independent of the number of degrees of freedom of the mechanism.

Acknowledgments.

The financial support of Honda Motor Company and NSF (grants IRI-9320017) is gratefully acknowledged. Many thanks to Alan Bowling, Arancha Casal, Francois Conti, Robert Holmberg, Jaeheung Park, Costas Stratelos, James Warren, and Kazuhito Yokoi for their valuable contributions to the work reported here.

References

- Arkin RC (1987) Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112.

- Asfour T, Berns K, Schelling J, Dillmann R (1999) Programming of manipulation tasks of the humanoid robot ARMAR. In Proceedings of the International Conference on Advanced Robotics.
- Brock O, Khatib O (1997) Elastic strips: Real-time path modification for mobile manipulation. In Proceedings of the International Symposium of Robotics Research, pp 117–122. preprints.
- Cameron JM, MacKenzie DC, Ward KR, Arkin RC, Book WJ (1993) Reactive control for mobile manipulation. In Proceedings of the International Conference on Robotics and Automation, vol 3, pp 228–235.
- Carriker WF, Khosla PK, Krogh BH (1989) An approach for coordinating mobility and manipulation. In Proceedings of the International Conference on Systems Engineering, pp 59–63.
- Chang K, Holmberg R, Khatib O (2000) The augmented object model: Cooperative manipulation and parallel mechanism dynamics. In Proceedings of the International Conference on Robotics and Automation, pp 470–47, San Francisco, CA, U.S.A.
- Chang K, Khatib O (2000) Operational space dynamics: Efficient algorithms for modeling and control of branching mechanisms. In Proceedings of the International Conference on Robotics and Automation, pp 850–856, San Francisco, CA, U.S.A.
- Featherstone R (1987) Robot dynamics algorithms. Kluwer Academic Publishers.
- Hirai K, Hirose M, Haikawa Y, Takenaka T (1998) The development of Honda humanoid robot. In Proceedings of the International Conference on Robotics and Automation, vol 2, pp 1321–1326, Leuven, Belgium.
- Khatib M, Jaouni H, Chatila R, Laumond JP (1997) How to implement dynamic paths. In Proceedings of the International Symposium on Experimental Robotics, pp 225–36. Preprints.
- Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research, 5(1):90–98.
- Khatib O (1987) A unified approach to motion and force control of robot manipulators: The operational space formulation. International Journal of Robotics and Automation, RA-3 (1):43–53.
- Khatib O (1988) Object manipulation in a multi-effector robot system. In Bolles R, Roth B (eds) Robotics Research 4, pp 137–144. MIT Press.
- Khatib O (1995) Inertial properties in robotics manipulation: An object-level framework. International Journal of Robotics Research, 14(1):19–36.
- Khatib O, Yokoi K, Brock O, Chang KS, Casal A (1987) A unified approach to motion and force control of robot manipulators: The operational space formulation. International Journal of Robotics Research, 3(1):43–53.
- Khatib O, Yokoi K, Brock O, Chang KS, Casal A (1999) Robots in human environments: Basic autonomous capabilities. International Journal of Robotics Research, 18(7):175–183.
- Khatib O, Yokoi K, Chang KS, Ruspini D, Holmberg R, Casal A (1996) Coordination and decentralized cooperation of multiple mobile manipulators. Journal of Robotic Systems, 13(11):755–64.
- Koditschek DE (1987) Exact robot navigation by means of potential functions: Some topological considerations. In Proceedings of the International Conference on Robotics and Automation, pp 1–6.
- Kreutz-Delgado K, Jain A, Rodriguez G (1991) Recursive formulation of operational space control. In Proceedings of the International Conference on Robotics and Automation, pp 1750–1753.

- Krogh BH (1984) A generalized potential field approach to obstacle avoidance control. In *Robotics Research: The Next Five Years and Beyond*.
- Latombe JC (1991) Robot motion planning. Kluwer Academic Publishers, Boston.
- Lilly KW (1992) Efficient dynamic simulation of robotic mechanisms. Kluwer Academic Publishers.
- Lilly KW, Orin DE (1993) Efficient O(N) recursive computation of the operational space inertia matrix. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(5):1384–1391.
- Nishiwaki K, Sugihara T, Kagami S, Kaneko F, Inaba M, Inoue H (2000) Design and development of research platform for perception-action integration in humanoid robot: H6. In *Proceedings of the International Conference on Intelligent Robots and Systems*, vol 3, pp 1559–1564.
- Papadopoulos E, Dubowsky S (1991) Coordinated manipulator/spa cecraft motion control for space robotic systems. In *Proceedings of the International Conference on Robotics and Automation*, pp 1696–1701.
- Quinlan S, Khatib O (1993) Elastic bands: Connecting path planning and control. In *Proceedings of the International Conference on Robotics and Automation*, vol 2, pp 802–7.
- Rodriguez G, Kreutz K, Jain A (1989) A spatial operator algebra for manipulator modeling and control. In *Proceedings of the International Conference on Robotics and Automation*, pp 1374–1379.
- Ruspini DC, Khatib O (1999) Collision/contact models for dynamic simulation and haptic interaction. In *9th International Symposium of Robotics Research (ISRR'99)*, pp 185–195, Snowbird, Utah, U.S.A.
- Ruspini DC, Kolarov K, Khatib O (1997) The haptic display of complex graphical environments. In *SIGGRAPH Conference Proceedings (Computer Graphics)*, pp 345–352, Los Angeles, CA, U.S.A. ACM SIGGRAPH.
- Russakow J, Khatib O, Rock SM (1995) Extended operational space formation for serial-to-parallel chain (branching) manipulators. In *Proceedings of the International Conference on Robotics and Automation*, vol 1, pp 1056–1061.
- Seraji H (1993) An on-line approach to coordinated mobility and manipulation. In *Proceedings of the International Conference on Robotics and Automation*, vol 1, pp 28–35.
- Takanishi A, Hirano S, Sato K (1998) Development of an anthropomorphic head-eye system for a humanoid robot-realization of human-like head-eye motion using eyelids adjusting to brightness. In *Proceedings of the International Conference on Robotics and Automation*, vol 2, pp 1308–1314, Leuven, Belgium.
- Ullman M, Cannon R (1989) Experiments in global navigation and control of a free-flying space robot. In *Winter*, vol 17, pp 37–43.
- Umetani Y, Yoshida K (1989) Experimental study on two-dimensional free-flying robot satellite model. In *Proceedings of the NASA Conference on Space Telerobotics*.
- Williams D, Khatib O (1993) The virtual linkage: A model for internal forces in multi-grasp manipulation. In *Proceedings of the International Conference on Robotics and Automation*, vol 1, pp 1025–1030. preprints.
- Yamamoto Y, Yun X (1995) Coordinated obstacle avoidance of a mobile manipulator. In *Proceedings of the International Conference on Robotics and Automation*, vol 3, pp 2255–60.

Collaboration, Dialogue, and Human-Robot Interaction

Terrence Fong^{1,2}, Charles Thorpe¹, and Charles Baur²

¹ The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

² Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

Abstract. Teleoperation can be improved if humans and robots work as partners, exchanging information and assisting one another to achieve common goals. In this paper, we discuss the importance of collaboration and dialogue in human-robot systems. We then present collaborative control, a system model in which human and robot collaborate, and describe its use in vehicle teleoperation.

1 Introduction

1.1 Robot as Partner

A robot is commonly viewed as a *tool*: a device that performs tasks on command. As such, a robot has limited freedom and will perform poorly whenever it is ill-suited for the task at hand. Moreover, if a robot has a problem, it has no way to ask for assistance. Yet, frequently, the only thing a robot needs to work better is some advice (even a small amount) from a human.

Consider the situation in which a mobile robot is driving outdoors when it encounters tall grass in its path. Depending on its sensors, the robot's perception system may have difficulty deciding if the grass poses a danger. Thus, the robot may be unable to proceed or may take a long, resource consuming detour. If, however, the robot is able to discuss the situation with a human, a better solution can be found. For example, if the robot asks "Is there an obstacle ahead?" and shows a camera image, the human can help decide that it is safe to drive forward.

Generally speaking, robots are more adept at making some decisions by themselves than others. For example, structured planning (for which algorithms or well-defined solutions exist) has proven to be quite amenable to automation. Unstructured decision making, however, remains the domain of humans, especially whenever common sense is required (Clarke, 1994). In particular, robots continue to perform poorly at high-level perceptual functions, including object recognition and situation assessment (Milgram et al., 1993).

In order for robots to perform better, therefore, they need to be able to take advantage of human skills (perception, cognition, etc.) and to benefit from human advice and expertise. To do this, robots need to function not as passive tools, but rather as active partners. They need to have more freedom of action, to be able to drive the interaction with humans, instead of merely waiting for (or blindly executing) human commands.

1.2 Collaborative Control

To address this need, we have developed a new system model for teleoperation called collaborative control (Fong, 2001). In this model, a human and a robot work as partners (if not peers), *collaborating* to perform tasks and to achieve common goals. Instead of a supervisor dictating to a subordinate, the human and the robot engage in *dialogue* to exchange ideas, to ask questions, and to resolve differences.

With collaborative control, the human functions as a resource for the robot, providing information and processing just like other system modules. In particular, the robot can ask questions to the human as it works, to obtain assistance during task execution. With this approach, the robot has more freedom in execution and is more likely to find good solutions when it has problems.

Collaborative control is a radical departure from traditional teleoperation, from the conventional “robot as tool” system model. Collaborative control encourages human-robot interaction to be more natural, more balanced, and more direct. Collaborative control also allows robots to benefit from human assistance during perception and cognition, and not just planning and command generation.

1.3 Key Issues

To build a collaborative control system, we have found that there are four key issues that must be addressed. First, the robot must have *self-awareness*. This does not imply that the robot needs to be fully sentient, merely that it be capable of detecting limitations (in what it can do and what the human can do), determining if it should ask for help, and recognizing when it has to solve problems on its own.

Second, the robot must be *self-reliant*. Since the robot cannot rely on the human to always be available or to provide accurate information, it must be able to maintain its own safety. Specifically, the robot should be capable of avoiding hazards, monitoring its health, and taking action to “safe” itself when necessary.

Third, the system must support *dialogue*. That is, the robot and the human need to be able to communicate effectively with each other. Each participant must be able to convey information, to ask questions and to judge the quality of responses received. To an extent, traditional teleoperation has dialogue (i.e., the feedback loop), but the conversation is limited. With collaborative control, dialogue is two-way and requires a richer vocabulary.

Finally, the system must be *adaptive*. By design, collaborative control provides a framework for integrating users with varied skills, knowledge, and experience. As a consequence, however, the robot has to be able to adapt to different operators and to adjust its behavior as needed, e.g., asking questions based on the operator’s capacity to answer.

2 Dialogue

2.1 Communication and Conversation

Dialogue is the process of communication between two or more parties. Dialogue is a joint process: it requires sharing of information (data, symbols, context) and of control. Depending on the situation (task, environment, etc.), the form or style of dialogue will vary. However, studies of human conversation have revealed that many properties of dialogue, such as initiative taking and error recovery, are always present (Lansdale, Ormerod, 1994).

When humans and machines (computers, robots, etc.) communicate, dialogue is usually mediated by an interface. Some interfaces (e.g., computer command languages) offer great power and flexibility, though at an associated high learning cost. Other interfaces, such as menus, are easier for novices because they make few assumptions about what the user knows. Regardless the form, however, a good interface provides structure that facilitates human-machine dialogue and information exchange.

2.2 Dialogue Management

Unless the interaction is simple (e.g., fixed grammar), human-computer systems require dialogue management. The basic function of dialogue management is to translate user requests into a language the computer understands and the system's output into a language that the user understands (Goren-Bar, 2001). In addition, dialogue management must be capable of performing a variety of tasks including disambiguation, error handling, and role switching (Abella, Gorin, 1999).

Role switching occurs because at any stage in a dialogue, one participant has the initiative (control) of the conversation. In a sense, initiative is a function of the roles of the participants. Dialogue systems may allow the user or the computer to take the initiative, or may allow both to switch roles as required. By far, the hardest dialogues to model are those in which the initiative can be taken at any point in the dialogue (Churcher et al., 1997).

2.3 User Model

Dialogue cannot make sense unless the user and the system have a reasonable understanding of each other. Given a user model, dialogue adaptation can be performed by referring to a user's expertise, knowledge, and preferences (Goren-Bar, 2001). For example, the way in which information is collected (filtering, classification, etc.) and presented (text, graphics, speech) can be adapted to the user.

The stereotype approach is the most popular user modeling method. With the stereotype approach, a designer defines appropriate subgroups of the user population (the stereotypes), identifies user behaviors that enable a system to categorize users into a subgroup, and represents the set of features (a "user profile") that characterizes each stereotype (Terveen, 1994).

3 System Design

3.1 Architecture

We have implemented collaborative control as a distributed set of modules, connected by a message-based architecture (Fig. 1). The architecture includes a safeguarded teleoperation controller that supports varying degrees of cooperation between the operator and robot (Fong et al., 2001b). Our primary user interface is the PdaDriver, which runs on WindowsCE-based PocketPC's (Fong et al., 2001a). PdaDriver provides a variety of command modes including rate, position and image/map waypoint control. We are currently using collaborative control to operate Pioneer mobile robots, which are equipped with a variety of sensors including ultrasonic sonar, color CCD camera, and differential GPS (Fong et al., 2001a).

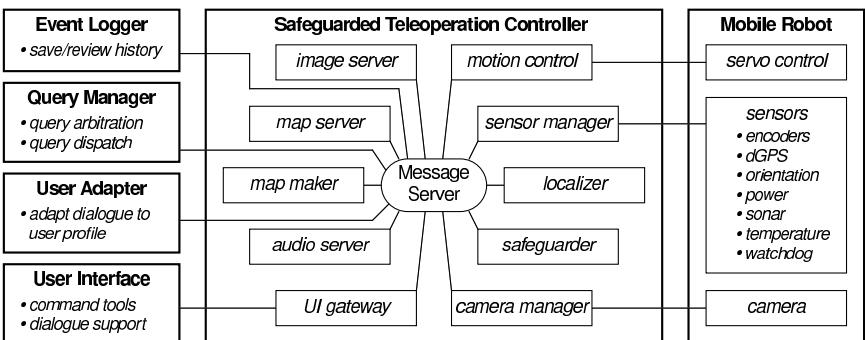


Fig. 1. Collaborative control architecture

3.2 User Model

Our collaborative control system manages dialogue with attribute-based, stereotype user profiles. We currently use three *user attributes* (accuracy, expertise, and query interval) to describe users, chosen because they are well-suited for vehicle teleoperation in unknown environments and because they provide a sufficiently rich basis for experimentation.

Accuracy estimates how accurately a user answers questions. If a user is highly accurate, then the robot can place greater confidence in responses to safety-critical questions. *Expertise* estimates the task skill (or domain knowledge) that the user possesses. This is valuable for adapting dialogue and autonomy. *Query interval* indicates how often the user can answer questions. Query interval may indicate: availability (amount of time a user can dedicate to questions), efficiency (how quickly he responds), and preference (how often he prefers to be interrupted by questions).

To support our initial experiments and evaluation of collaborative control, we defined three user stereotypes. A *novice* has no training or teleoperation experience

and is not expected to answer questions well. A *scientist* is also untrained and inexperienced, but is able to answer domain specific questions. An *expert* knows everything: he is experienced, understands how the system is designed, and can answer questions quickly and accurately.

For each user stereotype, we assigned the attribute values shown in Table 1. In our current system, all user attributes are constant (i.e., no adaptation occurs during use) and reflect *a priori* assumptions about each user stereotype (e.g., novices take longer to respond than experts).

Table 1. Stereotype attributes

Stereotype	Accuracy	Expertise	Query interval
Novice	30	30	60
Scientist	50	100	30
Expert	100	100	0

We use each user profile in three ways. First, we modify interaction by configuring the user interface for each type of user (e.g., selecting which control modes and displays are available). Second, we adapt dialogue by filtering messages: only messages that are appropriate for the user are selected. Finally, we modify how the robot acts by dynamically varying its autonomy.

3.3 Query Manager

Under collaborative control, multiple robot modules may ask questions at the same time. Thus, a collaborative control system needs query arbitration: a mechanism for choosing which questions to ask based on immediate (local) needs and overall (global) strategy. In our system, the QueryManager performs this task with an attribute-based filtering scheme (Fong, 2001).

Whenever a robot has a question to ask the human, it sends a message to the QueryManager. A message is defined by user attributes, query attributes (type, priority level, expiration time), and question-specific data (image, text, etc.) Our collaborative control system currently supports two query types: *y/n* (user must answer *y* or *n*) and *value* (user must provide a numeric value).

The QueryManager stores incoming messages into a cache. When the human indicates that he is available to answer a question, the QueryManager selects a message by filtering the cache. Because the cache is priority-sorted, urgent questions have preference. Expired questions are discarded undelivered (i.e., the user is never asked a question which is no longer valid).

3.4 Dialogue

In our system, dialogue arises from an exchange of messages between human and robot. Effective dialogue does not require a full language, merely one that is pertinent to the task at hand and that efficiently conveys information. Thus, we do not use natural language and we limit the vocabulary and grammar to vehicle mobility issues (navigation, obstacle avoidance, etc.).

Table 2. Robot-to-user queries

Query	Type	Accuracy	Expertise
Can I drive through (<i>image</i>)?	y/n	50	50
Is this a rock (<i>image</i>)? If you answer y, I will stay here.	y/n	0	50
The environment is very cluttered (<i>map</i>). What is the fastest I should translate?	value	0-100	50
My motors are stalled. Can you come over and help?	y/n	0	0
Motion control is currently turned off. Shall I enable it?	y/n	50	0
Safeguards are currently turned off. Shall I enable it?	y/n	50	0
Stopped due to collision danger. Disable safeguards?	y/n	100	0
Stopped due to high temperature. What should the safety level be?	value	0-100	0
Stopped due to low power. What should the safety level be?	value	0-100	0
Stopped due to rollover danger. Can you come over and help?	y/n	0	0

Table 2 lists the queries that a robot can ask. Two queries have variable accuracy levels because the importance of these questions can change with time or situation. Low accuracy values means that the robot is willing to accept any response. High accuracy values, however, indicate that the setting is critical to the robot's continued health.

Three of the queries in Table 2 have non-zero expertise values. To answer these queries, the human must have a certain level of expertise. For our experiments, we did not distinguish between different types of experts (e.g., skilled pilot vs. geologist). In practice, however, we would use additional attributes to target queries to specific task or domain expertise.

4 Results

To gain insight into collaborative control, we examined three vehicle teleoperation scenarios. In each case, we found that collaborative control enabled the robot to perform better. In particular, we observed that when the robot is operating poorly, or when it does not know what to do, a simple human answer can lead to a significant improvement.

4.1 “A to B”

Perhaps the most basic task in vehicle teleoperation is “A to B”: controlling the robot so that it moves from point A to B. As simple as this may seem, successful execution is critical to many applications. In reconnaissance, for example, performance is often governed by how well the robot moves from point to point. Thus, we need to make “A to B” as successful as possible (Fong et al., 2001b).

The most effective command mechanism currently used in vehicle teleoperation is waypoint driving: the operator specifies a series of points that must be passed *en route* to a target position. Waypoint driving has many advantages over direct (manual) control, e.g., it can tolerate significant delay. However, waypoint driving is not without problems. For example, if the robot has trouble deciding if an obstacle is in the way, it may operate slowly or be forced to stop. Collaborative control remedies this by allowing the robot to confer with the human before proceeding.

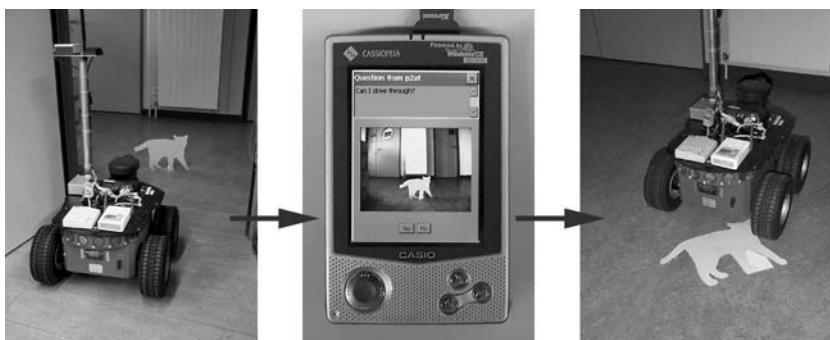


Fig. 2. Query to the human: “Can I drive through?”

Figure 2 shows an example of this interaction occurring in an experiment we performed in an office environment. During this test, we placed a cardboard cat in the path of the robot. The cardboard cat was detected as an obstacle by the robot’s sonar, thus forcing it to stop. At this point, the robot sent a camera image to the human and asked if it was safe to drive forward. When the human answered “yes” (based on his interpretation of the image), the robot was able to proceed without delay.

4.2 Collaborative Exploration

Although considerable research has focused on using human and robotic systems for planetary exploration, scant attention has been given to developing joint human-robotic systems. Yet, such systems offer significant potential to improve planetary missions by increasing productivity while reducing cost, particularly for surface operations such as material transport, survey and sampling (Fong et al., 2001a).

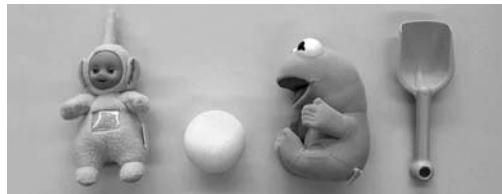


Fig. 3. Some “rocks”

To examine collaborative human-robot exploration, we developed a perception module, *RockFinder*, that autonomously locates “rocks”. Our intent was to study the assistance an exploration rover might provide. Thus, rather than examine morphology, RockFinder simply searches for objects (Fig. 3) that match a color signature. Whenever RockFinder detects a potential “rock”, it converses with the human to decide what to do.



Fig. 4. Collaborative exploration

Figure 4 shows an example of this interaction. With human and robot collaborating, exploration becomes very efficient: the human can say, “Tell me when you find something interesting” and the robot can do the search even if its autonomy (e.g., RockFinder) is limited.

4.3 Multi-Robot Teleoperation

The American military is currently developing mobile robots to support future combat systems. These robots will be used to perform a variety of reconnaissance, surveillance and target acquisition (RSTA) tasks. Because these tasks have traditionally required significant human resources (manpower, time, etc.), one of the primary areas of interest is determining how a small number of operators can control a larger number of robots. We believe that collaborative control provides an effective solution for this problem.

For example, consider the situation in which a single operator needs to control multiple robots, each of which is capable of limited autonomous RSTA functions (e.g., “move to point Tango and collect imagery”). As they traverse unknown, unexplored or changing terrain each robot will likely have questions such as: “Is it safe to continue driving at this power level?”, “Is this obstacle dangerous?”, and “Is it safe to drive over this terrain?”.

Since the human can only focus his attention on one robot at a time, we can use dialogue to unify and coordinate the multiple requests. Specifically, we arbitrate among the questions so that the human is always presented with the one that is most urgent (in terms of safety, timeliness, etc). This helps to reduce the level of attention and control the operator must dedicate to each robot. Consequently, the human can more effectively perform simultaneous, parallel control. In addition, because each robot is aware that the human may not be able to respond, it can still try to resolve problem on its own.

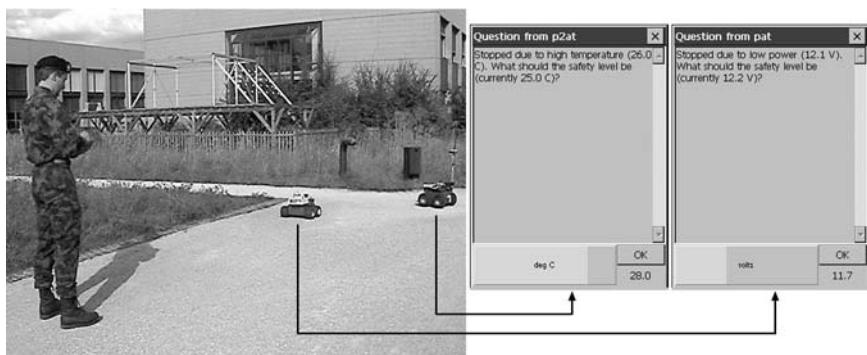


Fig. 5. Multi-robot teleoperation

Figure 5 shows an example of this behavior occurring in multi-robot teleoperation. In this experiment, an operator is using two robots for reconnaissance. Collaborative control allows the human to quickly switch his attention between the two, directing and answering questions from each as needed. In our testing, we found that this to be an effective way to interact with independently operating robots. In particular, coordination arises naturally from query arbitration.

5 Discussion

5.1 Benefits

By enabling humans and robots to work as partners, we have found that teleoperation is easier to use and more productive. Collaboration enables the human and robot to complement each other, as well as allowing the robot to proceed when the human is unavailable. Dialogue lets us build systems that are user adaptive and that encourage teamwork.

We have observed that dialogue makes human-robot interaction adaptable. Since the robot is “aware” of to whom is speaking, it can dynamically decide whether or not asking a question to the human is worthwhile. We have also found that there are situations for which dialogue enables the robot to perform significantly better than without human intervention. Moreover, this is true regardless of whether the human is a novice or an expert. In other words, even a novice can help compensate for inadequate sensing and autonomy.

Lastly, it seems evident (though we have not yet confirmed this) that specific combinations of collaboration and dialogue are appropriate for multiple situations. In other words, it may possible that the interaction used for a specific user could be appropriate for all users when the system is constrained by factors such as bandwidth and delay.

5.2 Limitations

Although collaboration and dialogue provide significant benefits, we recognize that there are limitations. First, identifying which parameters are well-suited to a given task and assigning appropriate values for each query is difficult. If there are many tasks to perform or if task execution creates many questions, then dialogue may add considerable complexity to system design.

Second, if human-robot interaction is adaptive, then the flow of control and information through the system will vary with time and situation. This may make debugging, validation, and verification harder because it becomes more difficult (though not impossible) to precisely identify an error condition or to duplicate a failure situation.

Finally, working in collaboration requires that each partner trust and understand the other. To do this, each collaborator needs to have an accurate model of what the other is capable of and of how he will carry out a given assignment. If the model is inaccurate, or if the partner cannot be expected to perform correctly (e.g., a novice answering a safety critical question), then care must be taken. For the robot, this means that it may need to weigh human responses instead of accepting them at face value. For the human, this means the robot may not always behave as expected.

6 Related Work

6.1 Human-Robot Interaction

Human-robot interaction (HRI) can be defined as *the study of humans, robots, and the ways they influence each other*. Sheridan notes that one of the challenges for HRI is to provide humans and robots with models of each other (Sheridan, 1997). In particular, he claims that the ideal would be analogous to two people who know each other well and who can pick up subtle cues from one another (e.g., musicians playing a duet).

In recent years, much effort has focused on developing robots that work directly with humans, as assistants or teammates (Baltus et al., 2000; Laengle et al., 1997; Nourbakhsh et al., 1999). The problem with most of these human-robot systems is that they are limited by their user-centric designs. Since effective collaboration requires an exchange and sharing of information between parties, collaborative control considers both user and robot needs.

6.2 Human-Robot Control Architectures

Numerous robot control architectures have addressed the problem of mixing humans with robots. The two most common approaches are prioritized control (Albus et al., 1987) and command fusion (arbitration) (Rosenblatt, 1997). In such systems, the human is limited to providing command input to the system. Collaborative control, however, relaxes this restriction and allows the human to also contribute high-level planning or perception input to robot modules.

Adjustable autonomy and mixed initiative systems have recently received considerable research attention (Bonasso, 1999; Myers, Morley, 2001). Although both approaches share some aspects of collaborative control, neither completely addresses the idea of peer interaction between humans and robots. The distinctive feature of collaborative control is that it uses human-robot dialogue as a mechanism for adaptation and a framework for coordination.

6.3 Human-Computer Collaboration

There are two major approaches to human-computer collaboration (Terveen, 1994). Human Emulation (HE) assumes that the way to get computers to collaborate with humans is to endow them with human-like abilities, to enable them to act like humans. Human Complementary (HC) assumes that computers and humans have fundamentally asymmetric abilities. Thus, the focus of HC is to develop techniques that make the computer a more intelligent partner.

Collaborative control is a hybrid approach that exhibits characteristics of both HE and HC. Collaborative control is HE-like because it emphasizes human-robot dialogue and because it includes techniques to model and adapt behavior to different users. Collaborative control is HC-like because it recognizes that humans and robots have different capabilities, skills, and needs. As a consequence, it provides different types of support to each.

Acknowledgements.

We would like to thank the EPFL Institut de production et robotique for providing research facilities and infrastructure. This work was partially supported by a grant from SAIC and the DARPA ITO Mobile Autonomous Robot Software (MARS) program.

References

- Abella A, Gorin A (1999) Construct algebra: analytical dialog management. In Proc Annual Meeting of the ACL.
- Albus J et al. (1987) NASA/NBS standard reference model for telerobot control system architecture (NASREM). Technical note, NIST.
- Baltus G et al. (2000) Towards personal service robots for the elderly. In Proc WIRED. Carnegie Mellon University, Pittsburgh.
- Bonasso P (1999) Issues in providing adjustable autonomy in the 3T architecture. In Proc Spring Symposium on Agents with Adjustable Autonomy, AIAA, Menlo Park.
- Churcher G et al. (1997) Dialogue management systems:a survey and overview. Report 97.6, School of Computer Studies, University of Leeds.
- Clarke R (1994) Asimov's laws of robotics: implications for information technology. *IEEE Computer*, 26(12), 27(1).
- Fong T (2001) Collaborative control: a robot-centric model for vehicle teleoperation. PhD thesis, Carnegie Mellon University, Pittsburgh. CMU-RI-TR-01-34.
- Fong T et al. (2001a) A personal user interface for collaborative human-robot exploration. In Proc iSAIRAS, Canadian Space Agency, Ottawa.
- Fong T et al. (2001b) A safeguarded teleoperation controller. In Proc Intl Conf on Advanced Robotics, IEEE, Piscataway.
- Goren-Bar D (2001) Designing model-based intelligent dialogue systems. In Information Modeling in the New Millennium. Idea Group, London.
- Laengle T et al. (1997) Cooperation in human-robot-teams. In Proc Intl Conf on Informatics and Control, Russian Academy of Science, Moscow.
- Lansdale M, Ormerod T (1994) Understanding interfaces: a handbook of human-computer dialogue. Academic Press, London.
- Milgram P et al. (1993) Applications of augmented reality for human-robot communication. In Proc Intl Conf on Intelligent Robots and Systems, IEEE, Piscataway.
- Myers K, Morley D (2001) Human directability of agents. In Proc. Intl Conf on Knowledge Capture, ACM, New York.
- Nourbakhsh I et al. (1999) An affective mobile robot educator with a full-time job. *Artificial Intelligence*, 114(1–2).
- Rosenblatt J (1997) DAMN: A distributed architecture for mobile navigation. PhD thesis, Carnegie Mellon University, Pittsburgh. CMU-RI-TR-97-01.
- Sheridan T (1997) Eight ultimate challenges of human-robot communication. In Proc ROMAN, IEEE, Piscataway.
- Terveen L (1994) An overview of human-computer collaboration. *Knowledge-Based Systems*, 8(2–3).

Part 6

Applications

Session Summary

Paolo Dario
Chair

Scuola Superiore Sant'Anna Pisa, Italy

Although the title of this Session looks rather “generic”, the papers that are included in the Session have a very important aspect in common: they represent the “ubiquity” of modern robotics. Robotics is intrinsically multidisciplinary. It puts together issues related to mechanical and electronic hardware, to software, to power supply, and to human-machine interfaces. Robotics is today the modern paradigm of machine design, intended as the design of the multitude of technological devices which take part in our lives. Every device is a system that comprises of basic components, designed by an integrated approach by which the design of each component takes into account the interactions with the others. In a very basic scheme of a robotic system, and of most current devices, we can identify a sensing part, a processing part and an action part, which together determine the device behavior, based on external stimuli. This includes the interaction with the user, that can rely on different communication channels, from purposive computer interfaces, to physical tools and gesture/verbal languages.

In this sense, robotics is becoming “pervasive” into virtually all devices and even in many other disciplines. The robot is not only a manipulator, but a concept (actually theory, models and technologies) that is hidden in a number of applications, and made transparent to the user. For the same reason, the potential of robotics applications is becoming unlimited, and is going to include fields that range from the “traditional” industrial manufacturing to services and to most everyday life activities.

Based on these concepts, whereas some sessions of this book focus on theory and technology of specific components and subsystems of robots, and some other sessions address applications of robots in some specific areas, the Session on Applications shows how robotics-derived methods and solutions (not simply the “manipulator”) can be used to solve different problems in different areas. The four papers reflects the variety of robotics applications, in the wide sense illustrated above. They address different aspects of robotics and different application areas.

The first paper, “Vertebrate-type perception and gaze control for road vehicles”, by M. Pelkofer, M. Lutzerland and D. Dickmanns, mostly investigates the part of sensing and perception of the typical scheme of robotic system. The paper presents a third generation dynamic vision system, called “expectation-based, multi-focal, saccadic vision”. The authors’ approach is based on the imitation of some aspects of the vision process of humans. Their final goal and achievement is a closed-loop system relying on a 4-camera vision sensor for planning and executing human-like eye movements, i.e. smooth pursuits and saccades. The mechanism of coordination between the visual sensor and the head movements takes advantage of the assembly

of functions that are found in the eye of vertebrates. The application context in which the system has been experimentally validated is autonomous navigation of vehicles.

The second paper, “Towards a realistic medical simulator using virtual environments and haptic interaction”, by C. Laugier, C.A. Mendoza, K. Sundaraj, addresses a different application domain, i.e. the biomedical field, that is becoming increasingly popular for robotics researchers and important for the society and for industry. The ultimate goal being to use robotics to increase the accuracy of intervention, here models are exploited in the simulation of human tissues. The importance of very accurate knowledge of the viscoelastic properties of tissues and the capability of modeling them is pointed out in the paper, along with the need for developing interfaces between the system and the human operator, which is based – in the case of the paper – on a haptic device.

The third paper addresses more specifically the problem of human-robot interaction, by presenting an application of verbal communication in the case of a robotic office assistant. The paper is entitled “Spoken language interface of the Jijo-2 office robot for map learning and office information service”, by T. Matsui, H. Asoh, F. Asano, J. Fry, T. Kurita, I. Hara, Y. Motomura and K. Itou, and presents how verbal communication with humans can help a mobile robot navigate in a partially-know environment.

The fourth paper, “Intelligent home appliances”, by H. Christensen, synthesizes the concept of “robotics which disappears”, that is robotics pervading common devices of everyday life activities, with special reference to the needs of the aging society. Methods derived from robotics are applied to the problems of control, manipulation, navigation, and software engineering. The paper points out how the field of “domotics” can become extremely popular and lead to very valuable industrial applications, provided that the robotics research community is capable of providing acceptable solutions to the challenging technical problems posed by the specific requirements of a new generation of home appliances, as well as to work together with other communities (for example, industrial designers).

The variety of robotics-related aspects which were covered in the presentations raised a number of important and interesting questions, and generated an animated discussion in the audience.

Vertebrate-type Perception and Gaze Control for Road Vehicles

M. Pellkofer¹, M. Lützeler², and E. D. Dickmanns³

¹ Institut für Systemdynamik und Flugmechanik, Universität der Bundeswehr München (UBM), 85577 Neubiberg, Germany, martin.pellkofer@unibw-muenchen.de, phone +49 89 6004-3581

² Institut für Systemdynamik und Flugmechanik, Universität der Bundeswehr München (UBM), 85577 Neubiberg, Germany, michael.luetzeler@bigfoot.com

³ Institut für Systemdynamik und Flugmechanik, Universität der Bundeswehr München (UBM), 85577 Neubiberg, Germany, ernst.dickmanns@unibw-muenchen.de, phone +49 89 6004-4151

Abstract. In the Expectation-based, Multi-focal, Saccadic (EMS-) Vision system of UBM, the vision sensor consists of four cameras with different focal lengths mounted on a highly dynamic pan-tilt camera head. Image processing, gaze control and behavior decision interact with each other in a closed loop form. The behavior decision module specifies the relevance of obstacles like road segments, crossings or landmarks in the situation context. The gaze control unit takes all this information in order to plan, optimize and perform a sequence of smooth pursuits, interrupted by saccades. The sequence with the best information gain is performed. The information gain depends on the relevance of objects or object parts, the duration of smooth pursuit maneuvers, the quality of perception and the number of saccades. The functioning of the EMS-Vision system is demonstrated in a complex and scalable autonomous mission with the UBM test vehicle VAMORS.

1 Introduction

Technical vision systems for practically useful real-time guidance of ground vehicles are in their second decade of development. Early quasi-static image evaluation systems for this purpose (Gennery, 1977; Nilson, 1969) date back some more years. In 1986, the use of CCD cameras with digital microprocessor systems onboard standard sized test vehicles for road driving started both in the USA and in Europe. While driving slowly and cross-country has been the goal in the DARPA-funded US-approach (Davis, Kushner, LeMoigne, Waxman, 1986; Kuan, Phipps, Hsueh, 1986; Thorpe, Hebert, Kanada, Shafer, 1988), in Europe driving on well structured roads at high speeds has been favored right from the beginning (Dickmanns, Zapp, 1986; Zimdahl, Rackow, Wilm, 1986). The 'Prometheus' project in the European 'EUREKA' framework has advanced vision for road vehicles considerably in the years 1987–94. Most researchers and developers selected a single camera fixed to the car body, for simplicity reasons (see proceedings of the Symposium on 'Intelligent Vehicles' since 1992). At UBM, rather early a combination of two cameras with different focal lengths fixedly mounted relative to each other on a pointing platform has been selected for a good combination of a larger field of view nearby

and good resolution in a smaller area further away. Good, but not fully satisfying results for the long run have been achieved (Dickmanns, Behringer, Dickmanns, Hildebrandt, Maurer, Tho-manek, Schiehlen, 1994). This has led to the development of a third-generation dynamic vision system since 1996, called 'Expectation-based, Multi-focal, Saccadic' (EMS-)Vision (Gregor, Lützeler, Pellkofer, Siedersberger, Dickmanns, 2000). Results with this new vision system are described in this paper.

The paper is organized as follows: Section 2 gives a motivation for the interaction mechanism developed between perception and gaze control. Section 3 describes the sensor concept of our vision system. The components of the gaze control system follow in section 4. Sections 5 and 6 explain how the different perception experts – especially for road recognition – specify their gaze requirements using regions of attention. Section 7 describes the algorithm for the optimization of the viewing direction running in the gaze control unit. In section 8 experimental results of an autonomous turn-off maneuver are presented.

2 Motivation

For driving a vehicle in different domains, many objects of different size, position and orientation relative to the own vehicle are to be recognized. For example, driving at high speeds requires large look-ahead distances on the trajectory planned in order to detect obstacles sufficiently early for collision avoidance. In cluttered environments with many subjects moving in an unpredictable manner, a wide field of view (f.o.v.) is required for collision avoidance. The capability of stereo interpretation helps in the near range to understand the spatial arrangement and motion of objects quickly. Additionally, on uneven and rough terrain, inertial stabilization of the viewing direction is necessary in order to reduce motion blur in the images (especially the tele-ones).

One approach for perceiving all relevant obstacles for autonomous driving is to mount numerous cameras with different optical properties and orientation on the vehicle. But this procedure is expensive, inflexible, error-prone and imposes the communication and handling of a huge array of image streams. Another approach is to imitate the vision process of the human driver. Human beings are able to change their viewing direction in a very quick and complex manner. Thereby, periods of smooth pursuit are interrupted by quick changes of viewing direction, so-called saccades. Humans carry out between 3 – 5 saccades in one second. This viewing behavior represents a highly dynamic perception-action-cycle. Saccades are triggered by optical stimuli or by intention. The field of research 'active vision' tries to transfer such complex viewing behavior to technical systems (Aloimonos, 1993; Ballard, Brown, 1993). The factors influencing gaze control in autonomous vehicles are summarized in (Pellkofer, Dickmanns, 2000). In our EMS-Vision system, we favor the last approach and, therefore, we have designed a sensor concept taking advantage of the assembly of functions which nature combined into the vertebrate eye.

The UBM perception modules, e.g. for road- or vehicle detection, are based on the 4D-approach to dynamic machine vision. A generic model of relevant objects in the 3D world is used with free parameters for adaptation to the actually encountered geometry and motion. Position and geometry parameters of the objects of interest are determined from image sequences originating from the camera configuration. More information about the 4D-approach can be found in (Dickmanns, Wünsche, 1999). An overview on the architecture of the EMS-Vision system is given in (Gregor et al., 2000).

3 Camera Configuration

The key sensor of the EMS-Vision system is the multi-focal active/reactive vehicle eye, called MARVEYE. The MARVEYE camera arrangement combines a wide f.o.v. nearby with central areas of high resolution (Dickmanns, 2000). MARVEYE consists of up to four cameras with three different focal lengths (see Fig. 1). Two cameras equipped with wide-angle lenses are set up as a horizontal stereo pair with parallel or skewed optical axes. The skewed arrangement is used for tight maneuvering on low order roads and early detection of overtaking vehicles. The parallel setup is utilized for vertical curvature recognition and obstacle detection as well as for tracking in off-road scenarios using stereo processing. On the third and fourth camera mild and strong tele-lenses are mounted. The image resolution of the tele cameras is 3 to 4 resp. 10 times higher than the resolution of the wide-angle cameras. The high resolution camera images and the corresponding different look-ahead distances¹ are useful for identifying new objects, estimating object states and parameters with high precision, e.g. road curvature estimation and landmark navigation. The mild tele camera is a 3-chip color camera, the others are monochrome cameras.

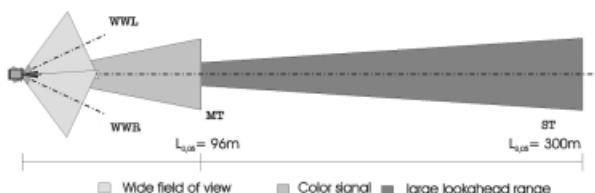


Fig. 1. MARVEYE camera configuration

¹ The look-ahead distances $L_{0,05}$ are the distances at which one pixel corresponds to five centimeters in real world perpendicular to the optical axis of the corresponding camera.

4 Gaze Control Unit

A core element of the EMS-Vision system is the active gaze control unit consisting of the active camera head with its embedded controller running on real-time hardware with a cycle time of 2 ms, the server process Gaze Control (GC) and a Behavior Decision module for Gaze & Attention (BDGA). In the sequel, the parts of the gaze control unit are described.

4.1 Active Camera Head

The MARVEYE camera configuration is mounted on a pan-tilt camera head so that the gaze control unit of the EMS-Vision system can point the tele cameras to an arbitrary part of the front hemisphere. The active camera head is mounted in UBM's experimental vehicles VAMORS and VAMP behind the windscreen. The camera head supports fast gaze shifts, soft turns for smooth pursuit of a moving target and inertial stabilization to compensate vehicle body motion in the frequency range of 0.7 – 2 Hz. The control software of the active camera head runs on an embedded transputer system (in VAMORS) or on an industrial PC (in VAMP). The embedded system is connected to the 'Behavior PC' via transputer link resp. CAN-link. The performance of the active camera head is evaluated in (Pellkofer, Dickmanns, 2000).

4.2 Gaze Control

The server process Gaze Control (GC) is the executive part of the gaze control unit. It communicates with the embedded controller of the active camera head, performs gaze maneuvers, monitors the performance of active gaze maneuvers and updates state and status of the camera head in the scene tree. GC offers the following gaze maneuvers:

- With a smooth pursuit the angular motion between a camera and a physical object can be compensated, so that the object remains visible in the camera image.
- With a saccade, a physical object can be centered in the f.o.v. of a certain camera. This is favorable for getting high resolution images from objects far away.
- With a search path a systematic search for new objects in a certain 3D area can be performed.

4.3 Behavior Decision for Gaze & Attention (BDGA)

BDGA is the planning part of the gaze control unit. It consists of four submodules: Situation Assessment for Gaze & Attention (SAGA), Visibility Analysis for Gaze & Attention (VAGA), Optimization of Viewing Behavior (OVB) and Perception Control (PerC).

SAGA uses the current and next mission elements in conjunction with domain information to add symbolic attributes and meanings to the object representation.

This symbolic information is stored in the situation representation and must be taken into account for the optimization of viewing behavior. For this purpose, the attention demand, a variable within the interval [0,1], is computed by using symbolic information. The attention demand specifies the relevance of an object for gaze: An object with an attention demand of 1 will be regarded exclusively, an object with 0 will be ignored. Objects of the same class can have different meanings due to the current mission context and the locomotion task. For example, two road crossings can simultaneously appear in the scene representation, but only one has a special meaning with respect to a planned turning off. While driving on highways, symbolic attributes of vehicles may affect the viewing direction, for example the attributes '*leisurely driven vehicle*' or '*aggressively driven vehicle*', which correspond to the frequency of lane crossings and the distances being kept to other vehicles.

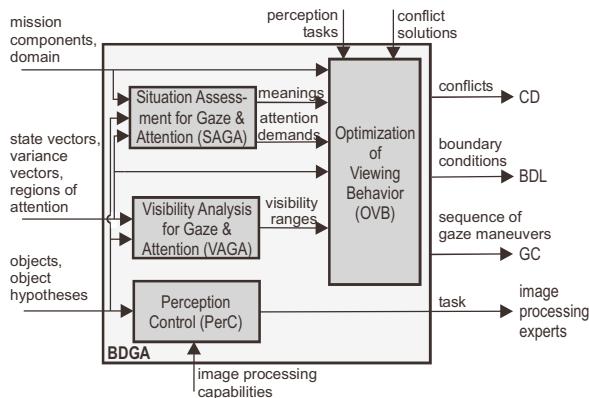


Fig. 2. Behavior Decision for Gaze & Attention

As described in section 6 the perception experts specify regions of attention (RoAs) for each perception object. VAGA uses these RoAs to calculate visibility ranges for all objects. Section 8 gives a definition of a visibility range and describes the algorithm running in the submodule VAGA.

If several objects have to be perceived and not all RoAs can be centered in the camera images in parallel, OVB calculates a sequence of smooth pursuits linked by saccades to satisfy the needs of all perception experts. How this is done is described in section 7. For task assignment, PerC evaluates the contents of the dynamic knowledge representation in order to find the appropriate perception expert for each object to be perceived; it then sends a perception task to the expert specified for this class of objects.

5 Requesting Attention

An autonomous agent must perceive its environment to perform the given task safely and intelligently. Different object classes may be encountered which help to complete the mission successfully, like roads or landmarks. On the other hand, obstacles or other vehicles may obstruct the desired path and must be avoided.

Within the framework of EMS-Vision, the perception tasks are distributed in a twofold manner. Dedicated perception modules are specialized for different object classes, e.g. vehicles (Hofmann, Rieder, Dickmanns, 2000) or roads. The implementation of the EMS-Vision-system on a computer network as shown in section 4 allocates the perception experts on the processing node where the most relevant video signal is available. For each physical object to be recognized, the gaze control module needs an indication for the optimal viewing direction. Additionally, some perception tasks may only be accomplished using a specific camera signal, e.g. color processing requires a color camera signal or detecting a landmark at a large look-ahead distance relies on evaluating the images of the tele-camera.

All perception modules are based on the 4D-approach (Dickmanns, Wünsche, 1999) to dynamic machine vision. Modeling of physical objects in 3D-space and time, applying appropriate dynamical models, allows to compensate for lacking measurement updates for short periods of time. Exploiting this approach, multiple objects not simultaneously visible in the fields of view of the cameras may be perceived by sequentially focussing on each object. Lacking measurement update or low resolution measurements will lead to increasing estimation errors, whereas ‘optimal’ high resolution measurements will diminish estimation errors.

This knowledge on how to recognize an object is the expertise of the perception module. A common interface towards the gaze control unit must be designed to allow various perception tasks to specify their requirements for viewing direction.

Our initial approach specified a fixation point on the object’s geometry in 3D-space (Lützeler, Baten, 2000). Although this resulted in simple and straight forward control laws for viewing direction, it did not reveal the relevant object parts for image processing to the gaze control expert. Thus finding compromises with respect to different objects requiring measurement updates proved difficult, as no indication was given, on how far the viewing direction may deviate from this optimal point and still be satisfactory for image processing.

5.1 Regions of Attention

In order to determine optimal gaze for the recognition of an object, the geometrical description of object shape is not sufficient, as it gives no indication which object characteristics form features for the sensor and the measurement method used. To solve this task, the perception expert itself communicates its needs by specifying regions of attention (RoA) and a two layer hierarchical logic, combining these. The regions are given in 3D object coordinates covering relevant object parts that should be visible in the camera f.o.v. If the vision system comprises multiple cameras, the desired camera in which the RoAs should be visible must be specified as well.

Different RoAs may be combined in groups by Boolean operations OR and AND (first level). These groups are linked on the second level by an OR-operator; the order of appearance in the second level term states the relevance of each group, starting with the most important group with the highest information gain. To every combination of RoAs a measure is assigned, describing the information gain relative to the best combination. The best combination describes the optimal way to image the object exclusively and has a relative information gain of 1. All other combinations lead to lower information gain being in the interval [0,1]. The actual viewing direction can then be computed by the BDGA process as shown in the sequel in an optimal manner for all perception tasks, possibly spanning multiple objects and multiple perception experts involved.

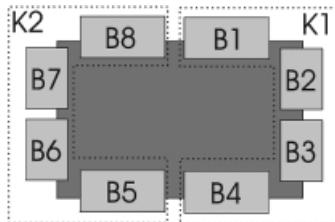


Fig. 3. Object model with 6 RoAs

Figure 3 shows a simple rectangular object with eight RoAs, marked B1...B8. Regions on the left and right hand side are grouped together respectively, K1 and K2. The adjoining logic would express that the object can be recognized in an optimal manner if all regions are within the camera f.o.v. Localization is still possible if the left or right group is visible only. Additionally, this representation conveys the information that the measurement method used (edge-detection) relies on the visibility of the object boundaries. Using a narrow f.o.v. camera, the object may be larger than the f.o.v. and a fixation on the object's center would give no useful measurements.

6 Active Vision for Road Recognition

In this section, details on road recognition are presented. The road recognition module applies the same geometry model for a road segment and the same visual detection and tracking methods to two distinct perception tasks that occur while performing a turn-off maneuver. The first perception task is to track a road while the vehicle is following this segment. The second task is to determine pose and shape of a branching road at the intersection. For details on the coordinate systems which are used for describing the position of road segments relative to the own vehicle see (Lützeler, Dickmanns, 1998). The geometric road model is based on the description in (Dickmanns, 1988; Lützeler, Baten, 2000).

6.1 Edge-based Measurement Model

From the parametric shape model, measurement points on the object boundary are computed. Their projections into the image plane represent the centers of search windows for edge extraction. Due to the possible curvature of the road segment, a single rectangular box spanning the entire road boundary would cover large image areas not used for image processing. Thus, on each side the length of the segment is divided into three RoAs, marked $B1 \dots B6$ in Fig. 4. The width of the areas, marked Tol , corresponds to the image window size used for edge extraction. The distance l_{start} between the road segment's origin (x_r, y_r) and the beginning of the first two RoAs accounts for the area not used for image processing. This area is neglected:

- if the road segment starts at the center of gravity of the vehicle and is thus underneath the vehicle body, or
- at intersections, where the start of a branching segment is within the area of the crossing.

The combinations of RoAs are given in (1) – (4). The first group $G1$ combines all regions, specifying the request to depict all areas simultaneously (see (1)). Each of the three groups in (2) combines two RoAs, one from each road side. If measurements on both sides are performed, the width of the road can be estimated.

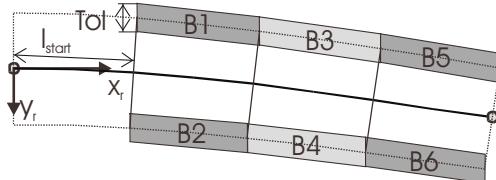


Fig. 4. Edge-based measurement model

The groups in (3) each contain two RoAs on one road side thus conveying to depict a certain portion of the road boundary. The first two groups G_5 and G_6 are on the right hand side as in continental Europe driving is performed on the right side. The ninth term $G9$ states the minimal requirement of depicting at least one RoA, (4). The second level feature entry FE combines the nine groups in descending order of importance.

$$G_1 = B_1 \wedge B_2 \wedge \dots \wedge B_6 \quad (1)$$

$$\begin{aligned} G_2 &= B_1 \wedge B_2 \\ G_3 &= B_3 \wedge B_4 \\ G_4 &= B_5 \wedge B_6 \end{aligned} \quad (2)$$

$$G_5 = B_2 \wedge B_4$$

$$G_6 = B_4 \wedge B_6$$

$$G_7 = B_1 \wedge B_3$$

$$G_8 = B_3 \wedge B_5$$

$$G_9 = B_1 \vee B_2 \vee \dots \vee B_6 \quad (4)$$

$$FE = G_1 \vee G_2 \vee G_3 \vee G_4 \quad (5)$$

6.2 Area-based Measurement Model

An area-based segmentation approach is used to determine the position of the skeleton line of the road segment from the video images. The segmentation is performed in one-dimensional cuts at dedicated image positions only, see the vertical markings, Fig. 10 (c) in the rightmost image, as an example. Successful segmentation is possible only if both road segment boundaries are visible in the f.o.v. of the camera. Due to this nature of the applied measurement technique, the regions of attention are built using points on both segment boundaries.

Figure 5 depicts a road segment with regions of attention, B1, B2 and B3. The distance between the two points on the boundaries is enlarged by a factor of 1.5 to ensure the visibility of the boundaries and some part of the road shoulder. With these RoAs four groups are built following the principles layed out for the edge-based measurement model.

As can be seen from Fig. 5, the RoAs are not equally distributed over the segment but cover only a subsection of the segment length L_{max} . This enables the system to shift the focus along the skeleton line depending on the distance between sensor (denoted by the coordinate system x_k, y_k) and road-segment (index r). If the vehicle is approaching an intersection and the branching road is still far away, the attention is focused on the start of the branching road segment, as the vehicle will acquire the branch here. As the distance to the road segment decreases, the RoAs are shifted along the skeleton line in a linear fashion. Within a certain ‘catch’ area around the origin of the segment, the focus lies on the center of the road at $\frac{L_{max}}{2}$.

$$G_1 = B_1 \wedge B_2 \wedge B_3 \quad (6)$$

$$G_2 = B_1 \wedge B_2$$

$$G_3 = B_2 \wedge B_3$$

$$G_4 = B_1 \vee B_2 \vee B_3 \quad (8)$$

$$FE = G_1 \vee G_2 \vee G_3 \vee G_4 \quad (9)$$

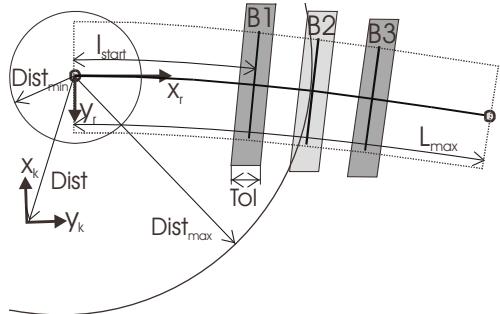


Fig. 5. RoAs while approaching a road segment

7 Optimization of Gaze Behavior

The optimization process of gaze behavior consists of two phases (see Fig. 8): the planning phase and the fixation phase. The aim of the planning phase is to find a compromise in form of a sequence of smooth pursuits linked by saccades, so that the perception requests of all perception experts in the system are satisfied. In the fixation phase, the compromise found is executed.

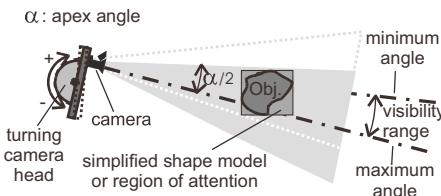


Fig. 6. Visibility range

As described in section 4, external objects are represented by scene nodes in the scene tree. For every object relevant for gaze control a so-called VAGA-object is generated (see Fig. 7). A VAGA-object references the IDs of the cameras in which the object should be imaged and contains the attention demand of the scene object calculated by the submodule SAGA (see section 4.3).

When no sequence is executed or the last sequence is done, the planning phase becomes active at the beginning of a cycle (see Fig. 8). In the first step of the planning phase, the attention demands of all VAGA-objects are scaled, so that the sum of all attention demands is 1. After that, the visibility range for every RoA specified by a perception expert is calculated. The visibility range is the angular range for both pan and tilt axis of the active camera head, in which the RoA is visible in the corresponding camera image (see Fig. 6). In the third step, the two layer hierarchical logic – also specified by the perception expert – is used to combine the visibility ranges of a VAGA-object on its own to find intersections. Every intersection found

is pushed to the list of visibility ranges of the VAGA-object (see Fig. 7). In this way, the information contained in the two layer hierarchical logic and the RoAs is processed and stored in the elements of the visibility range list and the list order. The higher up the element is within the list, the better the imaging described by the element is, considering a maximum of RoAs and information input. Thus, fixating the first visibility range in the list is the best imaging of this object alone, but allows a minimal variation of gaze only (minimal angular range).

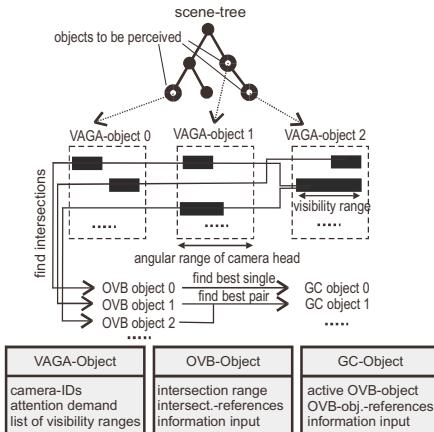


Fig. 7. Description of objects

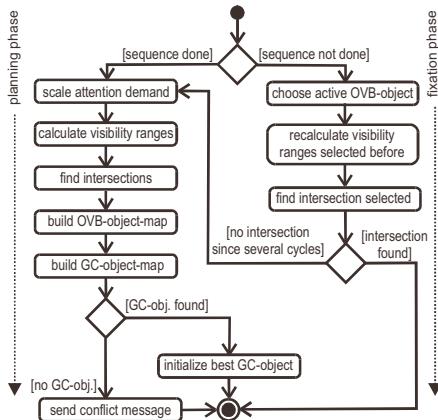


Fig. 8. Application flow

In the next step of the planning phase, the visibility ranges of all VAGA-objects are combined to find intersections. For every intersection found, an OVB-object is

generated and inserted in an OVB-object-map. An OVB-object contains the angular range of the intersection found, references to the visibility ranges of the VAGA-objects incorporated in the intersection and the information input gained by the combination of the RoAs. The information input of an OVB-object I_{OVB} is calculated according to the following equation:

$$I_{OVB}(k) = \sum_i^{n_{VAGA}(k)} (A_i \cdot \sigma_i^j) \quad (10)$$

where

$I_{OVB}(k)$: information input of OVB-object k

A_i : attention demand of VAGA-object i

σ_i : relative information gain of combination j of

VAGA-object i being considered.

$n_{VAGA}(k)$: number of VAGA-objects incorporated in OVB-object k

After building the OVB-object-map, a GC-object-map is generated by performing the following search:

- Search the OVB-object that considers all VAGA-objects and gains the highest information input. This object – called the ‘Best Single’ – is pushed as reference to a GC-object which is generated and inserted in the GC-object-map.
- Search a pair of OVB-objects which considers all VAGA-objects and gains the highest information input. This object – called the ‘Best Pair’ – is pushed as reference to a GC-object that is generated.
- If the ‘Best Single’ object does not exist, the ‘Best Triple’ referencing three OVB-objects is searched in the same way. If the ‘Best Single’ object exists, the search for best combinations is stopped.

GC-objects represent best combinations with different numbers of OVB-objects, like ‘Best Single’, ‘Best Pair’, ‘Best Triple’ and so on. The information input of a GC-object is calculated with the following equation:

$$I_{GC}(k) = \sum_i^{n_{OVB}(k)} (I_{OVB}^i \cdot T_{Fix}^i) \quad (11)$$

where

$I_{GC}(k)$: information input of GC-object k

I_{OVB}^i : information input of OVB-object i

$n_{OVB}(k)$: number of OVB-objects referenced in GC-object k

T_{Fix}^i : effective fixation time of OVB-object i

Thereby, the effective fixation time T_{Fix} is calculated with

$$T_{Fix}^i = \frac{I_{Komb}^i}{\sum_j^{n_{OVB}} I_{Komb}^j} \cdot (T - \sum_m^{n_{OVB}} T_{sac}^m) \quad (12)$$

where

T : planning period of BDGA (4 s)

T_{sac}^m : saccade duration from OVB-obj. m to $m + 1$

I_{Komb}^i : max. information gain of all combinations of ROAs in OVB-object i .

and

$$I_{Komb}^i = \text{Max}_{k \in K_i} \{A_n \cdot \sigma_n^k\}, \quad (13)$$

where

K_i : set of combinations of ROAs of VAGA-objects considered by OVB-object i .

The duration of a saccade is calculated according to the following linear approximation:

$$T_{sac} = T_{sac}^0 + \Delta T_{sac} \cdot \Delta A \quad (14)$$

where

T_{sac}^0 : initialization time of saccades (~ 170 ms)

ΔT_{sac} : saccade. duration increment per amplitude of saccade
(3.7 ms/deg).

ΔA : amplitude of saccade.

The parameters specified have been determined by evaluating the performance of the active camera head being used (Pellkofer, Dickmanns, 2000). After the search for the best combinations, the GC-object with the highest information input is initialized. A GC-object specifies a whole sequence of smooth pursuits interrupted by saccades. The number of saccades is $n_{OVB} - 1$ (without a possible saccade at the beginning of the sequence).

For a planning period of a few seconds, only a limited number of saccades are reasonable. A search for combinations of OVB-objects with more saccades is suppressed. If no GC-object can be generated with this maximum number of saccades

a conflict message is sent to CD (compare Fig. 2). In this case, CD must reduce the number of objects relevant for gaze control using the situation and mission context.

If a sequence is not finished at the beginning of a new cycle, BDGA performs the fixation phase (see Fig. 8). The first step in the fixation phase is to test whether the currently active OVB-object describing a smooth pursuit for a group of objects is to stop. A smooth pursuit is stopped, when its period of time is expired. In the case of a stop, the next OVB-object being referenced in the GC-object is initialized. If no OVB-object is left, the sequence is done and a new planning phase is initiated in the next cycle.

In the second step of the fixation phase the visibility ranges specified in the intersection reference of the active OVB-object are calculated. Using the updated visibility ranges, the intersection range of the OVB-object is recalculated. If the intersection still persists, the center of the intersection is fixated. If the intersection selected in the planning phase has disappeared over several cycles because of the movement of the scene objects, the sequence is aborted and a planning phase is initialized.

8 Experimental results

Autonomous turn-off maneuvers have been conducted with the experimental vehicle VAMORS on both unmarked campus roads and dirt roads on a military proving ground. Recognition of the intersection has been fully integrated into the mission context, controlling both locomotion of the vehicle and perception tasks. The viewing direction of the active pan-tilt camera head has been controlled dynamically by the gaze control unit as described above.

Figure 9 visualizes the results of the optimization process of viewing behavior. During road driving (before second 91 in Fig. 9) two objects are relevant: The **local road** segment (object-ID 2355) is to be imaged in the wide f.o.v. of the wide-angle cameras and the **distant road** segment (ID 2356) is to be imaged in the mild tele camera. As described in section 8, the submodule VAGA calculates the visibility range for every combination of RoAs specified in a two layer hierarchical logic. If the visibility range is not zero, it is pushed to the list of visibility ranges of the appropriate VAGA-object. Figures 9 (i) and 9 (j) show the first elements of the lists of visibility ranges of the **distant** and **local road** segment (pan angle). For every planning phase a dot appears in the plots.

Before second 91, a GC-object 0 containing the ‘Best Single’ solution and a GC-object 1 describing a ‘Best Pair’ solution can be found (see section 8 and Fig. 9 (c), 9 (d)). GC-object 0 gains a higher information input (~ 0.5) than GC-object 1 (~ 0.09); therefore GC-object 0 is performed. In Fig. 9 (a) the pan angle of the active camera head during the whole turn-off maneuver is shown.

After 91 seconds, an object hypothesis for the **cross-road** segment (ID 2359) is instantiated (see Fig. 9 (e)) so that three objects are relevant for gaze control and additional RoAs are specified. Figures 9 (g) and 9 (h) show the best and second-best

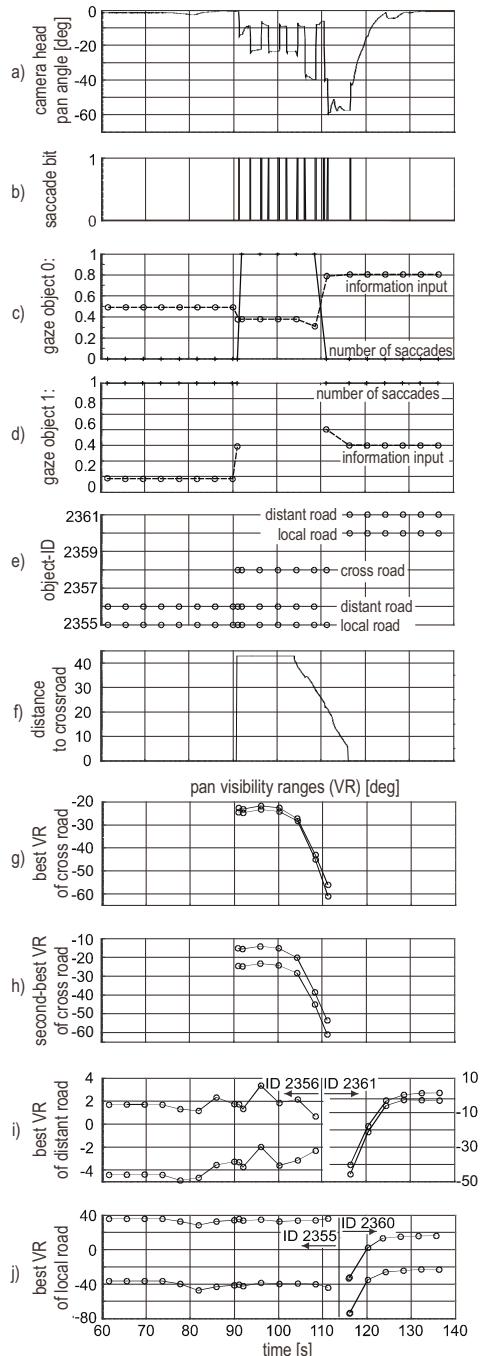


Fig. 9. Results of the optimization process

visibility ranges of the **cross-road** segment resulting from the combinations G1 and G2 of this object (see (6,7)).

In Fig. 9 (f) the distance between the vehicle and the **cross-road** segment is plotted. Approaching the intersection, the RoAs of the **cross-road** segment move along the skeleton line of the **cross-road** segment to improve the aspect ratio for perception (see section 7). Due to this, the pan angle of the visibility ranges of the **cross-road** segment and the amplitude of the saccades increase (see Fig. 9 (a), 9 (g), 9 (h)). According to Fig. 9 (c), the visibility ranges of the **distant road** and the **cross-road** segment do not overlap between second 91 and 109, so that no ‘Best Single’ solution can be found and GC-object 0 describes a gaze sequence with 1 saccade. In figure 9c the information input of this sequence can be seen also. During the smooth pursuit with a pan angle of about 10° the distant and the **local road** segment are imaged, and during the second smooth pursuit the **local road** and the **cross-road** segment are imaged. Figure 9 (b) shows the saccade bit denoting the execution of a saccade to all perception experts for measurement interrupts.

After second 109 the **distant road** segment is no longer visible and the **cross-road** and the **local road** segment are fixated only without performing a saccade. After second 116 the vehicle enters the intersection and a scene tree reorganization takes place. From this time on, the new **distant road** and **local road** segments (IDs 2360 and 2361) are relevant for gaze control. The best visibility ranges of these two objects intersect, a ‘Best Single’ solution exists and saccades are not necessary (see Fig. 9 (c)).

The top-most row of Fig. 10 shows the initial phase of the approach, the left & middle column show images from the two wide angle cameras, the right hand side from the color camera. Vertical search windows are applied at the expected position of the **cross-road** in the color image. The **local road** segment is tracked in the wide-angle cameras using horizontal search windows to extract road boundaries. Smooth pursuits with a panning angle of ~ 10° to the left, facilitating **cross-road** detection. This focusing has little effect on road detection near by, see Fig. 10 (b), 10 (c). The last set of images shows the new road being tracked after the turn-off maneuver has been completed.

9 Conclusion

For driving a vehicle in natural and unprepared environments many objects of different size, orientation and relative position to the vehicle are to be perceived. One way to handle the different perception circumstances is to mount many sensors with different optical properties and orientations onto the vehicle. But a different way has been chosen similar to vertebrate vision. Using an elaborately designed vehicle eye, called MARVEYE, and a high performance active camera head, the autonomous system adapts the sensor properties to the actual situation and to the perception task. The image processing modules in the system announce their gaze requirements in form of regions of attention and a two layer hierarchical logic. The gaze control unit calculates an optimal sequence of smooth pursuits linked by saccades to meet all

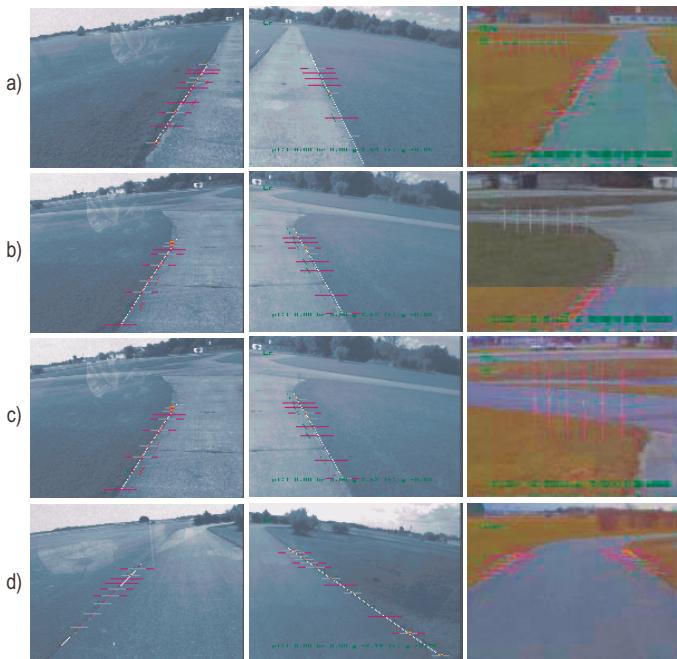


Fig. 10. Results of the optimization process

requirements. The interaction between perception and gaze control has been tested in real vehicles performing complex driving missions. Results of an autonomous turn-off maneuver have been presented.

References

- Aloimonos Y (1993) Introduction – active vision revisited. In Active Perception, pp 1–18. Erlbaum, Hillsdale.
- Ballard DH, Brown CM (1993) Principles of animate vision. In Active Perception, pp 245–282. Erlbaum, Hillsdale.
- Davis LS, Kushner TR, LeMoigne JL, Waxman AM (1986) Road boundary detection for autonomous vehicle navigation. *Optical Engineering*, 25(3):409–414.
- Dickmanns ED (1988) Dynamic computer vision for mobile robot control. In Procs. 19th Int. Symp. and Expos. on Robots, Sydney, Australia.
- Dickmanns ED (2000) Vertebrate-type vision for autonomous vehicles. In Conf. 'Intelligent Systems and Applications', Australia. University of Western Australia.
- Dickmanns ED, Behringer R, Dickmanns D, Hildebrandt T, Maurer M, Tho-manek F, Schiehlen J (1994) The seeing passenger car VAMORS-P. In Procs. Intelligent Vehicles Symposium 1994, IEEE Industrial Electronics Society, Paris.
- Dickmanns ED, Wünsche HJ (1999) Dynamic vision for perception and control of motion. In Handbook on Computer Vision and Applications, vol 3. Academic Press.

- Dickmanns ED, Zapp A (1986) A curvature-based scheme for improving road vehicle guidance by computer vision. In Proc. SPIE 86, pp 161–168.
- Gennery DB (1977) A stereo vision system for an autonomous vehicle. In Proc. 5th IJCAI, Cambridge (MA), USA.
- Gregor R, Lützeler M, Pellkofer M, Siedersberger KH, Dickmanns ED (2000) EMS-Vision: A perceptual system for autonomous vehicles. In Procs. IEEE Intelligent Vehicles Symposium 2000.
- Hofmann U, Rieder A, Dickmanns ED (2000) EMS-Vision: An application to intelligent cruise control for high speed roads. In Procs. IEEE Intelligent Vehicles Symposium 2000.
- Kuan D, Phipps G, Hsueh A (1986) A real-time road following vision system for autonomous vehicles. In Proc. SPIE 86, pp 152–160.
- Lützeler M, Baten S (2000) Road recognition for a tracked vehicle. In Procs. SPIE Conf. on Enhanced and Synthetic Vision, Orlando, FL. SPIE.
- Lützeler M, Dickmanns ED (1998) Robust road recognition with MARVEYE. In Procs. IEEE Conference on Intelligent Vehicles, IEEE Industrial Electronics Society, Stuttgart, Germany.
- Nilson NJ (1969) A mobile automation: An application of artificial intelligence. In Proc. IJCAI, pp 509–521.
- Pellkofer M, Dickmanns ED (2000) EMS-Vision: Gaze control in autonomous vehicles. In Procs. IEEE Intelligent Vehicles Symposium 2000.
- Thorpe C, Hebert M, Kanada T, Shafer S (1988) Vision and navigation for the Carnegie-Mellon Navlab. IEEE Trans. PAMI, 10(3):401–412.
- Zimdahl W, Rackow I, Wilm T (1986) OPTOPILOT – ein Forschungsansatz zur Spurerkennung und Spurführung bei Straßenfahrzeugen. VDI Berichte, 162:49–60.

Towards a Realistic Medical Simulator using Virtual Environments and Haptic Interaction

Christian Laugier, César Mendoza, and Kenneth Sundaraj

INRIA* Rhône-Alpes,
Laboratoire GRAVIR[†], SHARP Project,
ZIRST-655, Avenue de l'Europe, 38334 Saint Ismier Cedex,
FRANCE.

Abstract. This paper presents our experience towards the conception of a virtual reality medical simulator coupled with haptic interaction aimed at training surgeons. This area of research has a long history and a wide variety of approaches have been used. Generally, human tissue can be considered as a deformable body of viscoelastic material. To enable dynamic simulation of these bodies, we have patched three well known physical models onto their geometrical model : mass-spring networks which is more of a discrete object model, finite element method (FEM) based on continuum mechanics and recently long element method (LEM) which we believe to be more promising. We make some comparisons between these models. We also present some numerical resolution method for simulation of deformable bodies. As far as real-time interactions are concerned, we present our work on collision detection, haptic interaction and topology modifications. In the haptic system, we separate the physical simulation and the haptic interaction to ensure stability; the link between the two process is achieved by means of a local model which will be elaborated. We present some experimental results to highlight these works.

Keywords. Medical simulators, physical models, numerical methods, deformable objects, virtual reality, haptic interaction.

1 Introduction

Computer assisted surgery and medical robots is an emerging area of research. Extensive research is being done on the application of computers and robots for surgery, planning and execution of surgical operations and in training of surgeons. Surgeons are trained through apprenticeship. The basic techniques are taught with simple training equipment, but the rest of the training is done by using books describing surgical procedures and techniques and in the operating room by watching and participating in actual operations. Although operating room training is essential and invaluable, it does not provide the optimal environment to try or practice new techniques and procedures due to the risks to the patient. This method of training also limits the diffusion of knowledge since only a limited number of people can be trained by one experienced surgeon.

Virtual environments present an alternative to this training scheme. With virtual environments it is possible to create an interactive 3D simulator, where the surgeons,

* Institute National de Recherche en Informatique et en Automatique

† GRAphique, VIision et Robotique

using a haptic interface, can touch, manipulate or cut models of organs and tissues simulated on a computer. Virtual reality provides an environment where there is no risk to a patient and therefore less stressful. They are interactive and 3D in contrast to books, and they are less expensive compared to training in the operating room using animals or cadavers. Virtual environments also give a unique advantage, as it is possible to generate arbitrary anatomies and pathologies with which the surgeons can be trained for cases that they will encounter only a few times during their whole career, but nevertheless must be trained for. This way, it is also possible to standardize the training and accreditation in surgery. Already, a large variety of medical skills are being treated by virtual reality environments; laparoscopic surgery (Fig. 1), endoscopic exams, echography tests, etc.



Fig. 1. Laparoscopic surgery being performed on a human liver

Due to these advantages, multiple research centers have been created to focus on the development of medical simulators (Berkeley-UC, 2000; Carnegie, Mellon University, 2000; Harvard University, 2000; John Hopkins University, 2000; MIT, 2000; Stanford Medical School, 2000; Stanford/NASA, 2000; TIMC-CHU Grenoble, 2000). Most of the existing medical simulators use only the anatomic geometry ignoring the use of linear viscous elastic properties of tissues. Furthermore, surgical training simulators require deformable models of different human organs. However, the state of the art for interactive deformable object simulation is not sufficiently advanced to build realistic real-time simulators. Some work on realistic tissue simulations can be found in (Bro-Nielsen, Cotin, 1996; Cotin, 1997; d'Aulignac, Laugier, Cavusoglu, 1999; Delingette, 1998; Delingette, Cotin, Ayache, 1999; Gibson, 1997).

Thus, the main challenge to achieve realism in medical simulators is to obtain deformable tissue models that are interactive, i.e. efficient enough to be simulated in real-time, visually and haptically realistic. These models are also required to be manipulated, cut and sutured. It is the aim of this paper to present our experience in the conception of a medical simulator. We begin by presenting two commonly used physical model before describe the construction of a new one in more detail. Next, to simulate forward integration in time, we present some numerical resolution methods and briefly touch on their complexity and stability problems. Following this, we lay out how we handle some aspects of real-time interaction that are inevitable in current medical simulators. We end this paper with some conclusions.

2 Physical Models for Soft Tissue

Modeling and simulation of deformable objects for real-time applications is indeed a non-trivial task. A survey of deformable modeling was done by Gibson and Mirtich (Gibson, Mirtich, 1997). They describe much of the work done until 1997. In brief, they divided the work done on deformable objects into two parts: non-physically based models and physically based models. Physically based models can further be divided into discrete object models and other models based on continuum mechanics. The latter is usually solved using the finite element method (FEM). Recently, our research work has brought us to the conception of a physical model : Long Element Method (LEM). We believe that this model is particularly interesting for soft tissue simulation. The following sub-sections are devoted to these models, with special attention to the LEM model.

2.1 Mass-Spring Model

Principle The method of using mass-spring networks (Boux de Casson, Laugier, 1999; Delingette, Subsol, Cotin, 1994; Gibson, 1997) consists of a mesh of point masses connected by using elastic links. The mass-spring network is mapped onto the geometric mesh, i.e. the masses are the vertices and the springs are the edges of the mesh.

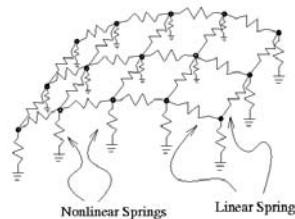


Fig. 2. Double layer non-linear mass-spring mesh of a human thigh

This mass-spring network is used to discretize the equations of motion. For example the link connecting pairs of nodes allows the local description of the elastic properties and consequently the displacement of the tissue. Any change of length relative to the resting length causes an internal force to occur between the two connected nodes. The simulation of the deformable object is done by modeling the effect of inertial and damping forces. Then, each node N in the mesh is subject to an equation of the form :

$$m \ddot{p} + b \dot{p} + \sum_i F_i = \sum F^{ext} \quad (1)$$

where p is the position of node N , \dot{p} and \ddot{p} are its velocity and acceleration, m is the mass attached to N , b is the damping coefficient to model viscosity, F_i is the internal

force exerted by a node N_i to which N is connected by a link (the sum Σ_i is taken over all such nodes N_i), and F_{ext} is the total force applied to N , for example, it can be the force exerted by a surgical tool or gravity. The force F_i is the viscous-elastic response of the spring connectors and is given by :

$$F = (-\lambda \Delta d - \mu \dot{d})k \quad (2)$$

where λ is the rigidity factor of a spring connector, μ is a damping factor, Δd is the relative variation of the distance between the two connected particles, \dot{d} is the relative speed between these two particles and $k = (b - a)/(|b - a|)$ where a and b are the connected particles. In Fig. 2 we can see a double layer non-linear model of a human thigh and its mass-spring network representation. The vertices are the masses and the edges are the springs. We implemented this model in an echographic simulator (d'Aulignac, Balaniuk, Laugier, 2000), of a human thigh shown in Fig. 9. The network is modeled by the following equations where Δx is the difference in the length of the springs with respect to their original resting length. This nonlinear spring response is chosen to model the incompressibility of the thigh after a certain deformation. The values of the surface elements are chosen uniform whereas the parameters of the nonlinear spring vary around the mesh to model the heterogeneous nature of the thigh.

$F = k \Delta x$	<i>- linear springs</i>	(3)
$F = \Delta x/(m \Delta x + n)$	<i>- non-linear springs</i>	

Advantages & Limitations The mass-spring model is relatively easy to implement. The physics of the model is simple and well understood. These models can also be simulated in real-time without much difficulty (Boux de Casson, Laugier, 1999; Delingette et al., 1994), unlike other continuum models. Since interactions in this model are local between nodes, parallel computations are possible. Since we are concerned with deformable objects, the stiffness phenomenon rarely appears. It has been used as the physical model in numerous applications (deformable objects), especially real-time applications, due to its rapidity. We have implemented a non-linear mass-spring network of a human liver in (Boux-de Casson, 2000). We find that this model generally conforms with reality and suitable for real-time interactions.

On the other hand, mass-spring models, as we have seen in the preceding sections, only takes into account two particles. When representing a volume using these binary connectors, the model can lead to several problems. Mass-spring connectors by itself has no volume. Of course, more springs will improve connectivity and thus produce a better approximation of the volume. Thus a volumetric object could perhaps be accurately modeled by infinite amount of particles and springs, but this is clearly not an option computationally speaking. To remedy this problem, it has been proposed to add cross springs, thereby connecting opposing corners. However, this implies that the physical behavior of the object is intrinsically dependent on the connectivity of the springs. When aiming for physical realism this is clearly a handicap. Alternatively (Deguet, Joukhadar, Laugier, 1998) proposed the use of angular and torsion springs, but again physical behavior is dependant on the topology and choice of the spring's parameters remains a black art.

2.2 Finite Element Method (FEM)

Principle Since the mass-spring model, which is a discrete model, suffers from certain drawbacks for certain applications, we decided to use a continuum model. The full continuum model of a deformable body gives the equilibrium of a body when subjected to external forces at any time. The model does this by minimizing the total potential energy of the system. In such a model, object deformation is described as material displacement. When the derivative of the potential energy of a body with respect to material displacement is zero, the continuous differential equilibrium criterion is satisfied.

FEM divides the object into a set of volumic elements and approximates the continuous equilibrium equation over each element. In our medical simulator, we chose to represent an elementary volume by a tetrahedron, that is a set of four points in three-dimensional space. We may express the deformation of this volume with respect to its original shape, using the Green-Lagrange tensor (O'Brien, Hodgins, 1999), which has the nice property of being invariant to rotation or translation. Given a position a of a point in the undeformed volume and let x be the position of the same point in the deformed configuration, then the deformation tensor is :

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial x}{\partial a_i} \cdot \frac{\partial x}{\partial a_j} \right) - \delta_{ij} \quad \text{where} \quad \delta_{ij} = \begin{cases} 1 : i = j \\ 0 : i \neq j \end{cases} \quad (4)$$

Since the internal stress of the volume is proportional to the deformation (or strain) we may calculate the forces on the particles if the stress-strain relationship is known. For Hookean (linearly elastic) and isotropic (identical elasticity along all axes) materials this relationship can be condensed into two parameters, also known as the Lamé coefficients.

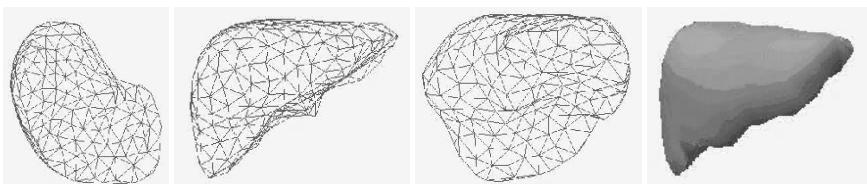


Fig. 3. Volumetric mesh for a finite element model of a human liver

Once the forces on the particles are known, the following second order differential equation can be solved using a numerical integrator for the node displacements where M , C , and K are the mass, damping and stiffness matrices respectively for the deformable body. F is the externally applied forces matrix and U is the displacement vector matrix. In Fig. 3 we can see a volumetric tissue of a human liver and its finite elements representation using tetrahedras.

$$M\ddot{U} + C\dot{U} + KU = \sum F^{ext} \quad (5)$$

Advantages & Limitations In contrast to the mass-spring model, FEM has a solid physical and mathematical foundation. The PDE's approximates exactly the deformation of an elementary element. Thus, FEM produces a more realistic physical simulation than mass-spring models with fewer nodal points. If the problem is of type static linear elasticity and linear geometry, the computational process of finding a solution amounts to solving a set of linear system of type $KU = F^{ext}$, where K has size $3N \times 3N$.

However, this model also has some drawbacks. The computational time spent on calculating the force of a nodal point is significantly longer when compared to a mass-spring model. If the topology of the object changes during the simulation, or if there is a large deformation, the mass and stiffness matrix must be re-evaluated during the simulation. The choice of the deformation tensor is arbitrary and generally depends on the application intended for. Also, FEM was intended for linear systems: non-linear systems can be simulated but at the loss of accuracy. Thus large deformations are generally not allowed.

2.3 The Long Element Method (LEM)

Principle We now present LEM (Costa, Balaniuk, 2001; Sundaraj, Laugier, F., 2001), a new physical model suitable for soft tissue simulation. The two basic principle that describe this model is Pascal's Principle and the conservation of volume. Unlike the previous two models, we use bulk variables such as pressure, density, volume, stress and strain to represent our object.

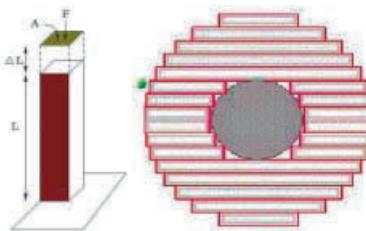


Fig. 4. A Long Element(right) and Modeling of non-homogenous elements(left) : soft tissue with rigid nucleus in the center

Consider the long element shown in Fig. 4. The force per unit area is defined as pressure. However, deformation produced by this pressure induces stress. For small applied forces, the stress ε , in a material is usually related to its deformation (i.e. elongation in our long elastic element). By introducing elasticity E , we can derive the force F , due to elongation where $K = AE/L$ (K is dependent on L).

$$P = F/A \quad \varepsilon = E \Delta L / L \quad F = K \Delta L \quad (6)$$

The static solution produced by our model requires that the external pressure P_{ext} , equal the internal pressure P_{int} . The external pressure on the surface is affected

by the atmospheric pressure P_{atm} , and by the stress due to elongation. The surface tension and environmental contact also contribute to the external pressure and we will integrate these factors into our model later. If the object is filled by fluid of density ρ , the internal pressure at a distance δ (from the upper part of the fluid) is due to pressure of fluid P_{fluid} and the effect of gravity g .

$$P_{ext} = P_{atm} + E\Delta L/L \quad P_{int} = P_{fluid} + \rho g \delta \quad (7)$$

Now, let us consider a set of long elements. Applying the previous derived equations to the group of N long elements and applying Pascal's Principle which gives constant ΔP , we then obtain the following continuous equation where $\Delta P = P_{fluid} - P_{atm}$ and $i = 1..N$. Since the fluid is incompressible, we can add the constraint of conservation of volume to this set of long elements.

$$E_i \Delta L_i / L_i - \Delta P = \rho_i g \delta_i \quad \sum A_i \Delta L_i = 0 \quad (8)$$

Finally, we add the surface tension P_{st} . These terms are of the form $P = kx/A$ where x is the difference between the deformation of an element and its neighbors and k is a local spring constant for each neighboring j of element i . This will give us the final set of equations.

$$P_{st}^i = \sum k_i^j (\Delta L_i - \Delta L_j) / A_i \quad E_i \Delta L_i / L_i + P_{st}^i - \Delta P = \rho_i g \delta_i \quad (9)$$

Advantages & Limitations Unlike the previous models, LEM uses bulk variables such as pressure, density and volume to model the object. These parameters are relatively more easy to be identified as compared to a mass of an element. Furthermore, the concept of an object being filled by an uncompressible fluid seems appropriate for soft tissue. By discretizing the object into columns of fluid, we get a system with much less variables leading to fewer equations and resulting in faster computation. These columns can also be used to represent non-homogeneous materials or even composite materials. No pre-calculation or condensation is required in the implementation of this model. The static solution produced by this model is sufficient for soft tissue simulation as these kind of materials are known to be well-damped. The complexity of this model is of $\mathcal{O}(3N^2)$ which is generally one order of magnitude less than the finite element method. The system is numerically solved by using any standard numerical method.

Nevertheless, there are some limitations in this model also. LEM is only valid for small deformations since the matrix A and its inverse A^{-1} changes for large deformations. Hence, it needs to be re-evaluated in the case of large deformations.

2.4 Construction of Models

The construction of these models can be divided into two phases : geometrical and physical. These phases are explained below :

Geometrical This phase deals with the 3D reconstruction. Typically, the data set is obtained from slices of MRI medical images. Several database of images exist;

one of them being the *Visible Human Project*¹. We obtain our surface mesh in our simulator from project *EPIDAURE* at INRIA Sophia-Antipolis². A volumic mesh can then be generated accordingly. We use *GHS-3D*, a software by INRIA to obtain a volumic triangulated mesh. For the LEM model, the surface mesh is treated by our method presented in (Sundaraj et al., 2001) to derive the columns of long elements filled with fluid. A liver modeled using LEM is shown in Fig. 5.

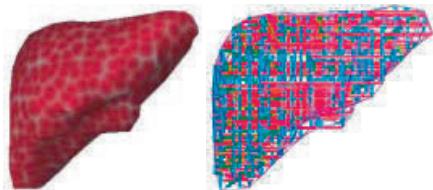


Fig. 5. A human liver discretized into long elements

Physical In this phase, we map a physical model onto the previously obtained geometrical model. For the mass-spring model, masses and springs are mapped onto vertices and edges respectively. In the FEM model, we use tetrahedras as an elementary volume and then lump the mass at the nodes of the tetrahedra. What remains to be done is to find the respective physical parameters. For the human liver, we took the physical parameters from (Davies, Carter, Roxburgh, Cuschieri, 1999). The parameters for the human thigh was obtained using a robot as shown in Fig. 9 and used as reference data.

3 Numerical Resolution

Real-time interaction is inevitable in a domain such as surgical training. A system which is computationally expensive is generally not acceptable. For any physical model, integrating it forward in time can be most difficult in terms of time and stability. Generally, system of particles/elements is represented as :

$$\begin{aligned} M\ddot{x} + C\dot{x} + Kx &= \sum F^{ext} \\ \dot{v} &= M^{-1}(-Cv - Kx + \sum F^{ext}) \\ \dot{x} &= v \end{aligned} \tag{10}$$

where M , C and K are the $3n \times 3n$ mass, damping and stiffness matrices. These matrices are typically quite sparse, M and C being diagonal or banded depending on the representing function. K is generally banded. F^{ext} is the total external forces. An analytical solution is not possible for such a system because of its complexity. So we have to search for a numerical one. This system of equations can give a dynamic resolution or a static resolution depending on the formulation.

¹ http://www.nlm.nih.gov/research/visible/visible_human.html

² <http://www-sop.inria.fr>

3.1 Dynamic Resolution

Dynamic resolution can be obtained using *explicit* or *implicit* integration.

Explicit Integration Once the force acting on a particle is known it becomes possible to evaluate its change in velocity and position. The simplest way to do so is using the Newton-Euler integration :

$$\begin{aligned} v_i^{t+\Delta t} &= v_i^t + \Delta t a_i^t \\ x_i^{t+\Delta t} &= x_i^t + \Delta t v_i^t \end{aligned} \quad (11)$$

However, if the object we are simulating is not *highly* elastic the resulting ordinary differential equations are stiff resulting in poor stability and requiring the numerical integrator to take very small time steps.

Implicit Integration To get around the problem of stiff springs (Baraff, Witkin, 1998) proposed to use implicit integration. Then velocity and position is obtained as follows :

$$\begin{aligned} v_i^{t+\Delta t} &= v_i^t + \Delta t a_i^{t+\Delta t} \\ x_i^{t+\Delta t} &= x_i^t + \Delta t v_i^{t+\Delta t} \end{aligned} \quad (12)$$

where if we let y be any required state and $f(y)$ the derivative, then we have the following for a general linear system :

$$\begin{aligned} f(y^t) &= \lambda y^t \\ y^{t+\Delta t} &= y^t + \Delta t f(y^{t+\Delta t}) = y^t + \Delta t \lambda y^{t+\Delta t} \\ y^{t+\Delta t} &= y^t / (1 - \lambda \Delta t) \end{aligned} \quad (13)$$

where λ is a constant. For linear differential equations, the iteration cannot diverge. When applied to non-linear differential equations, there is a possibility of divergence. Using this method, the update of a state over a time step requires solving a linear system the size of the number of degrees of freedom of all particles. Since the system is sparse a conjugate gradient can be used for solving.

3.2 Static Resolution

If the dynamics of our system are not essential like for soft tissue applications, we may decrease the complexity by neglecting the velocity. Since at equilibrium the external and internal forces are perfectly balanced we may use the principle of virtual work. Hence we find the displacements u using the tangent stiffness matrix K for a given external force. Recall that a similar expression is obtained when LEM is used based on equilibrium of internal and external pressure. Hence, the same resolution method can be applied.

$$Ku = f_{ext} \quad (14)$$

However, this is only valid for *small* displacements, since the stiffness matrix changes for larger displacements, and the solution of the system is non-linear beyond that. We use a Newton-Raphson iterative scheme for this purpose; initially

the displacements are calculated as in the linear case, but the residual internal force is reevaluated at the new configuration. The system is then solved again and again until convergence. This, of course, implies evaluating the stiffness matrix at each iteration. In the modified Newton-Raphson scheme the tangent stiffness matrix is *not* updated at each iteration, normally at the expense of an increased number of iteration. In our experience it is, however, a computationally interesting alternative.

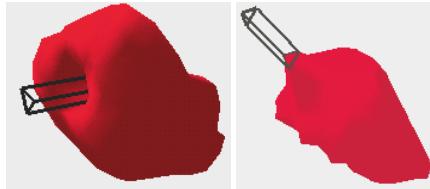


Fig. 6. A ‘pull’ and ‘push’ deformation by static resolution

Both the modified and the full Newton Raphson schemes demand the solution of a linear system at each iteration. Due to the large size of the matrix for non-trivial objects a direct solution is not computationally feasible. We choose to solve the system iteratively using successive over-relaxation (SOR) due to its simple implementation and potential for parallelization. Summarizing, we have two nested iterative schemes to solve the non-linear system. We use SOR to solve the linear system at each Newton-Raphson (full or modified) iteration in order to minimize the residual force. It is important to note that the Newton-Raphson iteration may experience convergence problems. In this case the correction to the incremental displacement vector should be weighted by a value < 1 .

4 Real-Time Interactions

Ultimately, the goal of a medical simulator is to allow real-time interactions and conform with reality. It is clear that the ability of a trainee surgeon to successfully grasp the skills of an operating procedure depends on the *learning* environment. It is our aim to provide the trainee the maximum possible. As far as a simulator is concerned, it is well known that during a simulation, given *any* physical model, the most difficult aspects, in terms of computational time or updating data structures are collision detection, the different rates for haptic interaction between graphical updates and physical simulation and topology modification during specific surgical procedures. We address these problems in the following sections.

4.1 Collision Detection

Deformable bodies may become concave during deformation. Many algorithms obtained from collision detection of convex objects (Gilbert, Johnson, Keerthi, 1988)

can be used by dividing concave polyhedra into several set of convex polyhedras. To avoid this costly time computations, Baraff and Witkin (Baraff, Witkin, 1992) divided objects into convex sub-objects that can only obey first order deformation (a facet or an edge can not be curved), which guarantees that they stay convex during simulation. In some litterature, Open-GL hardware acceleration has been suggested (Lombardo, Cani, F., 1999). We avoid this since its collision detection technique is highly depending of the configuration of the system, and performance can be lost while changing to another PC-class/MS Windows graphics environment (Linas Vepstas, 2000). These and other methods only give us the intersecting elements or primitives.

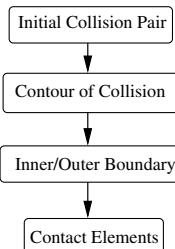


Fig. 7. Algorithm

However, computing the collision response requires us to try to evaluate the involved local deformation of the colliding objects; using a non-linear penalty method (Deguet et al., 1998). This can be done by determining the *fictitious interpenetration* of the objects. This means that the collision detection algorithm require not only the intersecting elements but also those elements that are interior to each other. Thus, we based our collision detection method on the bounding volume technique. We use Axis-Aligned Bounding Boxes (AABB) to bound our facets. By using a binary tree to represent the Bounding Box hierarchy, we apply the algorithm we have presented in (Sundaraj, Laugier, 2000) to localize all the interior facets.

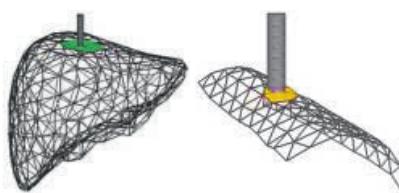


Fig. 8. Exact contact determination : liver(left) and thigh(right)

This alogorithm requires only an initial colliding pair of facet. Following this, a collision contour is constructed whereby allowing us to decide the facets which

are interior and exterior to the volume of collision. A simple recursive search will localize all interior elements. Figure 7 shows the these stages. The complexity of this algorithm is $\mathcal{O}(N)$, where N is the number of intersecting pairs. Figure 8 shows an example of our collision detection implementation. We are able to detect all interior facets as well.

4.2 Haptic Interaction

Haptic systems gives people the sensation of touching objects in virtual environments. Including haptic technology improves the perception of a surgeon thus producing a deeper sense of immersion. Multiple problems arise in haptic applications interacting with deformable objects, for example, costly computational time, numerical instability in the integration of the body dynamics, time delays, etc. Lengthy computations are forbidden in haptic systems which need high simulation rates (about 1KHz) to obtain realistic force feedback. The update rates of the physical objects being simulated is normally of the order of 20 to 150 Hz. This difference in simulation rates can cause an oscillatory behaviour in the haptic device that can become highly unstable and inflict harm on the operator (Adams, Hannaford, 1999). Several numerical approaches (Cavusoglu, Tendick, 2000; Ellis, Sarkar, Jenkins, 1997) have been proposed to solve the difference rate problem. However, due to the unknown nature of human behavior, there will always exists a small error that cause slight vibrations. In our approach, instead of interacting with the complete physical model we use a simple intermediate representation : local models, as in (d'Aulignac et al., 2000; Mark, Randolph, Finch, 1996; Mendoza, Laugier, 2001; Ruspini, Kolarov, O., 1997; Zilles, Salisbury, 1994) which allows to compute the force feedback at the haptic required frequency. The force F , is calculated using Hooke's law and the minimal distance d , between the local model and the haptic tool :

$$F_{haptic} = \begin{cases} kd f_{model} & \text{if } d \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where the local stiffness k , depends on the physical model force f_{model} to give a more realistic sense of touch. To avoid small vibrations due to sudden changes in the values of f_{model} (updated at the simulation frequency), the haptic update procedure is done gradually in each haptic iteration until the new value is reached. The computation of the minimal distance, d , depends on the intermediate representation used. We have two main approaches which are explained below.

Analytic Based This representation is oriented to non-highly deformable objects (d'Aulignac et al., 2000) whered is computed by obtaining the distance to a sphere or to a single plane. To update them we compute its configuration parameters $u = [u_1 u_2 \dots]$ (e.g. the center and radius of a sphere). For example, let d_{model} be the distance between the position of the virtual probe, $x_{virtual}$, and the complete object. Let d be $\phi(x, u)$, where $x = [x_1 x_2 \dots x_n]$ are the variables that correspond to the configuration space dimensions. The goal is to find u_i that keeps $d_{model} = \phi(x, u)$.

The algorithm is based in the first order approximations of (16).

$$\begin{aligned}\Delta\phi(x, u) &\simeq [\partial\phi(x, u)/\partial u]\Delta u \\ \Delta\dot{\phi}(x, u) &\simeq [\partial\dot{\phi}(x, u)/\partial u]\Delta u\end{aligned}\quad (16)$$

By introducing the Jacobian matrix J , the required parameters Δu are obtained using the pseudo-inverse of J ($J^\dagger = (J^T J)^{-1} J^T$) given by (17).

$$J = \begin{bmatrix} \frac{\partial\phi(x, u)}{\partial u} \\ \frac{\partial\dot{\phi}(x, u)}{\partial u} \end{bmatrix}, J\Delta u = \begin{bmatrix} \Delta\phi(x, u) \\ \Delta\dot{\phi}(x, u) \end{bmatrix} \text{ and } \Delta u = J^\dagger \begin{bmatrix} \Delta\phi(x, u) \\ \Delta\dot{\phi}(x, u) \end{bmatrix} \quad (17)$$

This approach has been implemented in an echographic simulator of a human thigh shown in Fig. 9. The data adquisition is done using a PUMA robot arm, as seen in Fig. 9-a. We use a two layer mass-spring model, as explained in Section 2.1. The physical parameters are adjusted using a non-linear least square estimation (d'Aulignac et al., 1999) and implicit integration is applied to solved the dynamics of the system. Due to the relatively small deformations of the thigh, planes or spheres are likely to be used as the intermediate representation, see Fig. 9-b. Echographic images are produced by the interaction between the deformable model and a virtual tool as shown in Fig. 9-c. The virtual tool is a representation of the haptic device (PHANToM type) which sends the force feedback to the user.

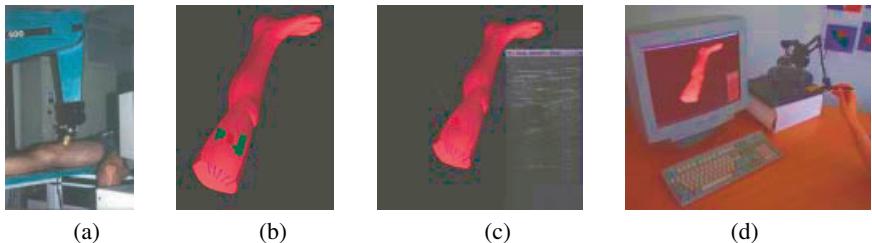


Fig. 9. (a) Data adquisition (b) Local Model : Plane (suitable for objects with small deformations) (c) Real time deformations with ecographic images (d) Haptic Interaction

Local Topology Based For highly deformable objects we propose a local model (Mendoza, Laugier, 2001) based on the topology of the object. The computation of d is more complicated than the previous approach, however updating the intermediate representation is easier since it is constructed by using the colliding facets and its neighboring facets as a set of planes. The minimal negative distance is calculated between haptic position, x_{haptic} and a point x_p , called proxy (Ruspini et al., 1997) lying on the surface of the planes. The proxy is constrained to never have a negative distance with respect to the other planes. Generally, the minimal negative distance problem is be written as :

$$\min d = \| x_p - x_{haptic} \| \quad \text{such that } P_i = 0 \quad (18)$$

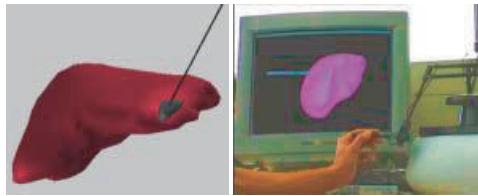


Fig. 10. Haptic interaction with the aid of a local model : (a) The topological local model and (b) Haptic interaction with a PHANToM device

The expression $P_i = 0$ is the equation of the different planes that form the local surface where i is the number of planes which can be reduced to 3 using (Zilles, Salisbury, 1994). Lagrange multipliers λ , are used to solve the minimization problem (Ruspini et al., 1997; Zilles, Salisbury, 1994). An illustrating example of this solution to obtain x_p is shown in (Mendoza, Laugier, 2001) where:

$$\lambda = [N^T]^{-1} (x_{haptic} - N^{-1} D) \quad x_p = x_{haptic} - N^T \lambda \quad (19)$$

$N = [n_1 \ n_2 \ n_3]^T$ is the combined set of normals belonging to each plane and $D = -[p_1 n_1 \ p_2 n_2 \ p_3 n_3]^T$ and p_i the baricentre point of the facet corresponding to plane i . This approach has been implemented for haptic interaction between different human organs like the liver or the vesicle which have a high degree of concavity meaning high deformability. The topological intermediate representation is shown in Fig. 10-b where the virtual tool (PHANToM device) touches a human liver. For the dynamic simulation of the liver we have implemented and experimented the three types of physical models described in Section II. Notice that the intermediate representation takes the local shape of the liver.

4.3 Topology Modifications

One important feature in virtual reality medical simulators is the *cutting* phenomena. Cutting can be implemented either in mass-spring models (Bielser, Maiwald, Gross, 1999; Norton, Turk, Bacon, Gerth, Sweeney, 1991; Terzopoulos, K., 1988) or finite element models (O'Brien, Hodgins, 1999). In both implementations, cutting is carried out by simply removing an element or dividing it. In our previous work (Boux de Casson, Laugier, 2000) we have introduced a methodology to simulate real-time tearing phenomena. This approach is based on an interesting *separation* of the involved physical elements of the model instead of destroying or splitting them. If the elements (e.g. facets, tetrahedrons, etc) are destroyed the complexity of the simulation is decreased but the discretization of the mesh needs to be larger in order to maintain realism. However, this leads to a drawback in real-time interactivity. The subdivision approach increases the number of elements, making the simulation slower at each cutting operation. Recently we have extended this approach to the cutting phenomena using a mass-spring mesh. This works well for surface meshes

(skin simulation cutting) and can be implemented for volumetric meshes. In (Nienhuys, Vanderstappen, 2000) some research is presented using finite element models but it is still computationally expensive for real time interaction.

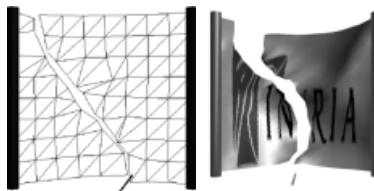


Fig. 11. (a) Global idea of the cutting process (b) Cutting a virtual 2D skin mesh using a haptic device (type PHANToM) to manipulate the virtual tool

We consider the cutting phenomenon as an interaction between a deformable virtual object and a rigid body or tool. In this process the deformable model is cut and its geometrical topology changes. Three important criteria are considered : (1) preserving the main physical properties, (2) obtaining realistic visual effects and (3) satisfying real-time constraints. Then, rather than dividing the facets of the initial mesh into smaller ones, we *separate* them. Separating the facets provides us a way to avoid matter from disappearing and to avoid an important growth in the complexity of the model. In Fig. 11-b, a 2D mesh is cut using our algorithm. This becomes important while simulating skin cutting which is an important feature towards a realistic laparoscopic simulator. The steps to model the cutting phenomena at time t once a collision between the rigid body and the deformable object has been detected is presented below.

Step 1 : Assigning cutting elements Let $S(t)$ be the closest vertex to the contact point $C_p(t)$. Then a cut is carried out between $S(t - 1)$ and $S(t)$ if the following conditions are verified :

- $C_p(t)$ and $C_p(t - 1)$ belong to different facets.
- $S(t - 1) \neq S(t)$.

Step 2 : Finding the cutting path In (Boux-de Casson, 2000) we propose an algorithm to determine the edges that have to be split between vertices $S(t - 1)$ and $S(t)$.

- Vertex $S(t - 1)$ is placed in the position of $C_p(t - 1)$ and vertex $S(t)$ in the position of $C_p(t)$.
- Vertices belonging to the cutting path are orthogonally projected to the line between $S(t - 1)$ and $S(t)$ as seen in Fig. 11-a.

Step 3 : Remeshing along the cutting path We take the following steps to remesh along the cutting path between vertices $S(t - 1)$ and $S(t)$:

- A new particle is created, at the same position as the split one, with the same velocity, and half of its mass.
- The rest-length of the broken spring is updated to be its actual length.
- A spring is created, with the same elasticity and viscosity as the broken one.

5 Conclusion

This paper has presented our experience towards the conception of a realistic virtual medical simulator. We have touched areas on physical models, numerical resolution techniques and real-time interactions. The physical models presented covers discrete object models to full continuum mechanics models. These models have been used in various application and they have been highlighted. We have also described LEM in detail which has complexity one order of magnitude less than FEM. We have also proposed the use of a static resolution instead of a dynamic one. Very essential real-time interactions have been presented together with some results. We use bounding volume techniques for collision detection which we believe to be suitable for deformable objects like virtual organs in medical simulators.

Table 1. Comparison of the 3 physical models

	Mass-Spring	FEM	LEM
Rapidity	★★	★	★★★
Realism	★	★★★	★★
Implementation	★★★	★	★★
Soft Tissue Model	★	★★	★★★
Updating Topology	★★	★	★★★

We have also presented a methodology to solve the instability problem of the update rate difference between the haptic and the physical simulation. The virtual reality 2D cutting phenomena has been described. We believe that the ideas can be exported to a 3D cases. Other real-time topology modifications will be implemented as well, e.g. suture operation. To end the paper, we compare the three physical models presented in this paper against some important criterias. They are presented in Table 1.

Acknowledgements.

This work is within the framework of project CAESARE by the French National Institute for research in Computer Science and Automation. It has also been supported by the National Council of Science and Technology of Mexico (CONACYT). We would like to thank I. F. Costa and R. Balaniuk for their contribution. K. Sundaraj would like to acknowledge the financial grant from the French Ambassy for his doctoral thesis.

References

- Adams R, Hannaford B (1999) Stable haptic interaction with virtual environments. In IEEE Transactions on Robotics & Automation 99.
- Baraff D, Witkin A (1992) Dynamic simulation of non-penetrating flexible bodies. In Computer Graphics.
- Baraff D, Witkin A (1998) Large steps in cloth simulation. In Computer Graphics (Proc. SIGGRAPH), pp 43–54.
- Berkeley-UC (2000) Virtual environments for surgical training and augmentation, (VESTA). URL <http://robotics.eecs.berkeley.edu/~mdownes/surgery/surgsim.html>.
- Bielser D, Maiwald VA, Gross MH (1999) Interactive cuts through 3-dimentional soft tissue. In EUROGRAPHICS'99, vol 18, pp C31–C38.
- Boux-de Casson F (2000) Simulation dynamique de corps biologiques et changements de topologie interactifs. PhD thesis, INRIA Rhône-Alpes and Université de Savoie.
- Boux de Casson F, Laugier C (1999) Modelling the dynamics of a human liver for a minimally invasive simulator. In Medical Image Computing & Computer Assisted Intervention-MICCAI'99 Proceedings.
- Boux de Casson F, Laugier C (2000) Simulating 2D tearing phenomena for interactive medical surgery simulators. In Computer Animation 2000, Pensilvania, USA.
- Bro-Nielsen M, Cotin S (1996) Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In Proc. Eurographics'96.
- Carnegie, Mellon University (2000) Center for medical robotics and computer assisted surgery. URL <http://www.mrcas.ri.cmu.edu>.
- Cavusoglu M, Tendick F (2000) Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments. In Int. Conf. on Robotics and Automation '00.
- Costa I, Balaniuk R (2001) Static solution for real time deformable objects with fluid inside. In ERCIM News, pp 44–45.
- Cotin S (1997) Modèles anatomiques déformables en temps-réel. PhD thesis, Epidaure-Sophia Antipolis.
- d'Aulignac D, Balaniuk R, Laugier C (2000) A haptic interface for a virtual exam of the human thigh. In Int. Conf. on Robotics and Automation, San Francisco, USA.
- d'Aulignac D, Laugier C, Cavusoglu M (1999) Modeling the dynamics of a human thigh for a realistic echographic simulator with force feed-back. In Medical Image Computing and Computer Assisted Intervention – MICCAI'99 Proceedings, pp 1191–1198.
- Davies P, Carter F, Roxburgh D, Cuschieri A (1999) Mathematical modelling for keyhole surgery simulations: spleen capsule as an elastic membrane. In Journal of Theoretical Medicine.
- Dequet A, Joukhadar A, Laugier C (1998) Models and algorithms for the collision of rigid and deformable bodies. In Robotics: the algorithm perspective, Proc. of the Workshop on the Algorithmic Foundations of Robotics, Houston, USA.
- Delingette A, Subsol G, Cotin S (1994) A cranofacial surgery simulation tesbed. In Proc. Visualization in Biomedical Computing (VBC).
- Delingette H (1998) Rapport de recherche, no.3506, towards realistic soft tissue modeling in medical simulation. Technical report, INRIA, Sophia-Antipolis.
- Delingette H, Cotin S, Ayache N (1999) A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. In Computer Animation, Geneva Switzerland.

- Ellis E, Sarkar N, Jenkins M (1997) Numerical methods for the force reflection of contact. In ASME Transactions of Dynamic Systems Measurement and Control, pp 768–774.
- Gibson F, Mirtich B (1997) A survey of deformable models in computer graphics. Technical report, MERL, Cambridge, MA, <http://www.merl.com/reports/TR97-19/index.html>.
- Gibson S (1997) 3D ChainMail: A fast algorithm for deforming volumetric objects. In Proc. Symp. on Interactive 3D Graphics, ACM SIGGRAPH, pp 149–154.
- Gilbert E, Johnson D, Keerthi S (1988) A fast procedure for computing the distance between objects in 3d space. In IEEE Int. Conf. on Robotics and Automation.
- Harvard University (2000) Image guided therapy program. URL <http://splweb.bwh.harvard.edu:8000>.
- John Hopkins University (2000) Nsf: Engineering research center for computer-integrated surgical systems and technology. URL <http://cisstweb.cs.jhu.edu>.
- Linas Vepstas (2000) High performance graphics hardware design requirements. URL <http://linas.org/linux/graphics.html>.
- Lombardo J, Cani M, F. N (1999) Real-time collision detection for virtual surgery. In Computer Animation, Switzerland.
- Mark W, Randolph S, Finch (1996) Adding force feedback to graphics systems: Issues and solutions. In Computer Graphics Proc. SIGGRAPH, New Orleans, Louisiana.
- Mendoza C, Laugier C (2001) Realistic haptic rendering for highly deformable virtual objects. In IEEE Virtual Reality VR 2001, Yokohama JAPAN.
- MIT (2000) Image guided surgery. URL <http://www.ai.mit.edu/projects/medical-vision/index.html>.
- Nienhuys H, Vanderstappen F (2000) Combining finite element deformation with cutting for surgery simulations. In EUROGRAPHICS 2000.
- Norton A, Turk G, Bacon B, Gerth J, Sweeney P (1991) Animation of fracture by physical modeling. In The Visual Computer, vol 7, pp 210–217.
- O'Brien J, Hodgins J (1999) Graphical models and animation of brittle fracture. In SIGGRAPH 99 Conference Proceedings.
- Rusipni D, Kolarov K, O. K (1997) Haptic interaction in virtual environments. In Proceedings of Int. Conf. on Intelligent Robots and Systems 1997, Grenoble-FR.
- Stanford Medical School (2000) Center for advance technology in surgery at Stanford. URL <http://catss.stanford.edu>.
- Stanford/NASA (2000) National center for biocomputation. URL <http://www-biocomp.stanford.edu>.
- Sundaraj K, Laugier C (2000) Fast contact localisation of moving deformable polyhedras. In Proc. of the Int. Conf. on Control, Automation, Robotics and Vision, Singapore.
- Sundaraj K, Laugier C, F. CI (2001) An approach to LEM modelling : Construction, collision detection and dynamic simulation. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.
- Terzopoulos D, K. F (1988) Modeling inelasctic deformation: Viscoelasticity, plasticity, fracture. In Computer Graphics, Proceedings of SIGGRAPH'88, Atlanta, Georgia.
- TIMC-CHU Grenoble (2000) Gestes médico-chirurgicaux assistés par ordinateur (GMCAO). URL <http://www-timc.imag.fr/gmcao>.
- Zilles C, Salisbury K (1994) A constraint-based god-object method for haptic display. In ASME, Haptic Interfaces for Virtual Environment and Teleoperator Systems, Dynamics Systems and Control, Chicago, Illinois, vol 1, pp 146–150.

Spoken Language Interface of the Jijo-2 Office Robot

Toshihiro Matsui, Hideki Asoh, Futoshi Asano, John Fry, Isao Hara, Yoichi Motomura and Katsunobu Itoh

Real-World Intelligence Center, National Institute of Advanced Industrial Science and Technology (AIST)

Abstract. In order for a mobile robot to provide information services in real offices, the robot has to maintain the map of the office. Rather than a completely autonomous approach, we chose to interact with office people to learn and update the topological map using spoken dialogue. To successfully apply a speech recognition technology to conversation understandings in real offices, we implemented a multiple microphone array system and a context and attentional manager in the robot. The robot could demonstrate simple map learning, route guidance, and information service about people's location .

1 Introduction

In 80th, robots became common workforce in factories. In the next decade, the robot technologies found a challenging exploration task on Mars. Recent home users are excited by the idea of petting robots for entertainment and welfare. In the middle of factories and homes, we believe there is a demand for office robots that work as secretaries. At AIST, we have developed a prototype office robot called Jijo-2 (Fig. 1). We suppose office robots should be able to provide information services combined with locomotion, such as route guidance, finding people, delivery, schedule arrangement, etc. Office robots are expected to have capabilities to autonomously adapt to environments and to communicate with people in natural manners. Therefore, the research topics are map learning, localization, speech dialogue, human recognition and the integration architecture.ver have been presented.

Practical mobile robots for delivery tasks in factories use preprogrammed maps or signal-emitting wires embedded in the floor for guidance. Whether a mobile robot can navigate in an unknown environment without preprogrammed information or artificial settings is a research issue (Asoh, 1999; Asoh, Motomura, Matsui, Hayamizu, Hara, 1996). For friendly human-robot interaction, spoken language interface plays an important role. Our interest is in how we can improve the speech recognition and dialogue techniques for robots in real offices taking into the account the language characteristics Asoh, Matsui, Fry, Asano, Hayamizu (1999); Matsui, Asoh, Fry et al. (1999). In the following sections, Jijo-2 robot's spoken dialogue system to acquire a topological map and to provide office information service is described.

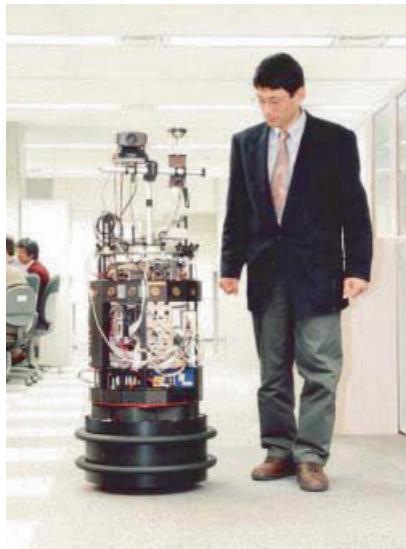


Fig. 1. Jijo-2 robot is talking with a human user while navigating in an office corridor. Multi-Agent architecture allows two behaviors to happen simultaneously

2 Robot Architecture

For the basis of the robot, we use a mobile robot platform, Nomad-200 (Fig. 1), of Nomadic Technologies. It has sonar sensors for measuring ranges up to several meters in 16 directions, infrared proximity sensors for faster and shorter distance measurement in as many directions, and bumper sensors for collision detection. We added two TV cameras, an array microphones and a speech synthesizer. Inside the robot is a Linux computer capable of communicating via radio Ethernet.

The control software is structured in two layers consisting of re-active modules onboard and deliberative modules for planning and dialogue implemented on a remote host. Since modules are running in separate processes, concurrent behaviors as the robot can talk while it navigates happen. The whole structure is a multi-agent architecture that allows dynamic addition, removal and replacement of modules to adapt to behaviors (Matsui, Asoh, Hara, 1997). Each module communicates by TCP. An integrator module requests a reactor module to monitor various events, i.e., low battery voltage, finding an open space, loud sound, etc. When a reactor module encounters a specified event, a reply message is sent to the requestor, which invokes registered event function. An integrator module consists of multiple event handling functions, and there is no explicit main program in the system.

3 Navigation and Map

3.1 Navigation using topological maps

It is well understood that mobile robot cannot solely rely on the odometer and sonar for long distance navigation because of accumulated. The most basic navigation behavior of Jijo-2 is to follow a corridor along walls. Though sonar cannot produce enough information to recognize objects, it can steer to an open space avoiding obstacles. Also they can count the number of openings to offices if there is a change of distances to walls at the entrances. Therefore, Jijo-2 exploits a topological map representing connectivity between the office entrances as shown in Fig. 2. Each arc describes actions to reach the destination together with a statistic history about the success and failure of the actions. This information is looked up and updated in the dialogue-guided map learning process.

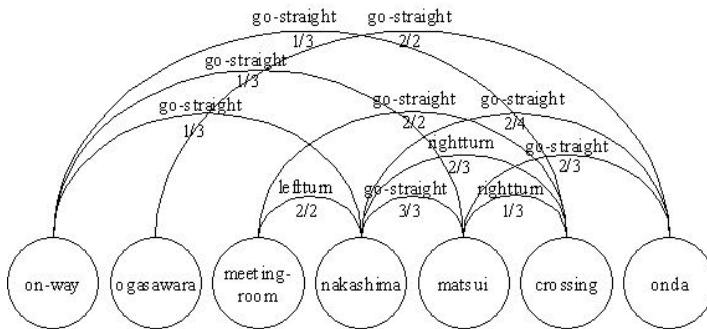


Fig. 2. Topological map connecting location names with behaviors

3.2 Dialogue-guided map learning

We do not assume Jijo-2 is given a map of an office environment in advance. From the moment the robot is put in an unknown environment, it begins to learn the map from office workers. This is like a situation where a freshman to a new office is taken to important places and gets instructions about the location names. If he encounters with someone on the way, he is also noticed the name of the person by the instructor. Jijo-2 does a similar task (Matsui et al., 1999).

Since a map is represented by a network consisting of nodes for location names and of arcs for actions to traverse, the map learning is a task to create new location nodes, to record actions, and to update success/failure statistics (Fig. 2). The statistics represent how Jijo-2 is confident about the location, and when he is not confident, Jijo-2 invokes an action to ask where he is Asoh (1999); Asoh, Matsui (1999); Asoh et al. (1996).

4 Spoken Language Dialogue System

4.1 Speech recognition

Jijo-2 uses a speaker-independent continuous Japanese speech recognizer developed at ETL. Most speech recognizers assume the microphones set close to mouths. Microphone setting is important to keep high S/N ratio and to keep the frequency spectrum distribution relatively equal to the phoneme models. If we apply the speech recognizer to voices at 1 meter distance, chances of mis-recognition increase. Also, since most commercial systems collected correlation data between phonemes and words from text corpus, they are good at recognizing readouts of texts like newspaper articles, but not very suitable for recognizing daily conversation.

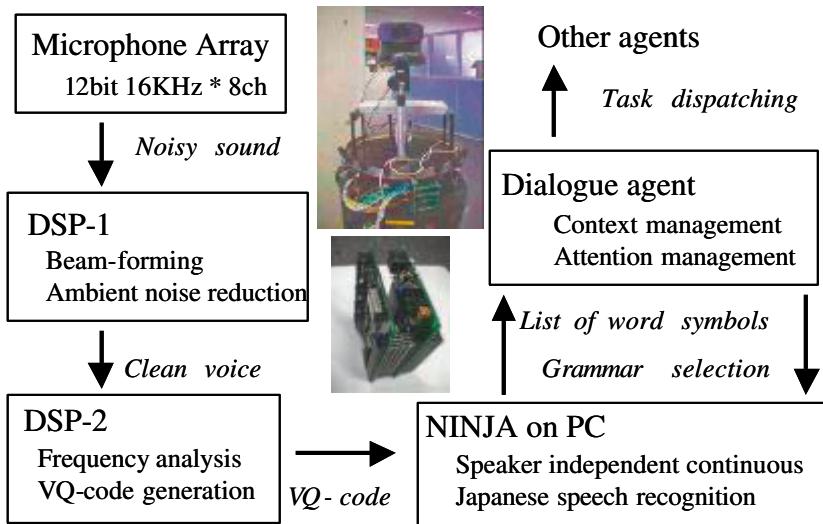


Fig. 3. Flow of speech input, recognition and task dispatching

To cope with this microphone problem, we developed an array microphone system for Jijo-2. Figure 3 depicts the entire structure of our speech recognition and understanding system. On the edge of the robot's top tray are eight omni-directional microphones forming a half circle of 40 cm diameter. Based on time difference for sounds to arrive at these microphones, the direction of the sound source is computed, and a focus beam is formed. By this technique we could improve the S/N ratio for consonants by 10 dB (Asano, Asoh, Matsui, 1999). This beam-forming processing is done by the first DSP (TI-C44/40MHz). The noise free voice signal is sent to the second DSP, which generates the vector quantization code every 10 ms. Each VQ-code is an integer representing a phoneme in a short time frame. The VQ-code is sent to the PC and recognized by the speaker independent continuous Japanese

speech recognizer, NINJA (Itou, Hayamizu, Tanaka, Tanaka, 1993). NINJA looks up the dictionary and grammar book consisting of about 150 words and 300 rules.

4.2 Natural Japanese dialogue

The Japanese language has an obvious tendency to omit subjects, objects or any words if they are easily inferred from the context. Omission occurs more frequently in daily conversation than written texts. This makes the speech understanding complicated on the one hand, and makes the speech recognition and syntactic analysis easier, since each utterance becomes shorter. Jijo-2 conducts a speech dialogue processing that exploits these characteristics of Japanese.

Jijo-2's dialogue manager traverses between several dialogue states as depicted in Fig. 4. The states are used to restrict the possible utterances to which Jijo-2 reacts. For example, a conversation must begin with user's utterance of "Hello". In the confirmation state, Jijo-2 only listens to "Yes" or "No". This state transition network is needed to eliminate spurious voice inputs that are often generated as noises in an office. This state transition is also used to choose an appropriate dictionary and grammar for the next utterance. Commands like stop and cancel are recognized in any state for safety.

4.3 Dialogue contexts

Each utterance of a user only gives a fraction of information. For example, though "Turn" is a complete Japanese imperative sentence, the robot does not know which way to turn. In order to keep track of a series of relevant utterances and to construct semantics for a robot's behavior, we use the dialogue context.

Currently, we define 7 contexts: *query-context*, *update-context*, *identification-context*, *navigation-context*, *call-context*, etc. A context is defined as to hold a number of required info and optional info as property variables of a EusLisp object. For example, a *query-context* that is created when an interrogative sentence is heard requires person property, and has location, start-time, and end-time as optional properties.

Figure 5 illustrates five dialogue contexts created in a series of utterances. Conversation is guided to fulfill the required property by giving appropriate questions and confirmations. Optional properties may either be given in the utterance, assumed from the attentional state, or assumed by predefined default. For example, "business trip" assumes the destination to be "Tokyo" unless it is explicitly spoken. The state transition network is programmed in Prolog implemented in EusLisp.

4.4 Slot filling by managing attentional states

The dialogue agent maintains an attentional state which indicates the relative salience of discourse referents (the individuals, objects, events, etc). Currently the attentional state is implemented as a total ordering. The function of the dialogue

State Transitions and Behaviour Control

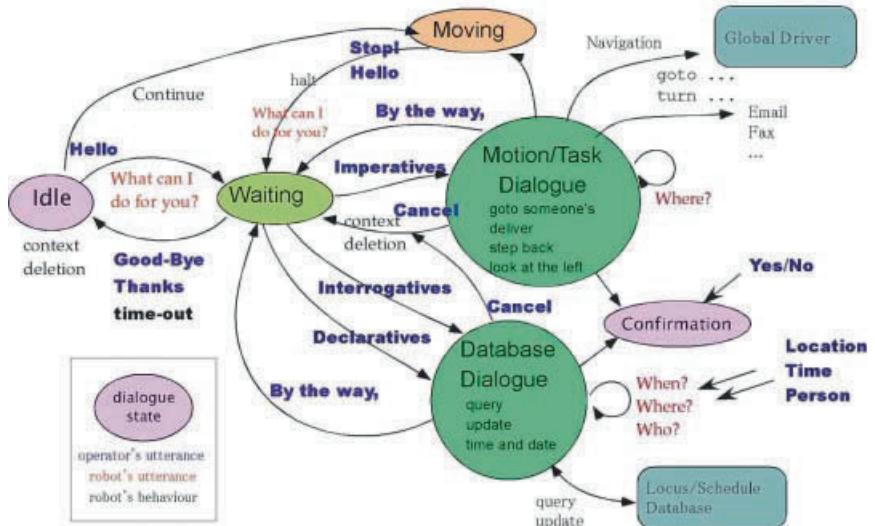


Fig. 4. Dialogue states and transitions.

manager is to exploit the attentional state in order to accomplish (1) processing of under-specified input (information “unpackaging”) and (2) natural-sounding language generation (information “packaging”).

Japanese zero pronouns In Japanese the problem of under-specified input is acute because in general the subject, object or any other argument to the verb is omitted from an utterance whenever it is clear from context. For example, the word “Todokete” is a well-formed Japanese sentence, even though its English translation “Deliver!” is not. The missing arguments in Japanese are referred to as zero pronouns. Thus the Japanese request *todokete* corresponds roughly to the English request (won’t you) deliver this (to him).

Centering To attack the zero pronoun ambiguity problem in a principled way we turned to the technique of centering (Grosz, Joshi, Weinstein, 1995). Centering is used for modelling discourse and for predicting the antecedents of pronouns based on the principle that the more salient an entity is in the discourse, the more likely it is to be pronominalized. The centering model is broadly language-independent (Fry, Asoh, Matsui, 1998).

Attentional states managed by file cards An attentional state is managed by a stack of seven file cards. Salient topics like location names, person names, times, etc., are recorded in the file. If a particular topic is needed in the next discourse

analysis, the topic is searched in the file cards from the top. If found, it is given as a default value to the required/optional slot. Since new topics are always pushed from the top of the stack, the oldest topic at the bottom is removed to keep the number of attentional states to seven. The information in the attentional states is kept during the switching between discourse. In other words, the attentional state stack bridges information from one script to another.

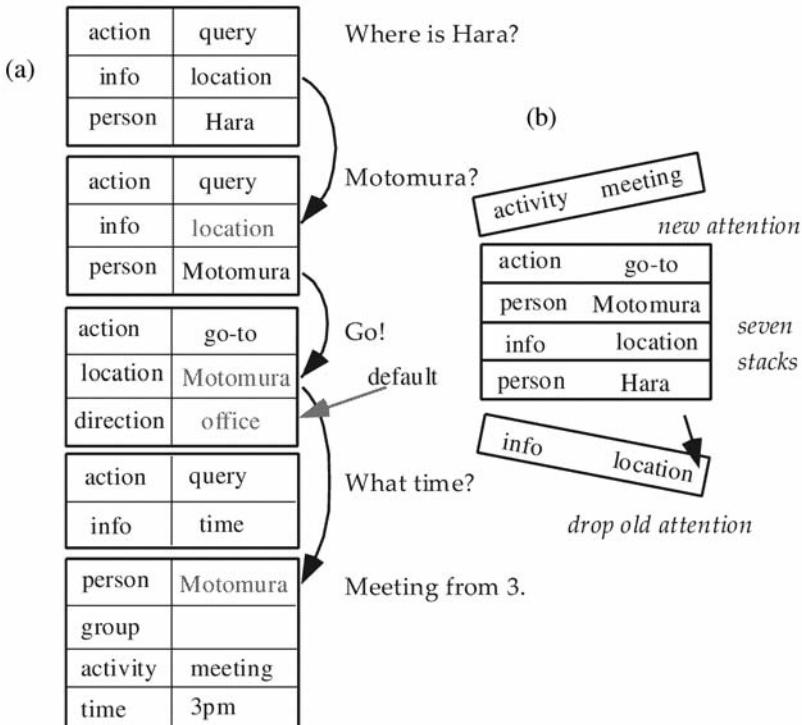


Fig. 5. Dialogue context (a) and attentional state (b). Attentional state stack carries focus from one context to another.

5 Task Execution

The dialogue is controlled in the integrator layer. Once an action to take is determined in a dialogue, the dialogue agent issues a request to other integrator agents. If it is navigation to someone's office, a go-to command is sent to the driver agent. If the driver agent could successfully plan a path to a destination, it immediately starts navigation. If not, the driver agent requests the dialogue agent to ask a way to reach there. In this manner, each integrator agent does not have fixed precedence

over other agents. The basic strategy is that if a task is attainable in the agent, it is processed locally, and if not, the task is dispatched to others.

5.1 Database behaviors

If the current context is a *query-context* or *update-context*, the task is dispatched to the database module. Normally, a response is immediate and the dialoguer pronounces the result. If query takes long, the user may want to start another conversation, which is properly handled by the dialogue module running concurrently in an event-driven manner. The schedule and location database is implemented on a PostgreSQL server, which also allows web-based access. Therefore, once a dialogue about a user's schedule updates the database, the information is made available to public access.

5.2 Navigation dialogue

For *navigation-* or *call-context*, the task is dispatched to the driver module. The map is maintained in the driver module, although it is learned through dialogue. Therefore, a situation where the dialogue module commands the driver to go to someone's office, but the driver does not know the path, may happen. In this case, the driver module requests the dialogue module to ask for navigation instructions. During navigation, the driver might encounter an unexpected landmark, which is usually a *close-to-open* event (an open space is found) from the sonar. This also leads the dialogue module to conduct a conversation to confirm the location.

Jijo-2 continues dialogue while it navigates. If the dialogue is handled within the module or with the database like a query about the current date and time, it is locally processed without interfering with the navigation. But if the dialogue contains commands to stop navigation and to change destination, the dialogue module retracts the current command to the driver and restarts another behavior.

6 Example Dialogue and Behavior

Figure 6 illustrates two patterns of dialogue between human users and Jijo-2 involving several different behaviors.

Dialogue (a) is composed of simple motion commands. Jijo-2 rotates its body by the turn command, and pans the camera by the look-at command. Though direction keyword is required for both, Jijo-2 can assume it simply from the previous utterance.

Dialogue (b) begins with “Hello” making Jijo-2 turn to the person's direction. Then the robot captures a face image and tries to recognize the person. If the recognition succeeds, the robot can know the speaker's name, which is pushed in the attention state.

When the user asks “Asoh-san wa doko (Where is Mr. Asoh?)”, Mr. Asoh is established as a topic of the dialogue, and therefore as the most salient entity in

H: Migi wo muite. (turn to the right)
 R: turn to the right
 H: Hidari. ([turn to the] left)
 R: turn to the left
 H: Ushiro. ([turn to the] back)
 H: Hidari wo mite. (Look at the left).
 R: Look at the left
 H: Migi. (Right)
 R: Look at the right

(a) Simple motion commands: the same direction keywords can cause different behaviors.

H1(Matsui): Kon-nichiwa. (Hello.)
 R: turn to the sound's direction, locate the person's face, and recognize.
 Konnichiwa, Matsui-san. (Hello, Matsui.)
 H1: Asoh-san wa doko? (Where is Asoh?)
 R: Honkan ni imasu. ([He is] in the main building.)
 H1: Yonde kudasai. (Please call [him].)
 R: Asoh-san ni me-ru wo okurimashouka?
 (Shall I send E-mail to him?)
 H1: Hai. (Yes.)
 H1: Motomura-san wa ? ([Where is] Motomura ?)
 R: Office ni imasu. ([He is] in [his] office.)
 H1: Tsureteitte kudasai. (Please take me [to his place].)
 R: Motomura-san no heya ni yukimasu.
 ([I'll] go to Motomura's office.)
 plan the path to the office, and execute the plan
 R: Motomura-san no tokoro ni tsukimashita.
 ([We are] at Motomura's office.)
 H2(Motomrua): Shokujii ni demasu.
 ([I'm] going out for lunch.)
 R: Motomura-san wa shokujii de yoroshii desu ka?
 (Is Motomura going out for lunch ?)
 H2: Hai. (Yes.)
 R: Schedule wo toroku shimasu.
 ([I'll] register [the data] to the database.)
 H1: Matsui mo. (Matsui, too.)
 R: Matsui-san wa shokujii de yoroshii desu ka?
 (Is Matsui going out for lunch ?)
 H1: Hai. (Yes.)
 R: schedule wo toroku shimasu.
 ([I'll] registrate [the data] to the database.)

(b) Zero pronoun filling in navigation and database tasks.
 Words in brackets are assumed.

Fig. 6. Office dialogue examples.

the attentional state. The answer given by Jijo-2 is the under-specified response “Honkan ni imasu ((he) is in the main building)”. Because it pronominalizes the current topic, this response is smooth and natural-sounding Japanese.

In the next sentence, the subject and indirect object of the verb *yonde* (call) are omitted in favor of zero pronouns. The most salient antecedent, which is *Mr. Asoh*, is assumed as the suitable referent. The *call-context* proposes sending an email message to Mr. Asoh. In the following sentences, other under-specified utterances are given appropriate antecedents or default values. This inference for referents is not only useful to make Japanese dialogue natural and speedy, but also to attain the better speech recognition.

7 Conclusion

In order for a mobile robot to provide services in an office communicating with people in spoken language, a method to maintain a topological map by dialogue and a natural Japanese dialogue system have been implemented on the Jijo-2 robot. The multi-microphone array played an important role to attain robust speech recognition in a usual office environment. Attentional state management and dialogue context management cooperatively worked to conduct smooth Japanese conversation which allows frequent omission of words. We could totally give over 100 demonstrations in three years without getting into a fatal crash.

Acknowledgment.

This research had been funded by Ministry of Economy, Trade and Industry of Japan as a part of Real-World Computing Partnership. Authors express sincere gratitude to Dr. Nobuyuki Otsu, Director of RWI Center, AIST, and to Dr. Jun-ichi Shimada, Director of RWC Tsukuba Research Center, for supporting the Jijo-2 robot project.

References

- Asano F, Asoh H, Matsui T (1999) Sound source localization and signal separation for office robot "Jijo-2". In Proc. of IEEE International Conference on Multisensor Fusion and Integration Systems.
- Asoh H (1999) Coping with uncertainty using interactions with humans. In Proc. of IJCAI-99 Workshop on Reasoning with Uncertainty in Robot Navigation (RUR-99), Stockholm.
- Asoh H, Matsui T (1999) A unified framework of map learning with a hierarchy of probabilistic maps. In Proc. of Field and Service Robots (FSR-99), Pittsburgh.
- Asoh H, Matsui T, Fry J, Asano F, Hayamizu S (1999) A spoken dialog system for a mobile office robot. In Proc. of Eurospeech-99, pp 1139–1142, Budapest.
- Asoh H, Motomura Y, Matsui T, Hayamizu S, Hara I (1996) Combining probabilistic map and dialogue for robust life-long office navigation. In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 807–812.
- Fry J, Asoh H, Matsui (1998) Natural dialogue with the Jijo-2 office robot. In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver.
- Grosz B, Joshi A, Weinstein S (1995) Centering: A framework for modelling the local coherence of discourse. Computational Linguistics, 21(2):203–225.

- Itou K, Hayamizu S, Tanaka K, Tanaka H (1993) System design, data collection and evaluation of a speech dialogue system. IEICE Transactions on Information and Systems, E76-D: 121–127.
- Matsui T, Asoh H, Fry J et al. (1999) Integrated natural spoken dialogue system of Jijo-2 mobile robot for office services. In Proc. of 16th National Conference on Artificial Intelligence, pp 621–627.
- Matsui T, Asoh H, Hara I (1997) An event-driven architecture for controlling behaviors of the office conversant mobile robot jijo-2. In Proc. of 1997 IEEE International Conference on Robotics and Automation, pp 3367–3371.

Intelligent Home Appliances

Henrik I. Christensen

Centre for Autonomous Systems,
Numerical Analysis and Computer Science
Royal Institute of Technology,
Stockholm, Sweden,
hic@nada.kth.se

Abstract. In this paper the use of robots as intelligent appliances is discussed. A number of advertised systems are reviewed and their basic characteristics analyzed. Open issues in terms of navigation, user interfaces, robustness, price are discussed as a basis for issues for future research to enable commercial delivery of such systems.

1 Introduction

Robotics is by now a well established domain for industrial automation. Yet, the major application area to open up is considered to be domestic robots (Karlsson, 2001). There are a number of fundamental differences between industrial and domestic robotics, that have to be considered to enable successful deployment of these new robots. Some of the differences include:

Environment The environment is much less constrained than in an industrial setting

Usability The system is to be operated by an untrained operator that might have limited or no computer experience

Energy The run-time for a system should preferably be close to 24 hours a day, in terms of availability, which implies that the system should recharge whenever nothing else is needed. I.e. ad hoc recharging rather than recharging when needed by the batteries

Price To gain widespread acceptance it is crucial that the price becomes low enough to allow acquisition by regular people for everyday tasks, which in general implies a price of less than \$1000.

The potential set of applications in a domestic setting involves a large number of tasks. The task domain can be divided into three categories:

1. Entertainment
2. Everyday tasks
3. Assistance to elderly and handicapped

The first category is where the majority of the business is today. The benefit of this category is that the performance metric is forgiving. I.e. the robot is not required to carry out specific tasks that requires accurate navigation and/or path following. As long as the robot does something interesting in terms of behavioural interaction the

customers will in general be satisfied. A challenge is here to have an “open-ended” repertoire of behaviours to maintain the interest of the user so that they are not bored after a few hours, days or weeks of use.

The second category is aimed at robot systems for everyday tasks such as vacuum cleaning, fetch-and-carry, ironing, window cleaning, These are extremely demanding tasks as the service to be delivered in general has a low monetary value and it has to be carried out in a rather complex setting. The key is here that the performance metric is well defined, while the domain is poorly defined. One wonders why industries want to enter this area.

Finally the area of assistance to elderly and handicapped is motivated by the demographic profile of the western world in general. Over the next few decades the society will have a rapid rise in the number of retired and handicapped people, which is one of the benefits of medical advances in terms of prolonged life-span and new possibilities to save life. Unfortunately to cope with the aging society it is necessary to maintain or increase the current level of productivity to ensure economic growth and in parallel it is necessary to provide assistance to elderly and handicapped so that they will experience an independent style of living (Wallace, 1999). A significant increase of the health sector might in part solve this problem, but it would be better if several of the services delivered to the person are automated so as to have independence. To this end it is also important to notice that tasks such as toilet visits can be assisted by robots without problems while it is less obvious that people are equally comfortable having the same tasks carried out by a human assistance. The performance metric is here a high degree of flexibility and easy instruction while the price potentially might be significant as the alternative would be managed care, which typically a gross price of \$100/hour.

Recently a number of robot systems for domestic use have entered the market or been advertised. Many of these new systems are “toy”/entertainment products, but there are also a number of service related products. A key factor for the widespread adaptation of these systems will be price. This in turn requires development of cost efficient methods in a price range that barely has been considered before. In this paper a few example robot systems will be considered and based on this general problems in domestic robotics will be identified as a basis for a discussion of limitations in current robotics research.

2 Example Systems

Two major categories of domestic robotics have been announced over the last three years, they are i) entertainment robots and ii) cleaning robots. A third category might be information management robots, or embodied IT appliances. For now we will, however, restrict our attention to the two initial categories. For these categories it is characteristic that most of the entertainment robots originate from Japan, while the cleaning/service robots all are from Europe. A brief sketch of a number of systems is provided below as a basis for a more details analysis.

2.1 Sony AIBO

The Sony Inc. AIBO dog pet has been released in two generations. Both are quadruped dog type toys with a camera build into the nose, a microphone for sound analysis and a loudspeaker for “simple” feedback. The system comes with predefined behaviours for basic motion etc. In addition the commercial model has a build in learning method for “adaptation” to user preferences. The robot is battery powered and uses a MIPS M4000 300 MHz processor with 32MB memory. The robot is programmed using the Open-R proprietary operating system from Sony. Limited information has been made available on the system layout, but it is evident that it uses a behaviour based method of programming (Arkin, 1998). A special version of the AIBO has been released for use with the RoboCup competition. A programming kit using a “finite-state automata” model is available as an add-on. Programmes can be uploaded to the dog through a memory-stick. Various version of more advanced programming kits have appeared on the internet, these programmes have, however, been retracted after Sony has indicated copyright issues and threatened to take legal actions.

2.2 Sony SDR-3X

Recently Sony Inc. advertised (Nov. 21, 2000) the SDR-3X robot as the “robotic partners”. This is a small humanoid style robot (biped) that has 21 degree of freedom with the ability to walk and “dance”. The robot has a height of 50cm, and a cross ratio of 22 x 14 cm. It has a weight of 5 kg, and a walking speed of 15 m / min (25cm/s). The robot has build in monocular vision system for motion based recognition, and microphone and loudspeaker for speech interactions. In addition the system has a built in 802.11 network connection. The robot is controlled using a “remote-brain” approach (Inaba, Kagami, Kanehiro, Hoshino, Inoue, 2000), where motion initially is planned/programmed in a PC based simulation system and subsequently downloaded to the Open-R system that resides on the SDR-3X system. At present it is not known in detail what type of programming interface that will be offered.

2.3 NEC M100

The NEC M100 robot is a small tricycle robot construction that is driven by a pair of differentially controlled wheels. The robot navigates using a set of ultra-sonic sonars. The robot has also a built in camera system that is connected to a face recognition system. Finally the system has a microphone and loudspeaker for speech recognition and replay. The robot is in principle a roving answering machine that automatically recognises the occupants in a house and replays messages to them upon encounter. It is not evident that robot can be programmed to carry out specific actions.

2.4 iRobot

The company iRobot has announced the iRobot platform for domestic and office use. The robot is basically a web-cam on wheels. The robot has a video dome with a build-in web cam sitting on a neck that can be raised and lowered. The head has a $\pm 45^\circ$ motion capability of the camera. The head is mounted on a set of 8 wheels that are mounted on an articulated body (1 DOF) so as to enable traversal of “simple” staircases and minor obstacles. In addition to the earlier mentioned camera the robot has microphone and stereo speakers for tele-conferencing. The robot is supposed to run Linux for basic control and a tele-conferencing system for remote control. The idea is to allow home owners to navigate around their home while away, and to use the robot as an embodied character for professionals that have to attend video conferences. So far (Spring 2002) the company has taken orders but not yet delivered any systems.

2.5 Dyson – DC06

The Dyson DC06 robotics vacuum cleaner was announced by summer 1999. The idea is to have a fully autonomous vacuum cleaner for home environment. The robot has a differential drive system and 2 caster wheels that are suspended. The robot is controlled by 3 computers (type 68xxx). In addition the system is equipped with 50 different sensors (a mixture of sonars, encoders and accelerometers) to allow structured traversal of a specific region. The robot is battery powered and it is not apparent if it comes with a recharging unit. The robot is supposed to have a running time of about 1 hour. It is not expected that it will be possible to programme the system.

2.6 Kärcher – RoboClean

The Kärcher RoboClean vehicle is a differential drive system for random traversal of rooms (Prassler, Ritter, Schaeffer, Fiorini, 2000). The robot has a weight of 1.4 kg. The robot has bumpers for detection of collisions but not other methods for mapping of the environment. The robot comes with a pollution sensor that allows control of suction power and speed in proportion to the amount of dirt picked up from the floor. The robot covers a particular region using a random motion pattern. The run-time between recharging is 20-30 minutes. The robot comes with a structured method for automatic docking with the recharge station (phase based sound direction estimation). The performance is $\sim 15m^2/h$.

2.7 Electrolux – Trilobite

The Electrolux vacuum cleaner is another differential drive robot system. It has a weight of about 2 kg, a diameter of ~ 40 cm, and a height of ~ 12 cm. It features a number of casters in addition to the drive wheels. The robot has a 180° sonar sensor for mapping of the environment. In addition the robot has a shock absorbing

bumper for handling of small objects (Prassler et al., 2000). The robot has a run time of about an hour between recharging. The robot uses a semi-structured approach to traversal of rooms. Initially the robot will drive around the boundary of the room and upon completion of the cycle it will move at random through the rest of the room (Prassler et al., 2000). The robot comes with a recharging circuit that allows extended operation for hours at a time. It is not evident that it will be possible to programme this system. It is further not obvious if the system will have a traditional user interface.

2.8 Siemens/Hefter – ST81 VarioTech

In a collaboration between Siemens AG and Hefter Cleantech a robotics floor scrubber has been developed as a example of a new brand of floor cleaning systems. This is not really a domestic system, but more clearly aimed at the professional market. The robot is a differential drive system with 4 additional caster wheels. The robot has a weight of about 800 kg and a run time of a few hours. The robot is about 80 cm high and covers a ground area of 1.5×0.7 meters. The robot features 24 ultra-sonic sonars and a SICK laser scanner for mapping and navigation. The robot can move at a speed up to 0.5 m/s. The user interface is through a single line LCD. In addition the programming of the robot is through specification of an area, that is to be covered using a Zamboni pattern.

2.9 Analysis

A quick review of some of the most dominant/credible systems about to enter our houses reveals several things. For the purpose of a more in-depth analysis the major characteristics of the system are summarised in table 1.

Table 1. Summary of robot system characteristics

Manuf.	Name	Run-Time	Navigation	User-If	Cost (est.)
SONY	AIBO ERS	~ 30 min-1 h	Prog. Walk	Audio, Behaviour	~ \$1500
SONY	SDR-3X	1 hour	Prog. Walk	Audio, Behaviour	~ \$5000
IRobot	IRobot	2 hours	Tele. Oper	Video/Audio	~ \$2500
NEC	M100	3-4 hours	Track/Sonar	Audio, Face Rec.	Unknown
Kärcher	RoboClean	20-30 min	Bumper	None	~ \$1000
Electrolux	Trilobite	1 hour	Sonar/Bumper	LCD	~ \$1200
DYSON	DC-06	45 min	Sonar/INS	None	~ \$4000
Siemens/Hefter	VarioTech	2 hours	Sonar/Laser	LCD	~ \$10000

The table (1) reveals a number of issues. First of all the price for a home robot is still beyond \$1000 for the majority of the units, which implies that the true mass

market is still beyond reach. In addition most of the systems have very limited facilities for navigation (most of them have either no facilities for absolute navigation or is based on ultra-sonic sonar systems). Finally the user interfaces are either of the entertainment variety (lot of activity/relative low information contents) or a “primitive” LCD display with minimal feedback. Further it is noticeable that the run-time for most of these systems is very limited and a most on the order of an hour. The weight/run-time ratio is still too high for many applications.

3 Analysis/Synthesis

3.1 Localisation and Mapping

To make truly useful robots it will be necessary to equip them with facilities for versatile navigation in rich environments such as a regular house, that has obstacles such as door thresholds, clothing on the floor, furniture, ... To be useful in such environments the robots must be able to perform automatic mapping and localisation in these environments, which requires methods such for “Simultaneous Localisation and mapping” (Austin, Jensfelt, 2000; Castellanos, Montiel, Neira, Tardos, 1999; Dissanayaka, Newman, Clark, Durrant-Whyte, Csorba, 2001; Feder, Leonard, Smith, 1998; Leonard, Feder, 2000; Thrun, Fox, Burgard, Dellaert, 2001). A problems with most of these approaches is unfortunately a relatively high computational cost that is close to prohibitive for domestic systems. There is here a need for inexpensive methods that can run on a standard DSP type processor with limited need for high dimensional linear algebra or similar. Simply optimization is not considered adequate for construction of such systems. Thus, while there might exist algorithms to perform SLAM in specific setting the problem is by no measure solved.

3.2 User interfaces

It is further characteristic that systems are designed either for no interaction with the user or very limited feedback. It is not yet obvious how a user is supposed to give “unconstrained” commands to a mobile platform. There has been significant research on human-computer interaction and often speech and/or gestures are used. Unfortunately there are few if any speech systems available that can manage large vocabularies and provide enough robustness to facilitate a true dialogue with an (inexperienced) user. First of all it is essential to be able to generate feedback to the user beyond simple line style LCD displays. In addition it is essential to recognize that the user might not be standing next to or in front of the robot all the time, which implies a need for flexible user interfaces (Christensen, Hüttenrauch, Severinsson-Eklundh, 2000; Thrun, 2000). In addition the addition of methods for gesture recognition is complex, non-robust and expensive (Christensen, Krägic, Sandberg, 2001; Waldherr, Momero, Thrun, 2000). Thus there have to be strong arguments for considering how and if there is a need for direct interaction. It might here be possible to build systems with well defined vocabularies, which are used in “word spotting”

mode to allow for fairly general interaction. In addition it might be useful (for some types of robots) to have pattern matching methods for face recognition for identification of different types of users, but here only if the functionality can be provided at a very limited cost.

3.3 Everyday use of robots

Almost all of the research systems that are reported in the literature worked on the course of a few hours or a few days. Most of them never ran more than a couple of hours continuously. This is in part due to battery problems. Today it is possible to provide “intelligent” recharging systems that allow one to operate a robot for days or months (Nourbakhsh, 1998). Once a system is to operate over extended time periods it is essential to consider automatic map adaptation, handling of memory, recovery from (minor) failures, ... These are problems that often are neglected in research systems and several commercial system can only operate for very limited periods or within narrow operational bounds. To gain widespread acceptance it will be necessary to setup and run *long-term* experiments. To this end it might be useful to have standard benchmarks for comparison of performance and evaluation of algorithms and/or integrated systems. This is a field that only is about to emerge.

3.4 Mobile Manipulation

Few if any of the systems available today or in the short term have facilities for mobile manipulation. I.e. all system have facilities for navigation. To perform mobile manipulation there are two options: i) to engineer the object so that it is easy to locate and pickup, or ii) to provide the robot with facilities for recognition of objects. The latter option is a truly hard problem in computational vision. Some progress has recently been reported for systems that utilize statistical learning theory (Roobaert, 2001), it remains however to be demonstrated that these methods will generalize to large collections of objects that are placed in general environments where there are variations in illumination, shadows, clutter, viewing angle, Finally there is a need for flexible grippers that will allow pick-up and manipulation of a significant number of different objects. The grippers available today are either extremely expensive or lack adequate sensor or actuation facilities for use with several domestic objects.

3.5 Mobility

Traditionally mobile robotics has been designed as wheeled systems. Yet, there are few standard wheeled platforms around as commercial products. Recently the locomotion problem has also been extended to legged systems in particular in relation to R&D in Japan. Walking is, however, much harder than wheeled locomotion and it introduces additional complexity in terms of balancing, gait selection, etc. It is, however, important to consider that small domestic robots, such as the previously

mentioned vacuuming robots, will have difficulties handling many everyday objects such as staircases, clothing, and toys. Relatively to the size of the robot these are large objects that have to be attended to. In particular staircases poses an interesting challenge as it might be fatal to a robot to interact with a staircase. Thus wheeled locomotion might still be the easiest solution to implement, but there is a need for additional methods to ensure robust handling of both obstacles and lack of ground support.

3.6 Intelligent Environments

Today's robots are largely built for autonomy, i.e. under the assumption that it will use its own set of sensors, its own computational resources. A notable exception is the work by Inoue et al using the "remote brain" approach (Inaba et al., 2000). In parallel to the introduction of robots there is also a constant diffusion of other appliances and aids to the domestic market. This includes burglary alarms (ir-detectors and cameras), web servers, personal computers, PDAs, etc. With the introduction of inexpensive means of communications it is not obvious that all methods have to be integrated on the robot. It might be beneficial to exploit existing infra-structure for the operation of the robot to reduce price, optimize energy consumption, etc. Recently a number of new electronic components have appeared with ...

4 Summary

A number of service robot applications have been advertised and there are great expectations as to the use of such systems. So far few have managed to deliver systems for operation in regular houses. The only widely available systems are the SONY AIBO and the Electrolux Trilobite.

It is evident that a number of fundamental problems will have to be addressed to allow delivery of *robust* systems that can be used by regular people at a reasonable cost.

In this paper a number of the advertised systems have been analyzed and major issues to be resolved for wide spread use of robots have been identified. The UN World Robotics (Karlsson, 2001) predicts that the market for service robots will grow by an order of magnitude over the next five years, yet it is not obvious that the technology is mature enough to meet such a demand.

Acknowledgments This research has been carried out within the Centre for Autonomous Systems, that is a facility sponsored by the Swedish Foundation for Strategic Research. The support is gratefully acknowledged. A number of the issues presented in this paper has arisen out of discussions with the PhD students and post-doctoral researchers affiliated with the centre. The support and encouragement of the involved researchers is much appreciated.

References

- Arkin RC (1998) Behaviour based robotics. In Intelligent Robots and Autonomous Agents. MIT Press, Boston. ISBN 0-262-01165-4.
- Austin D, Jensfelt P (2000) Using multiple gaussian hypotheses to represent probability distributions for mobile robot localization. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA'00), vol 2.
- Castellanos JA, Montiel JMM, Neira J, Tardos JD (1999) The spmap: A probabilistic framework for simultaneous localization and map building. *IEEE Trans on Robotics and Automation*, 15:948–952.
- Christensen H, Hüttenrauch H, Severinsson-Eklundh K (2000) Human-robot interaction for service robots. In Robotik-2000, pp 315–324. Keynote.
- Christensen HI, Kragic D, Sandberg F (2001) Vision for robot interaction. In M2VIP.
- Dissanayaka MG, Newman P, Clark S, Durrant-Whyte H, Csorba M (2001) A solution to the simultaneous localisation and map building problem. *IEEE Trans on Robotics and Automation*, 17:229–241.
- Feder H, Leonard J, Smith C (1998) Adaptive concurrent mapping and localization using sonar. ICRA-98, 2:892–898.
- Inaba M, Kagami S, Kanehiro F, Hoshino Y, Inoue H (2000) A platform for robotics research based on the remote-brained robot approach. *International Journal of Robotics Research*, 19:933–954.
- Karlsson J (2001) World robotics 2001. United Nations Press, Geneve.
- Leonard JJ, Feder HJS (2000) A computationally efficient method for large-scale concurrent mapping and localization. In Robotics Research, Berlin. Springer Verlag.
- Nourbaksh I (1998) URL <http://www.cs.cmu.edu/~illah/SAGE>.
- Prassler E, Ritter A, Schaeffer C, Fiorini P (2000) A short history of cleaning robots. *Autonomous Robots*, 9:211–226.
- Roobaert D (2001) Pedagogical support vector learning: A pure learning approach to object recognition. PhD thesis, Electrical Engineering and Computer Science, Royal Institute of Technology, Stockholm, Sweden.
- Thrun S (2000) Interaction with mobile robotics in public places. *IEEE Intelligent Systems*.
- Thrun S, Fox D, Burgard W, Dellaert F (2001) Robust monte carlo localisation for mobile robots. *Artificial Intelligence*, 128:99–142.
- Waldherr S, Momero R, Thrun S (2000) A gesture based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–174.
- Wallace P (1999) AgeQuake: Riding the demographic rollercoaster shaking business, finance and our world. Nicholas Brealey Publishing Ltd., London. ISBN 1-85788-192-3.

Part 7

Field Robots

Session Summary

Chuck Thorpe¹ and Hugh Durrant-Whyte²

Chairs

¹ Robotics Institute, Carnegie Mellon University, Pittsburgh USA

² Australian Centre for Field Robotics, The University of Sydney, Sydney NSW 2006, Australia

Abstract. Field Robots are machines that work in unstructured environments, including under water, in mines, in forests and on farms, and in the air. These applications involve both advanced ideas in robotics and careful attention to engineering details. This paper discusses the overall challenges of field robotics and the current state of the art. The paper is not a thorough survey, nor a tutorial; but is instead intended to serve as a discussion starter for further design and development.

Introduction

Field Robots is concerned with the automation of vehicles and platforms operating in harsh, unstructured environments. Field robotics encompasses the automation of many land, sea and air platforms in applications such as mining, cargo handling, agriculture, underwater exploration and exploitation, highways, planetary exploration, coastal surveillance and rescue, for example. Field robotics is characterized by the application of the most advanced robotics principles in sensing, control, and reasoning in unstructured and unforgiving environments. The appeal of field robotics is that it is challenging science, involves the latest engineering and systems design principles, and offers the real prospect of robotic principles making a substantial economic and social contribution to many different application areas.

In general, field robots are mobile platforms that work outdoors, often producing forceful interactions with their environments, with no human supervision. Classic examples are backhoes that automatically load trucks, or automated bull-dozers that clear fields. Other examples don't quite meet all the criteria, but are still considered field robots. Automated underground mining machines are not technically operating outdoors, and are operating in semi-structured environments, but still share most of the technical challenges of other field robots. Robot cranes are not necessarily mobile in the usual sense, but still have to reach over wide areas, sense obstacles, deal with vagaries of outdoor illumination and wind gusts, and otherwise handle field robotics issues. Military scout robots are not (supposed to) perform forceful interaction, but do deal with highly irregular terrain.

A major impetus in field robotics is the comparative advantage provided by automation of large expensive manned machines (a mining haul vehicle or a container handling crane) over the automation of low cost domestic systems (a vacuum cleaner or product mover for example). A typical automation package may cost of the order

of US\$50 – – \$100,000. While this is a substantial increase in cost for a typical product mover (AGV), it is quite modest in comparison to the \$2M cost of a mining haul truck. Additionally, the relative pay-off in increased equipment utilization, reduced maintenance and fleet controllability is itself a major motivation for automation.

The last ten years have seen substantial real progress in development and implementation of field robotics systems. Indeed, in the next few years there is a very good chance that substantial commercially available field robotics systems will go into operation. Consequently, there is a high degree of excitement in the area about the future prospects and opportunities in field robotics.

In this paper, we try to take a broader view of the field than is offered by any single application. We structure our discussion around categories of difficulty: what technical problems are fairly well understood and can be used in applications? Which areas are partially understood, and are somewhat useable, but still require significant research? Finally, which areas are still difficult problems, where a solution can perhaps be kludged up but for which no solid engineering solutions are readily available? This paper is not meant to be a survey of the state of the art, but rather a discussion starter; we cannot hope to adequately cover the growing body of literature and increasing number of projects. (Major conferences that cover the topic include the Field and Service Robotics Conferences, as well as the International Symposium on Robotics Research). Instead, we concentrate on big emerging themes, and illustrate those areas with projects with which we are familiar.

Things we understand

Field robotics has made considerable progress in the past 10 years with a significant number of systems now deployed in both advanced research and near-commercial applications. Technically, field robotics has been able to make best progress in applications where the environment is cooperative, constrained, and, ideally, structured for automation. Typical of such applications are cargo container handling, haulage in mining, and high altitude unmanned air vehicles (UAVs). Field robotics is as much about engineering as it is about developing basic technologies. While many of the methods employed derive from other robotics research areas, it is the application of these in large scale and demanding applications which distinguishes what can currently be achieved in field robotics. Broadly, field robotics technologies can be divided into three main areas; navigation and sensing, planning and control, and safety.

In the sensing and navigation area the problem of position determination (localization) has been the subject of considerable work. GPS has made the localization problem tractable in many field robotics applications. This is particularly true in environments where good views of the sky can be guaranteed (in surface mining, in UAV or autonomous ship systems for example). The development of low cost inertial sensing has also been a major advance for these applications. Combined GPS inertial systems are now in common use in many field robotics applications. In ap-



Fig. 1. Automated container hauler

plications where GPS availability is poor, such as port cargo handling (the cranes get in the way!), or non-existent, such as underground mining or indoor vehicles, the localization problem is not so easily solved. In such situations there has been a common use of lasers and mm-wave radar to observe the relative location of artificial landmarks or beacons and, by referencing these to a map, to deduce position. The use of artificial landmarks together with appropriate position estimation algorithms is now well understood. The use of natural landmarks for relative navigation is also reasonably well developed. Good examples include the use of vision to delineate crop lines in agricultural applications, the use of vision to maintain an automobile on a highway, or the use of lasers to determine approach vectors for vehicles in cargo and mining applications.

Over the past 10 years or so, we have become better at controlling the motion of large vehicles and of planning both small and large motions for these machines. At the beginning of the 1990's, speeds of a few meters per second were typical. Now, it is not uncommon for autonomous land vehicles to achieve speeds of $60 - 100 \text{ Kmph}$. There are two basic reasons for this. First, processing of navigation, particularly visual, information has become much faster and can now easily achieve the required bandwidths necessary for high-speed vehicle control. Second, we have got much better at modelling the complexity of high-speed vehicle motion, particularly the effect of pneumatic tires or tracks, and the interaction of vehicles and rough ground. Many complex land vehicle control problems have now been solved. Planning motions of large vehicles has also been tackled with some success. At the simplest level, issues such as linear route planning and scheduling of vehi-



Fig. 2. Automated Straddle Carrier

cles has been implemented on a number of commercial field robotics systems. The planning of motions on rough 3D terrain has also received considerable attention although this is not yet a closed book. The planning of vehicle missions, in air or sub-sea vehicles for example, has also made progress although there is some way to go to make these methods generally applicable.

Safety is a major (and in some ways, unique) element of field robotics applications. Large vehicles travelling at high speed pose a significant health hazard. There are a number of elements to this problem. First is the issue of collision detection. Some quite advanced collision detection systems exist in advanced research or near commercial stages of development. These include mm-wave radar in automobile and mining applications, and lasers or radar in cargo handling or air-borne applications. A major requirement in field applications is long range (for stopping distance from high speed), all weather operation and reliability. It is the reliability issue which is hard to come to grips with. In a rough terrain environment it is not easy to obtain the sensitivity and discrimination necessary to distinguish a potential collision from a simple artifact of the environment. A second major safety issue is the problem of overall system integrity. The cost, size and speed of vehicles in

many applications means that some guarantee must be made about the operation of the machine over its operating life. A 15 minute demo is not enough. The challenge is to ensure continuous operation 24 hours a day 365 days a year. For autonomous systems, the main issues are in sensor and software integrity. Safety is understood although there are still some very complex and challenging issues to overcome.

Given a reasonably structured, ideally open air environment that can be controlled to a degree, field robotics is now able to deliver operational systems.

Things we kind of understand

As the environment becomes less structured or the interactions between machine and environment become more complex, we understand less about what we are doing. Examples of such applications include autonomous reconnaissance in unstructured terrains, excavation in mining or construction, sub-sea navigation and low altitude UAVs.



Fig. 3. Underwater robot for reef exploration

Probably the most complex and demanding research issue facing field robotics is the full 3D perception and understanding of typical unstructured environments. This may include the problem of land-vehicle terrain estimation from sensors such as vision and lasers, or equivalent problems with sonar underwater and with air-borne laser and radar systems. Such terrain models will be an essential component of functions such as navigation, path and mission planning in unstructured domains.

While there is, and has been, a great deal of research currently being undertaken in these areas (outside and inside field robotics), the reliable construction and understanding of terrain models is still some way off. This is for two main reasons. First, we need a general representation of such environments capable of dealing with the intrinsic uncertainty (location, structure and data association) together with the broad (unpredictable, time varying, multi structured) types of terrain likely to be encountered by field robotics machines. A number of terrain construction methods have been employed with varying degrees of success. These include the use of grids, triangular tessellations, and probabilistic methods. Second, the problem of terrain understanding remains a major hole in current research effort. Terrain understanding really involves the manipulation, reasoning and dynamic use of terrain models in navigation and planning. While there has been some notable work (on grids for example), these fall a long way short of the necessary genericity required in typical field applications. To some degree the problem of terrain understanding is not even well posed until there is some better consensus on the methods required for terrain construction. While we understand what is needed, the necessary methods have yet to be developed.

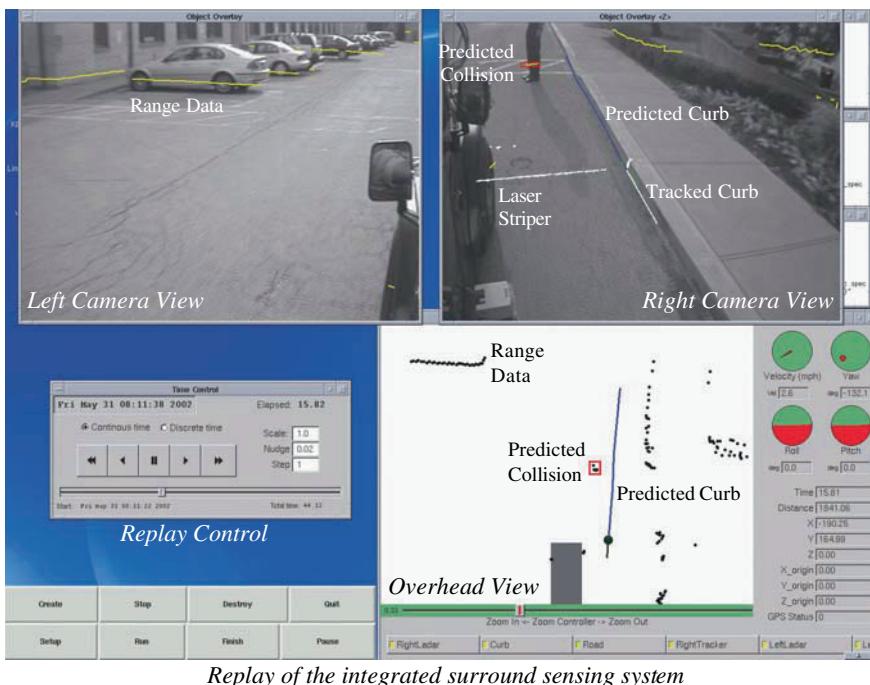


Fig. 4. Safe driving in cluttered environments using multi-sensor fusion

A second major issue that we have yet to come to grips with is how to plan and make robust decisions in environments with little structure or model. Fundamen-

tally, the problem is concerned with making decisions in the presence of structural uncertainty or error; placing a foot on terrain of unknown geometry and composition, planning a path through an area with incomplete geometric or structural knowledge. This is the essence of working in field robotics where the environment can not be known and will not conveniently fall into some standard model. While there are well-paraded mathematical methods for formulating such problems it has been hard to see how these can reasonably be applied in even simple field environments. A demonstration in which a machine can autonomously make decisions such as these, and be equally effective in a variety of domains, is a long way from our current achievements. It may be that less traditional approaches (controller compositions, qualitative modelling and reasoning methods) will be a better way of dealing with this type of structural uncertainty. Again, there is a general understanding of the problem, but a concerted effort to formulate a method is still required. As always, substantive implementations are also necessary.

There are several other significant issues which the field robotics community has been addressing, but which still lack a basic understanding. Notable, is the general problem of contact control and environment modification as typified by autonomous excavation. Here, a large (very flexible, low bandwidth) manipulator comes into contact with a poorly known environment (variable geometry and material composition), then interacts and changes that environment in a manner which is unknown and can not often be predicted with any degree of accuracy. It is not at all clear how to deal with this problem in general. However, it is worth pointing out (maybe a lesson to be learned?) that the simple solution, so far, seems to work; just dig until certain forces and motions are detected, and carry on regardless.

The problem with lack of structure is at the heart of many field robotics applications. Once we have a grip on how to describe and model this lack of structure and once we can make decisions robust to these models, many new field applications will become solvable.

Things we mostly don't understand

The “grand challenge” problem for field robotics is:

Create smart, reliable, mobile machines, capable of moving more capably than equivalent manned machines in a wide variety of unstructured environments.

Current field robots all fall well short of that goal, on at least one axis. Most field robots operate at a much lower performance than would be dictated by the raw hardware; usually, this is driven by limitations in perception and judgment rather than limitations in control. Humans are still much better than robots in driving over rough terrain, maneuvering through heavy traffic, detecting small obstacles by texture or color cues, and general visual recognition. Humans excel in ‘common sense’; a combination of general-purpose perception, reasoning about the behavior of other

entities (specifically other people), planning in the face of uncertainty, and generating reliable and redundant systems.

A typical example is the electronic chauffeur of the future. Autonomous driving research has focused on perception of the road and of other vehicles, and accurate vehicle controls. The resulting vehicles have a wide variety of capabilities, for road following, headway maintenance, lane changing, and so forth. But moving from an impressive demo system to a truly functional vehicle will require significant further advances, notably in reliable perception and in modelling the behavior of other drivers.

Specific challenges for continued research in field robotics include the following, with an example task given for each challenge:

- General recognition: Perception systems are becoming adequate for well-defined tasks and well-defined objects. More general recognition is still difficult. Parts of the problem could be solved by better sensors or innovative combinations of sensing. Other parts of the problem sit at the intersection between sensing and world knowledge; the classical problem of recognizing a chair requires both shape measurement and reasoning about object use and social conventions. Example task: an automated car distinguishing between radar returns from another car vs. from a Mylar balloon blowing into the highway.
- Dealing with terrain: Perception works adequately for some indoor mobility tasks. Existing sensing systems are inadequate for driving over outdoor terrain. New sensors or sensing strategies need to be developed for detecting surface conditions (mud, ice, loose gravel), inferring partially-observable surfaces (rocks and ditches covered by grass), and estimating other surface properties (bearing strength of dirt, slip angle of sand). Example task: automated dune buggy deciding how to climb a steep hill.
- Building human interfaces: While some robots will be autonomous, many will work cooperatively with a human supervisor. There is a long and deep literature on pure teleoperation, where a human controls a robot directly for example via joystick. There is an emerging field of graphical interfaces, with good remote representations of current vehicle state. But there is very little work done on high-level interfaces between robots and humans. Users would like to generate commands at a variety of levels, including explicit joint commands; movement commands; safeguarded moves; subtasks; and task sequences. The robot in turn should be able to ask for clarification; report on problems and suggest alternatives; and understand what level of deviation is allowed from the commands. This work will require new results in dialogue modelling and planning systems, as well as more traditional robotic, graphics, and user interface studies. Example task: working with a lunar exploration robot over a link with time delays, in order to accomplish a prospecting task while safeguarding the robot.
- Dealing with humans: Robots operating in proximity to people need to be able to sense people and guarantee safe actions. Parts of this task involve sensing: which objects in the environment are people? How are they moving? Where are they looking? Parts of the task involve social conventions: will this person move

to the right or the left when she meets me on the sidewalk? How likely is a driver to use a turn signal when changing lanes in front of an automated car? Example task: robot cleaning machine in a crowded grocery store, deciding when to go around people, when to wait for them to move, and when to politely ask them to clear a space.

- Legged systems: Rough terrain and narrow passages are best traversed by legged systems. Truly efficient and reliable legged locomotion requires fundamental advances in mechanisms, control, gait and footfall planning, and perception. Inspiration can come from biomimesis, but actual mechanisms may be different from biology. Example task: robot mule.
- Learning: Most mobile systems are brittle: if conditions change, or if they are asked to perform a task out of their design envelope, they are unable to cope. Systems of the future will need to use learning at many levels. Map learning is already being addressed. Example areas that need additional work include learning controls for extreme maneuvers, or changing controls as the system ages; learning behavior patterns of other agents; and learning new sensing strategies and encapsulated behaviors. Example task: adapting helicopter control as it picks up a load swaying in the wind.
- Cooperation: Until recently, most laboratories were fortunate to have a single functioning robot, so multi-robot research was infeasible. There is a large upsurge in multi-robot cooperative planning research, most visibly in the context of robot soccer. While some interesting performance has been achieved, there is still a great deal of work to be done on the fundamental principles of multi-agent cooperation. Subproblems include deducing another agent's plans from partial observation; problem decomposition without hierarchies or without communication; cooperative sensing; and cooperative mobility. Example task: multiple robot explorers in unknown terrain with difficult mobility, co-operating to examine all of the terrain in minimal time.

Conclusion

Field robotics is the most difficult, but most exciting, area of robotics. In terms of economic impact, field robots have arrived at a stage where they will soon be productive members of society, beginning to handle dirty and dangerous jobs and to be real contributors to the economy. In terms of scientific inquiry, robots that move through the real outdoor world, with all its unpredictability, offer a direct experimental testbed for developing truly intelligent machines. And in terms of public excitement (and our own motivation), it is fascinating to build machines that can extend our senses under the ocean, on the surface of Mars, or deep underground.

There are immediate applications, both in terms of complete systems and in terms of robotic components used in other applications. But there is also a rich set of open problems, in fundamental research as well as in applications, that will keep us all busy well into the future.

References

- Bares J (ed) (1999) Second International Conference on Field and Service Robotics, Carnegie Mellon University.
- Giralt, Hirzinger (eds) (1995) Seventh International Symposium on Robotics Research, Herrsching, Germany. Springer-Verlag.
- Halme A (ed) (2001) Third International Conference on Field and Service Robotics (FSR '01), Helsinki University of Technology, Helsinki, Finland.
- Hollerbach J, Koditschek D (eds) (1999) Ninth International Symposium on Robotics Research, Snowbird, Utah, USA. Springer-Verlag.
- Shirai, Hirose (eds) (1997) Eighth International Symposium on Robotics Research, Kamakura, Japan. Springer-Verlag.
- Zelinsky A (ed) (1997) First International Conference on Field and Service Robotics, Australian Robot Association.

Part 8

Vision-Based Navigation

Session Summary

Yoshiaki Shirai

Department of Computer-Controlled Machine Systems, Osaka University

Vision based navigation can roughly be divided into the following two classes:

- Navigation without maps.
- Navigation with maps.

Navigation without map requires two subtasks: map building, and motion planning. The former has been intensively studied for a few decades. A typical way is so called SLAM (simultaneous localization and map building), which uses dead reckoning and visual observations (see papers in another session).

Motion planning is necessary to reach a goal. An important problem is how the goal is specified. The goal can be pointed if it is seen from the starting position. If not, the goal can be specified using a rough topological map or specified in terms of properties of the goal. Another way is to manually guide a robot to the goal.

Navigation with a map is dealt with in this session. A problem is how to find correspondence between a given map and the environment which is observed by vision. For this purpose, prominent features of the environment are observed. Those features are called landmarks.

The navigation with a map can further be divided into the following classes according to available landmarks:

- (a) All landmarks are observed everywhere.
- (b) Only a part of landmarks are observed at a position.
- (c) No landmarks are given.

In case (a), the problem of finding landmarks is easy because they can be tracked. Such landmarks are equivalent to beacons emitted from known positions. Even if many landmarks are observable, three of them is enough to determine the position in the horizontal plane.

Peter Corke proposed a method similar to visual servoing because visual servoing is more efficient than solving geometrical problems.

The discussions on his presentation are:

Thorpe: A proof of convergence should not be too hard.

Dickmanns: The possibility of dynamic feature selection to optimize a Jacobian had been previously explored by Wunsch.

Nebot: When moving between local navigation regions some sort of landmark switching strategy would be required.

In case (b), visible landmarks changes as a robot moves. The important problems are what kind of landmarks should be used and how to find landmarks at each position.

Santos-Victor addresses those problems. He proposes the use of a topological map of touristic landmarks. Given a goal, the robot moves along a road or a corridor

while looking for the predicted landmarks. This is more effective than making a geometric representation of the environment. He also proposes to take the advantage of biological vision systems which are proved to be successful in different navigation tasks.

The discussion on his presentation are:

Chatila: The approach of using image motion (optic flow) for the control of a mobile robot is interesting from the viewpoint of demonstrating the biological plausibility of such models. However it seems to require massive modification to the environment such that their practical use is limited.

Answer: One of the examples shown may have lead to that impression since the environment had been modified to have highly textured artificial patterns. However, the reason is that this particular example has about 8 years old when the available computational power would require strong textures. We can currently apply the same techniques with higher resolution images at video rate. Examples were shown for the underwater robot control or for the binocular head.

Shirai: How important is the log-polar transformation? For example, when navigating in a corridor, many distant landmarks are useful.

Answer: Indeed, many distant landmarks are useful in the context of navigation. The omni-directional camera still suffers from perspective effects, so that distant objects will always remain smaller in the image. The advantage of the panoramic image is the uniform angular resolution. Related to the question, it is interesting to note that the optimization of the sensor radial distribution versus the mirror profile has some redundancy. For instance, one could use a hyperbolic mirror and determine the sensor distribution or do a joint optimization of both the sensor layout and mirror profile.

Featherstone: In the human retina, the photoreceptor in the foveal region have different characteristics of those located in the periphery, in terms of sensitivity to light and color. Have these aspects been taken into account in the log-polar sensor?

Answer: Indeed, there are two types of photoreceptor: cones and rods. Cones provide color vision and rods provides gray vision under low illumination. In the human eye, there are very few cones in the periphery of the retina. The central foveal is densely populated with cones. These aspects are not covered in the log-polar sensor.

Zelinsky: There have been other ways of producing image geometries similar to the log-polar by using mirrors instead of a special sensor.

Answer: Other ways of producing space variant resolution images is to use lenses, although there are some difficulties with severe brightness changes across the entire image. There have also been other efforts of building mirrors such that the panoramic images have some desirable properties: constant resolution. However, only a polar sensor can realize the equal angular resolution at no additional computational cost.

In some biological systems such as that of bees, the goal position is specified by other bees. The task is to fly toward the goal direction and to land at the goal. Srinivasan already presented a theory that utilizing information on optic flow, a bee can fly in the center of a narrow channel.

This time he investigates how to control the flying velocity for a smooth landing. The theory is similar to the previous one: a smooth landing is realized by keeping optic flow constant while approaching to the goal. This is a simple and beautiful theory which is proved by experiments. Because of its simplicity, the landing method may be applied to controlling flying vehicles.

On the other hand, Hartley deals with a problem of measuring the angle between two lines in a 3-D space. He proposes a mathematical method to precisely measure the angle by utilizing a horizon or an estimated horizon. This is effective for finding landmarks of artificial objects because their image usually have straight lines which meet a fixed angle.

The discussion on his presentation are:

Shirai: Since you have the horizon, is it not possible simply to compute a rectified image of the plane by synthetically rotating the view direction?

Answer: Yes. There are other ways of carrying out this task. One of the ways is to compute 2-dimensional projective transformation (homography) that will rectify the plane to a geometrically-correct Euclidean view. In doing this the difficulty is that one either needs to know the camera calibration accurately, or else to carry out a full 8-degree of freedom homography computation. In our method, we do not need to know a full calibration of the camera. Computation of the conformal point is simpler and more stable than computing the principal point and focal length of the camera. Computing a full 8-degree of freedom homography is not an ideal solution, either, since the problem of over-fitting the transformation arises. Knowing the conformal point, the correct homography relies on only 3 parameters.

Jarvis: It seems that you have invented a new method of doing odometry. Could you indicate some of the advantages of this method with respect to other methods?

Answer: One of the major advantages is speed; another is that it does not need a full camera calibration. There are two factors that affect speed – the speed of eliminating false point tracks on the ground and the rapid method of computing the rotation angle of the vehicle. Errors in tracking and determination of which points are on the ground plane are problems with a feature-based approach. Our method can efficiently eliminate false tracks, and tracks corresponding to objects not on the ground. The motion model for points (used in the RANSAC procedure) needs only two points to compute. RANSAC complexity increases exponentially in the number of points needed to instantiate the model (4 points for a full homography). Further, the computation of the rotation angle is very simple and by eliminating points not on the ground plane, we avoid the need for ground segmentation.

Vision-based Navigation, Environmental Representations and Imaging Geometries

José Santos-Victor¹ and Alexandre Bernardino¹

Instituto Superior Técnico, Instituto de Sistemas e Robótica, Lisbon, Portugal,
{jasv,alex}@isr.ist.utl.pt – <http://www.isr.ist.utl.pt/vislab>

Abstract. We discuss the role of spatial representations and visual geometries in vision-based navigation. To a large extent, these choices determine the complexity and robustness of a given navigation strategy. For instance, navigation systems relying on a geometric representation of the environment, use most of the available computational resources for localization rather than for “progressing” towards the final destination. In most cases, however, the localization requirements can be alleviated and different (e.g. topological) representations used. In addition, these representations should be adapted to the robot’s perceptual capabilities.

Another aspect that strongly influences the success/complexity of a navigation system is the geometry of the visual system itself. Biological vision systems display alternative ocular geometries that proved successful in different (and yet demanding and challenging) navigation tasks. The compound eyes of insects or the human foveated retina are clear examples. Similarly, the choice of the particular geometry of the vision system and image sampling scheme, are important design options when building a navigation system.

We provide a number of examples in vision based navigation, where special spatial representations and visual geometries have been taken in consideration, resulting in added simplicity and robustness of the resulting system.

1 Introduction

Most of the research on vision-based navigation has been centered on the problem of building full or partial 3D representations of the environment, which are then used to drive an autonomous robot. Instead of concentrating the available resources to progress towards the goal, the emphasis is often put on the process of building (or using) these 3D maps. This explains why many existing systems require large computational power, but still lack the robustness needed for many real-world applications. In contrast, examples of efficiency can be drawn from biology. Insects, for instance, can solve very large and complex navigation problems in real-time (Wehner, Wehner, 1990), in spite of their limited sensory and computational resources.

One striking observation in biology is the diversity of “ocular” geometries. Many animals eyes point laterally, which may be more suitable for navigation purposes. The majority of insects and arthropods benefit from a wide field of view and their eyes have a space-variant resolution. To some extent, the performance of these animals is related to their specially adapted eye-geometries. Similarly, one possibility to explore the advantages of having large fields of view in robotics is to use *omni-directional cameras*.

Studies of animal navigation suggest that most species utilize a very parsimonious combination of perceptual, action and representational strategies that lead to very efficient solutions when compared to those of today's robots.

Both robustness and an efficient usage of computational and sensory resources can be achieved by using visual information in closed loop to accomplish specific navigation tasks or behaviors (Santos-Victor, Sandini, 1997; Santos-Victor, Sandini, Curotto, Garibaldi, 1995). However, this approach alone cannot deal with global tasks or coordinate systems (e.g. going to a distant goal), because it lacks adequate representations of the environment. Hence, a challenging problem is that of extending these local behaviors, without having to build complex 3D representations of the environment.

At this point, it is worth discussing the nature of the navigation requirements when covering long distances, as compared to those for short paths. Many animals, for instance, make alternate use of landmark-based navigation and (approximate) route integration methods (Wehner, Wehner, 1990). For example, to walk along a city avenue, position accuracy to within one block is sufficient. However, entering our hall door would require much more precise movements.

This *path distance/accuracy* tradeoff between long-distance/low-precision and short-distance/high-accuracy mission segments plays an important role in finding efficient solutions for robot navigation.

In the following sections we discuss how different imaging geometries and environment representations can be used for improving the navigation capabilities of an autonomous system.

2 Imaging geometries

In this section we discuss two aspects of the imaging geometry. Firstly, we consider the case of omni-directional cameras whose enlarged fields of view can be advantageous for navigation. Then, we detail the log-polar mapping which is a space-variant image sampling scheme, similar to those found in natural seeing systems. Finally, a combination of both camera and image sensor design is introduced.

2.1 Omni-directional Vision

Omni-directional cameras provide a 360° view of the robot's environment and have been applied to autonomous navigation, video conferencing and surveillance (Baker, Nayar, 1998; Yagi, Nishizawa, Yachida, 1995) among others. Omni-directional images are usually obtained with a combination of cameras and convex mirrors. Mirror shapes can be conic, spherical, parabolic or hyperbolic (Baker, Nayar, 1998).

Visual landmarks are easier to find with omni-directional images, since they remain in the field of view much longer, than with a conventional camera. The imaging geometry has various properties that can be exploited for navigation or recognition. For example, vertical lines in the environment are viewed as radial image lines (see Fig.3).

Our omnidirectional system (Gaspar, Winters, Santos-Victor, 2000) combines a camera and a spherical mirror, mounted on top of a mobile platform, as shown in Fig. 1.



Fig. 1. Left: omni-directional camera. Center: camera mounted on the mobile robot. Right: camera (spherical mirror) projection geometry - symmetry about the z -axis simplifies the geometry.

The geometry of image formation is obtained by equalizing the incidence and reflection angles on the mirror surface. The resulting mapping relates the coordinates of a 3D point, P , to the coordinates of its projection on the mirror surface, P_m , and finally, to its image projection p , as in Fig. 1.

2.2 Space variant (log-polar) sampling

Foveated active visual systems are widely present in animal life. A representation of the environment with high-resolution and a wide field of view is provided through the existence of the space-variant ocular geometry and the ability to move the eyes.

The most common space-variant image representation is the log-polar mapping, introduced in (Schwartz, 1977), due to its similarity to the retinal resolution and organization on the visual cortex of primates. The log-polar transformation is a conformal mapping from points on the *cartesian* plane $x = (x, y)$ to points in the *cortical* plane $z = (\xi, \eta)$ (Schwartz, 1977), as shown in Fig. 2. The log-polar mapping is described by :

$$[\xi, \eta]^t = \left[\log(\sqrt{x^2 + y^2}), \arctan \frac{y}{x} \right]^t \quad [x, y]^t = [e^\xi \cos \eta, e^\xi \sin \eta]^t$$

The application of the log-polar mapping to artificial vision was first motivated by its perceptually based data compression capabilities. When compared to cartesian images, log-polar images allow faster sampling rates without reducing the size of the field of view and the resolution on the central part of the retina (fovea). In addition to rotation and scale invariance (Weiman, Chaikin, 1979), the log-polar geometry provides additional algorithmic benefits: easy computation of time-to-contact (Capurro, Panerai, Sandini, 1997; Santos-Victor, Sandini, 1997), increased stereo resolution on verging systems and good disparity selectivity for vergence control (Bernardino, Santos-Victor, 99; Bernardino, Santos-Victor, Sandini, 2002).

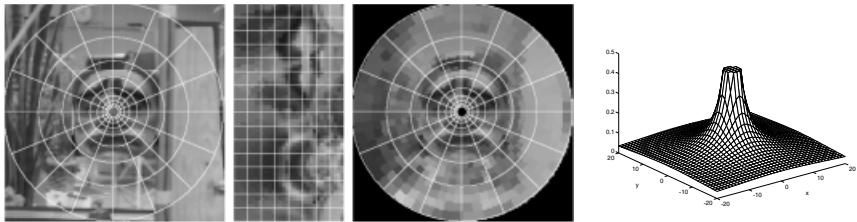


Fig. 2. The log–polar transformation maps points in the cartesian (far left) to the cortical planes (middle left). The effective image resolution becomes coarser in the periphery, as shown in the retinal plane (middle right). The log-polar mapping implements a focus of attention in the center of the field of view, equivalent to a weighting window in cartesian coordinates (far right).

2.3 Omnidirectional vision and Space variant sampling

Both omnidirectional cameras and the log-polar sensor have a rotational symmetry, which suggests the combination of both. As a result, rather than getting the usual omni-directional images, a so-called panorama can be directly obtained by reading out the image pixels. As an additional benefit, the angular resolution is constant, as the log-polar geometry is based upon circular rings with a constant number of pixels (see Fig. 3). The joint mirror profile and sensor layout design was addressed in the

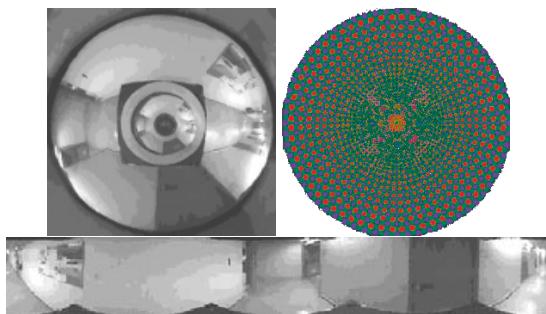


Fig. 3. Combination of omnidirectional images and a log-polar sensor (top) yields directly a constant resolution panorama (bottom).

EU project Omniviews.

3 Environmental representations

In this section we discuss various environmental representations for navigation. We will focus on *alternatives* to the traditional geometric maps. First, we discuss the

use of topological maps. Then, we shall see how to use various forms of image-based (local) representations. Finally, we mention visual servoing as an implicit local representation of the environment.

3.1 Topological Maps

Topological Maps (Brooks, 1985; Gaspar et al., 2000; Kuipers, 1978) can be used to travel long distances in the environment, without demanding accurate control of the robot position along a path. The environment is represented by a graph. *Nodes* correspond to recognizable *landmarks*, where specific actions may be elicited, such as entering a door or turning left. *Links* are associated with regions where some environmental structure can be used to control the robot (see Section 3.3).

Landmarks are directly represented by *images* and a map is thus a collection of inter-connected images (Fig. 4). Precise metric information is not necessarily required to go from one particular locale to another. For example, to get from the city center, *Rossio*, to *Saldanha*, we may *go forward* until we reach the statue in *Rotunda*, *turn right* in the direction of *Picoas* and carry on until we finally reach *Saldanha* Square. The navigation problem is decomposed into a succession of sub-

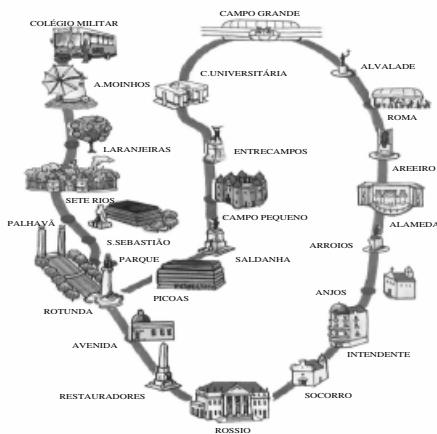


Fig. 4. A topological map of touristic landmarks in Lisbon, Portugal.

goals, identified by recognizable landmarks. The required navigation skills consist of following roads, making turns and recognizing landmarks.

3.2 Panoramas, Bird's eye views and mosaics

Images acquired with an omni-directional camera are distorted, when compared to those of a perspective camera. For instance, a corridor appears as an image band of variable width. However, the image formation model can be used to correct some

distortions, yielding Panoramic images or Bird's Eye Views, which may serve as local image-based environment representations that facilitate tracking or feature extraction.

Scan lines of panoramic images contain the projections of all visible points at constant angles of elevation. Hence, the unwarping consists of mapping concentric circles to lines (Chahl, Srinivasan, 1997). The horizon line is actually transformed to a scan line and vertical 3D lines are mapped as vertical image lines.

Bird's eye views are obtained by radial correction around the image center¹, corresponding to a scaled orthographic projection of the ground plane. For example, corridors appear as image bands of constant width, simplifying the navigation system. Image panoramas and bird's eye views are illustrated in Fig. 5.

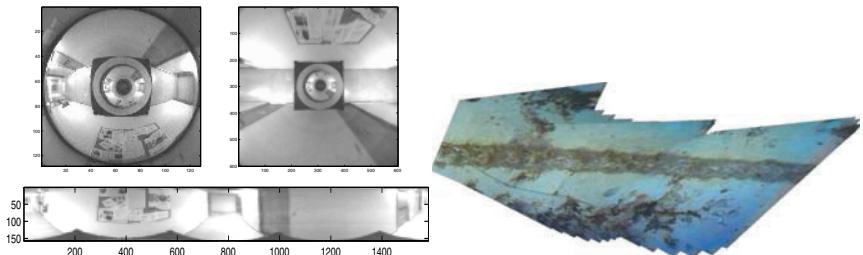


Fig. 5. Left: omni-directional image, the corresponding bird's eye view and the panoramic image. Right: Video mosaic of the sea bottom.

When the camera motion undergoes pure rotation or when the observed scene is approximately planar Video Mosaics constitute interesting representations. Video mosaics can be built by accurately registering images acquired during the camera motion. They offer high resolution and large fields of view, and can serve as visual maps for navigation (Gracias, Santos-Victor, 2000). Figure 5 shows a video mosaic of the sea-bottom².

3.3 Local structure (servoing)

Visual servoing can also be interpreted as yet another form of visual representation. The goal of (image-based) visual servoing consists in reaching desired configurations of image features. As such configurations, (uniquely ?) constrain the camera pose with respect to the work space, they can be considered as an implicit camera-environment representation, rather than describing their (world) coordinates explicitly.

¹ Hicks (Hicks, Bajcsy, 1999) obtained ground plane unwarped images directly, with the use of a custom-shaped mirror.

² Work developed in the EU ESPRIT-LTR Project 30185 NARVAL

4 Examples of Navigation and Vision based Control

In this section we give various examples of visual based navigation and control. All the different examples explore certain camera/image geometries and specific representations of the environment.

4.1 Topological maps and image eigenspaces

When using a *topological map* to describe the robot's *global* environment, a mission can be specified as: "*go to the third office on the left-hand side of the second corridor*".

The topological map consists of a large set of reference (omni-directional) images acquired at pre-determined positions (landmarks), connected by links in a graph. During operation, the reference image that best matches the current view indicates the robot's *qualitative* position.

Reference images can be interpreted as points in a high-dimensional space, each indicating a possible reference position of the robot. As the number of images required to represent the environment can be very large, we build a lower-dimensional linear subspace approximation using Principal Component Analysis (PCA), (Murase, Nayar, 1995).

Figure 6 shows the first 3 principal components (eigenimages) computed from 50 omni-directional images in one corridor, shown in descending order in accordance with their eigenvalues.

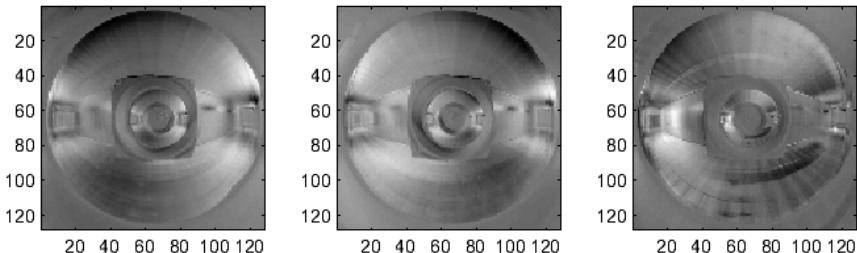


Fig. 6. The first 3 eigenimages obtained with the omni-directional vision system.

A "distance", d_k , between the current view and the set of reference images can be computed in real-time using the projections in the eigenspace. The position of the robot is that associated with the reference image, I_k having the lowest distance, d_k .

Omni-directional images help dealing with relatively *dynamic* environments, where people partially occlude the robot's view. Even when a person is very close to the robot, the occlusion is not sufficiently large so as to cause the robot to misinterpret its topological position.

We have built a topological map from omni-directional images, acquired every 50 cm, along corridors. Reference positions were ordered according to the direction of motion, thus maintaining a causality constraint.

We acquired a set of prior images, P , and ran the robot in the corridor to acquire a different set of run-time images, R . Figure 7 shows the distance d_k , between the prior and run-time images.

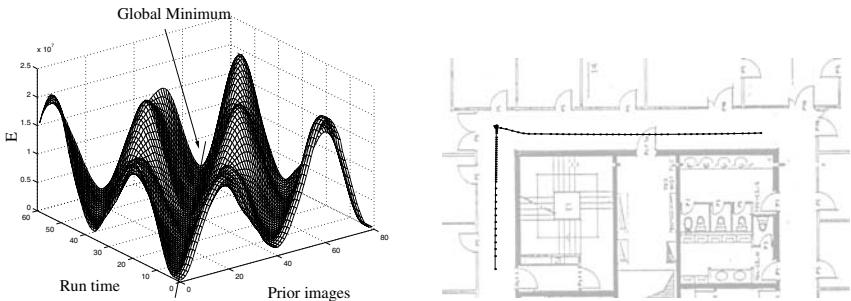


Fig. 7. Left: A 3D plot of the error (d_k) between images acquired at run time, R versus those acquired a priori, P. This plot represents the traversal of a single corridor. The global minimum is the estimate of the robot's topological position. Right: One of the paths traveled by the robot

The error surface presents a *global* minimum, corresponding to the correct estimate of the robot's topological position, and degrades in a piecewise smooth manner. Spurious local minima are due to distant areas of the corridor that may look similar to the robot's current position and can be avoided by restricting the search space to a neighborhood of the current position estimate. Figure 7 shows results obtained when driving the robot along a corridor. The distance traveled was approximately 21 meters. Odometry was used to display the path graphically.

4.2 Servoing on local structure

To navigate along the topological graph, we have to define a suitable vision-based behavior for corridor following (*links* in the map). In different environments, knowledge about the scene geometry can be used to define other behaviors. In this section we provide examples on how to explore local image structure for servoing the robot.

Centering Behavior: The first visually guided behavior is the *centering reflex*, described in (Srinivasan, Lehrer, Kirchner, Zhang, 1991) to explain the behavior of honeybees flying within two parallel “walls”. The qualitative visual measure used is the difference between the image velocities computed over a lateral portion of the left and right visual fields, (Santos-Victor et al., 1995). The ocular geometry (*Divergent Stereo*) was an early attempt to use wide fields of view images for navigation (Santos-Victor et al., 1995).

The robot control system involves two main loops. The *Navigation loop* governs the robot heading in order to balance the bilateral flow fields, hence maintaining the robot at similar distances from structures on the right or left sides. The *Velocity loop* controls the robot forward speed as a function of the amplitude of the lateral flow fields. The robot accelerates in wide spaces and slows down when the environments becomes narrower.

Additionally, a *sustaining mechanism* is embodied in the control loops to avoid erratic behaviors of the robot, in the absence of (localized) flow information. It allows the use of the robot in room-like environments or when the “walls” are not uniformly textured. Figure 8 shows the robot trajectories (from odometry) superimposed on the experimental setup, for various real-time experiments.

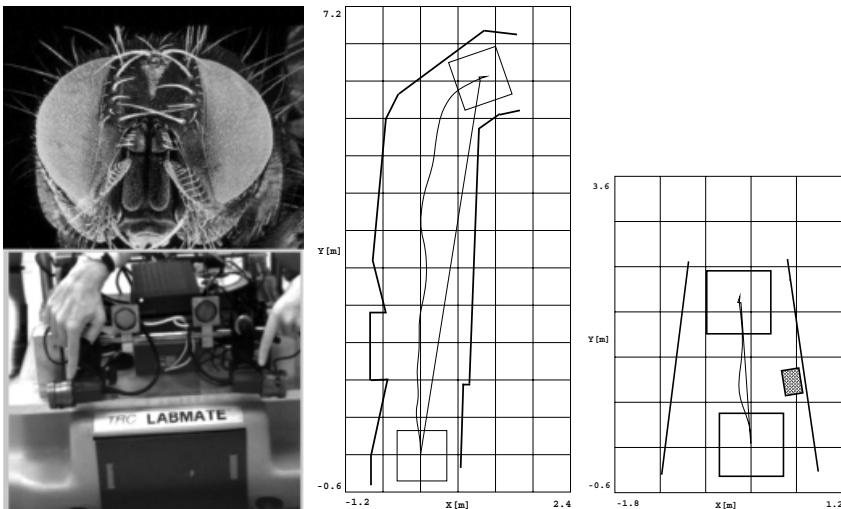


Fig. 8. Left to right: Compound eyes of insects and the divergent stereo configuration; results of the centering reflex obtained with the Divergent Stereo approach in closed loop operation for different scene layouts.

To test the velocity control, we considered the funneled corridor with varying width. As the corridor narrows down, the average flow increases and the velocity control mode forces the robot to slow down, enabling the robot to make a softer, safer maneuver.

Corridor Following with Bird’s eye views: In another example, the parallelism of the corridor guidelines is used to control the robot heading direction. To simplify the servoing task, the visual feedback is provided by *Bird’s eye* views of the floor, computed from omni-directional images.

Tracking the corridor guidelines is done with *bird’s eye* (orthographic) views of the ground plane (see Fig. 9). Projective-planar transformations, computed from

differential odometric data are used to predict the position of points and lines from one image to the next.

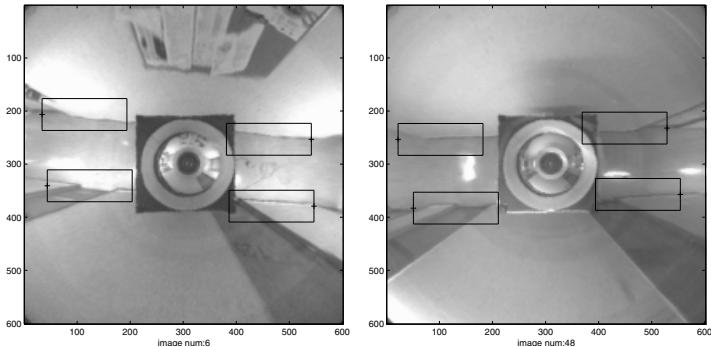


Fig. 9. Bird's eye views during tracking of the corridor guidelines.

The use of bird's eye views of the ground plane simplifies both the extraction of the corridor guidelines (the corridor has a constant width) and the computation of the robot position and orientation errors, with respect to the corridor's central path, which are the inputs of a closed loop controller.

Visual Path Following: Topological navigation can be used to travel between distant places, without accurate localization along the path. For local tasks, we rely on *Visual Path Following* when the robot must follow a reference trajectory accurately for e.g. door traversal, docking and navigation in cluttered environments.

Bird's eye views are used to track environmental features, estimate the robot's position/orientation and drive the robot along a pre-specified trajectory. Again, this geometry simplifies the tracking and localization problems.

The features used are corner points defined by the intersection of edge segments, tracked with a robust fitting procedure. Vertical lines project as radial (or vertical) lines, in the bird's eye view (or panoramic) images. Tracking is simplified by using bird's eye (orthographic) views of the ground plane, thus preserving angular measurements and uniformly scaling distances.

Figure 10 illustrates tracking and localization while traversing a door into a room. The tracked features (shown as black circles) are defined by vertical and ground-plane segments, in bird's eye view images. The robot position and orientation (in the image) are estimated with an Extended Kalman filter and used to control the robot's angular velocity (Gaspar et al., 2000) for trajectory following.

Figure 10 shows tracking and localization while following a reference trajectory, relative to a visual landmark composed of two rectangles. The figure shows the mobile robot at the final position after completion of the task. The processing time is about 0.4 sec/image.

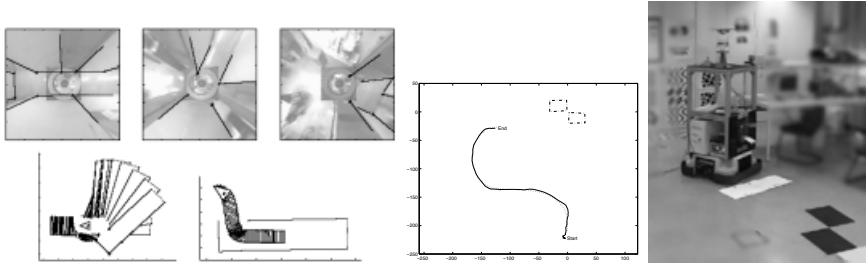


Fig. 10. Far left (clockwise) Feature tracking at three instants (black circles); estimated scene model and self-localization results. Far right: Visual Path Following results: Dash-dotted line shows the landmark. The dotted line is the reference trajectory, specified in image coordinates, and the solid line shows the filtered position estimates; robot at the final position

Tracking with log-polar images: The *Medusa* binocular head is an active tracking system shown in Fig. 11, running at video rate (25 Hz) without any special processing hardware. The mapping from 128x128 cartesian to 32x64 log-polar images takes about 3 ms, which is highly compensated by the reduction achieved (8 times) in the remaining computations.

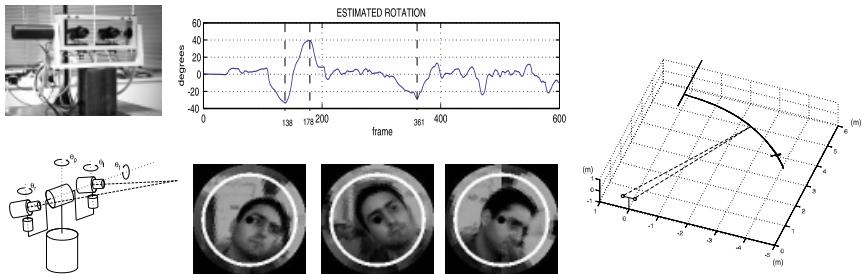


Fig. 11. Left: the Medusa stereo head with four joints (camera vergence, pan and tilt). Middle: estimated rotation in a tracking sequence (top). The frames shown below correspond to the notable points signaled in the plot, associated to local maxima in target motion. Right: Target trajectory measured with stereo

We developed tracking algorithms for objects undergoing general parametric 2D motions, using log-polar images (Bernardino et al., 2002). Figure 11 shows results during a face tracking experiment with a model containing both translation and rotation. Similar results are obtained for more complex motions.

Currently, the binocular system is placed on a static platform and is able not only to track object motion but also to measure its distance and orientation relative to the camera system (see Fig. 11). In future work we intend to place the robotic head on a mobile platform where, by tracking objects, navigation behaviours like following or avoiding objects would be possible. Also fixating static objects in the

environment can be important for several navigation tasks (Fermüller, Aloimonos, 1993), like ego-motion estimation or path planning.

4.3 Topological and local navigation

The following experiment integrates global and local navigation tasks, combining *Topological Navigation* and *Visual Path Following*. Figure 12 shows an image sequence of the robot during the entire run.

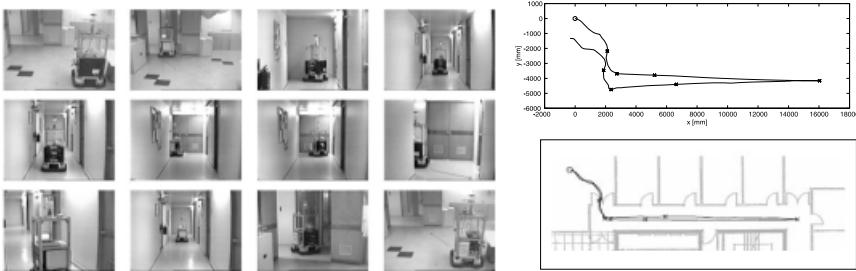


Fig. 12. Left: sequence of images of an experiment combining visual path following for door traversal and topological navigation for corridor following. Right: same type of experiment with showing the trajectory estimated from odometry (top) and the true one (bottom).

The mission starts in the VisLab. Visual Path Following is used to navigate inside the room, traverse the door and drive the robot out into the corridor. Then, control is transferred to the topological navigation module, which drives the robot all the way to the end of the corridor. At this position a new behavior is launched, consisting of the robot executing a 180° turn, after which the topological navigation mode drives the robot back to the lab. entry point. Once the robot is approximately located at the lab. entrance, control is passed to the Visual Path Following module, which locates appropriate visual landmarks and drives the robot through the door. It follows a pre-specified path until the final goal position, well inside the lab., is reached.

Figure 12 shows the robot trajectory during one experiment, and its estimate using odometry. When returning to the laboratory, the uncertainty in odometry is approximately 0.5m. Thus, door traversal would not be possible without the use of visual control.

4.4 Mosaic Servoing

In another example we have used video mosaics as a map for the navigation of an underwater vehicle (Gracias, Santos-Victor, 2000). The vehicle pose is estimated with respect to the mosaic, thus allowing us to control its trajectory towards a desired configuration. Figure 13 shows estimates of the camera pose over time when the underwater vehicle is swimming over an area covered by the mosaic.

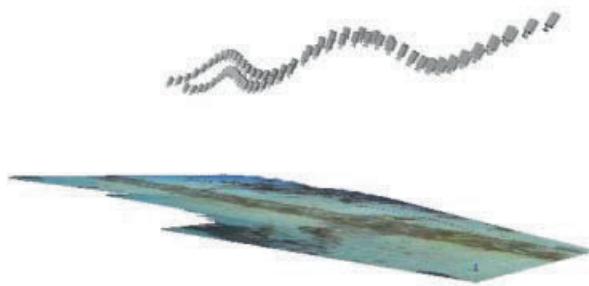


Fig. 13. Using video mosaics for navigation.

5 Conclusions

We discussed the fundamental issues of environmental representations and ocular/image geometries, when addressing the problem of visual based navigation.

In spite of all the recent progress in autonomous (visual) navigation, the performance of today's robots is still far from reaching the efficiency, robustness and flexibility that we can find in biology (e.g insect vision).

In biology, the diversity of ocular/image geometries seem to be extremely well adapted to the main tasks to handle by each species. Additionally, experiments show that spatial representations seem to be remarkably efficient.

Similarly, these aspects are determinant to the success of artificial visual navigation systems. Rather than being defined in advance, spatial representations and the geometry of the vision system should be at the core of the navigation system design. Different navigation methods, eye geometries and environmental representations should be used for different problems, with distinct requirements in terms of processing, accuracy, goals, etc.

In terms of eye geometries, we discussed omni-directional cameras and space variant image sampling. We gave examples of different environmental representations, including topological maps, image-based descriptions (panoramas, bird's eye views, mosaics) and local servoing structures.

Examples of vision based navigation and control were presented to illustrate the importance of the choice of eye geometry and spatial representation. In our opinion, studying eye/image geometries and spatial representations in artificial systems is an essential step, both for understanding biological vision systems and for designing truly flexible and robust autonomous systems.

Acknowledgments The authors would like to acknowledge the contribution of José Gaspar, Niall Winters and Nuno Gracias for the research described in this paper.

References

- Baker S, Nayar SK (1998) A theory of catadioptric image formation. In Proc. ICCV, pp 35–42.
- Bernardino A, Santos-Victor J (99) Binocular visual tracking: Integration of perception and control. *IEEE Trans. Rob. Automation*, 15(6).
- Bernardino A, Santos-Victor J, Sandini G (2002) Foveated active tracking with redundant 2D motion parameters. *Robotics and Autonomous Systems*.
- Brooks R (1985) Visual map making for a mobile robot. In IEEE Conf. Robotics and Automation.
- Capurro C, Panerai F, Sandini G (1997) Dynamic vergence using log-polar images. *IJCV*, 24 (1):79–94.
- Chahl JS, Srinivasan MV (1997) Reflective surfaces for panoramic imaging. *Applied Optics*, 36(31):8275–8285.
- Fermüller C, Aloimonos Y (1993) The role of fixation in visual motion analysis. *IJCV*, 11 (2):165–186.
- Gaspar J, Winters N, Santos-Victor J (2000) Vision-based navigation and environmental representations with an omni-directional camera. *IEEE Trans. on Robotics and Automation*, 16(6):890–898.
- Gracias N, Santos-Victor J (2000) Underwater video mosaics as visual navigation maps. *CVIU*, 79(1):66–91.
- Hicks A, Bajcsy R (1999) Reflective surfaces as computational sensors. In IEEE Workshop on Perception for Mobile Agents, CVPR 99, pp 82–86.
- Kato K, Tsuji S, Ishiguro H (1998) Representing environment through target-guided navigation. In Proc. ICPR, pp 1794–1798.
- Kuipers B (1978) Modeling spatial knowledge. *Cognitive Science*, 2:129–153.
- Lin LJ, Hancock TR, Judd JS (1998) A robust landmark-based system for vehicle location using low-bandwidth vision. *Robotics and Autonomous Systems*, 25:19–32.
- Murase H, Nayar SK (1995) Visual learning and recognition of 3D objects from appearance. *Int. J. Computer Vision*, 14(1):5–24.
- Santos-Victor J, Sandini G (1997) Visual behaviors for docking. *CVIU*, 67(3).
- Santos-Victor J, Sandini G, Curotto F, Garibaldi S (1995) Divergent stereo in autonomous navigation : From bees to robots. *Int. J. Computer Vision*, 14(2):159–177.
- Schwartz E (1977) Spatial mapping in the primate sensory projection : Analytic structure and relevance to perception. *Biol. Cyb.*, 25:181–194.
- Srinivasan M, Lehrer M, Kirchner W, Zhang S (1991) Range perception through apparent image speed in freely flying honeybees. *Visual Neuroscience*, 6:519–535.
- Wehner R, Wehner S (1990) Insect navigation: use of maps or ariadne's thread? *Ethology, Ecology, Evolution*, 2:27–48.
- Weiman C, Chaikin G (1979) Logarithmic spiral grids for image processing and display. *Comp Graphics and Image Proc*, 11:197–226.
- Yagi Y, Nishizawa Y, Yachida M (1995) Map-based navigation for mobile robot with omni-directional image sensor COPIS. *IEEE Trans. Robotics and Automation*, 11(5):634–648.

Mobile Robot Navigation As A Planar Visual Servoing Problem

Peter Corke

CSIRO Manufacturing Science and Technology, PO Box 883, Kenmore, AUSTRALIA 4069

Abstract. This paper demonstrates some interesting connections between the hitherto disparate fields of mobile robot navigation and image-based visual servoing. A planar formulation of the well-known image-based visual servoing method leads to a bearing-only navigation system that requires no explicit localization and directly yields desired velocity. The well known benefits of image-based visual servoing such as robustness apply also to the planar case. Simulation results are presented.

1 Introduction

Visual servoing is an approach to the control of robots in which tasks are defined visually, rather than in terms of previously taught Cartesian coordinates. In visual servo systems, image features obtained from a vision system at a sufficiently high-rate are used to “close a loop” around the pose of a robot end-effector. This is quite different to first-generation robot vision systems that used vision only to determine the initial state of the world, prior to visually open-loop task execution.

The basic train of thought underlying this paper is as follows:

1. Most mobile robot localization and navigation techniques are based on pose estimation (triangulation) from observed landmarks or beacons (Borenstein, Everett, Feng, 1996).
2. A camera is a sensor that returns bearing angle information though it is not generally considered in this way.
3. Navigation is therefore very similar to position-based visual servoing, in as much as it involves explicit pose estimation.
4. Image-based visual servoing has many advantages over position-based and is the basis of most published results in visual servoing.
5. Is there a navigational equivalent of image-based visual servoing?

Such an equivalent would be a localization-free navigation method that directly yields a velocity command for the robot, ie. visual steering. Key questions are whether such an equivalent shares the desirable properties of image-based visual servoing, most notably robustness.

Visual servoing has been previously applied to mobile robot navigation citepRives93, Swain97 but the approaches have used a standard camera and the 3-dimensional visual servoing formulation. In this paper the problem is explicitly reformulated for the planar case, and the analogies with image-based visual servoing and its properties are explored.

The remainder of this paper is structured as follows. Section 2 briefly recaps the two different approaches to visual servoing: position-based and image-based. Then Section 3 provides a planar reformulation of image-based visual servoing for navigation and presents some simulation results. Section 4 looks at research in the image-based visual servoing field that is applicable to the navigation problem. Finally Section 5 presents conclusions and ideas for future work.

2 Visual Servoing Revisited

Broadly speaking, there are two basic approaches to visual servo control: Image-Based Visual Servo (IBVS), and, Position-Based Visual Servo (PBVS). In PBVS systems, features are extracted from an image and used to compute a full or partial 3D reconstruction of the environment based on apriori knowledge (Ganapathy, 1984; Rizzi, Koditschek, 1991; Westmore, Wilson, 1991). An error is then computed in the task space, and is input to the control system. Thus, the central control problem confronted by a PBVS system is the classical robotics problem of tracking a Cartesian trajectory (Feddema, Mitchell, 1989). With PBVS, any errors in calibration of the vision system will lead to errors in the 3D reconstruction, and subsequently to errors during task execution. In addition, since the control law for PBVS is defined in terms of the 3D workspace, there is no mechanism by which the image is directly regulated. Thus it is possible that objects of interest (including features that are being used by the visual servo system) can exit the camera's field of view.

In IBVS an error signal is measured in the image, and is mapped directly to actuator commands (see, e.g., (Feddema, Mitchell, 1989; Sanderson, Weiss, 1980)). IBVS approaches have seen increasing popularity, largely due to the shortcomings of PBVS systems just mentioned. The fundamentals of Image-Based Visual Servo control are as follows. Let $p = (x, y, z)^T$ represent a point, and

$$\dot{r} = (T_x, T_y, T_z, \omega_x, \omega_y, \omega_z)^T$$

represent the corresponding end-effector velocity, composed of a linear velocity $\mathbf{v} = (T_x, T_y, T_z)^T$ and angular velocity $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$. Let $f = (u, v)^T$ be the image-plane coordinates of a point in the image and $\dot{f} = (\dot{u}, \dot{v})^T$ the corresponding velocities. The image Jacobian relationship is given by

$$\dot{f} = J(f, r)\dot{r}, \quad (1)$$

with

$$J = \begin{bmatrix} \frac{\lambda}{z} & 0 & \frac{-u}{z} & \frac{-uv}{\lambda} & \frac{\lambda^2 + u^2}{\lambda} & -v \\ 0 & \frac{\lambda}{z} & \frac{-v}{z} & \frac{-\lambda^2 - u^2}{\lambda} & \frac{uv}{\lambda} & u \end{bmatrix} \quad (2)$$

in which λ is the focal length for the camera. This equation is often referred to as the optical flow equation since it relates feature velocity (flow) to camera motion.

Derivations of this can be found in a number of references including (Aloimonos, Tsakiris, 1991; Haralick, Shapiro, 1993; Hutchinson, Hager, Corke, 1996).

The image Jacobian was first introduced by Weiss et. al (Sandersona, Weiss, Neuman, 1987), who referred to it as the *feature sensitivity matrix*. It is also referred to as the *interaction matrix* (Espiau, Chaumette, Rives, 1992) and the **B** matrix (Papanikolopoulos, Khosla, Kanade, 1993). The most common image Jacobian is based on the motion of points in the image (e.g., (Espiau et al., 1992; Feddema, Mitchell, 1989; Hashimoto, Kimoto, Ebine, Kimura, 1991; Papanikolopoulos et al., 1993; Sharma, Hervé, Cucka, 1992; Skaar, Brockman, Jang, 1990)), but other image features have been used in visual servo schemes, including: the distance between two points in the image plane and the orientation of the line connecting those two points (Feddema, Mitchell, 1989), perceived edge length (Weiss, Sanderson, Neuman, 1987), the area of a projected surface and the relative areas of two projected surfaces (Weiss et al., 1987), the centroid and higher order moments of a projected surface (Andersson, 1988; Lei, Ghosh, 1992; Weiss et al., 1987; Yoshimi, Allen, 1994), the parameters of lines in the image plane (Chaumette, 1990; Espiau et al., 1992) and the parameters of an ellipse in the image plane (Espiau et al., 1992). Of course, each different image feature requires its own specific image Jacobian.

The simplest approach to IBVS is to use (1) to construct the control law

$$u = \Gamma J(f, r)^{-1} \dot{f}^* \quad (3)$$

in which \dot{f}^* is the desired feature motion on the image plane, Γ is a gain matrix, and $u = \dot{r}$ is the control input, an end-effector velocity (this can be converted to joint velocities via the manipulator Jacobian). Of course this approach assumes that the image Jacobian is square and nonsingular, and when this is not the case a generalized inverse, J^+ , is used. Ignoring the dynamics of sensors and actuators the stability of this control law has been proved by (Chaumette, 1990).

IBVS also has its limitations, the most commonly cited being:

1. The need to invert the Jacobian admits difficulties if the matrix is poorly conditioned. Jacobian singularity occurs with degenerate viewing geometry such that object motion induces no change in image features.
2. The image Jacobian (2) contains elements which are a function of the unknown depth, z . Many solutions have been suggested and demonstrated in the literature and which will be discussed further in Section 4 One of the simplest and most common methods is to simply assume a constant value for the depth (Espiau et al., 1992), an approach that is reasonable if the motion of the object remains approximately in a plane parallel to the image plane.
3. The image plane feature trajectories are straight lines, but the Cartesian paths involve unintuitive and often undesirably large camera motions (Corke, Hutchinson, 2000). This is generally the case where large rotations are involved and has led to recent work on hybrid IBVS techniques.

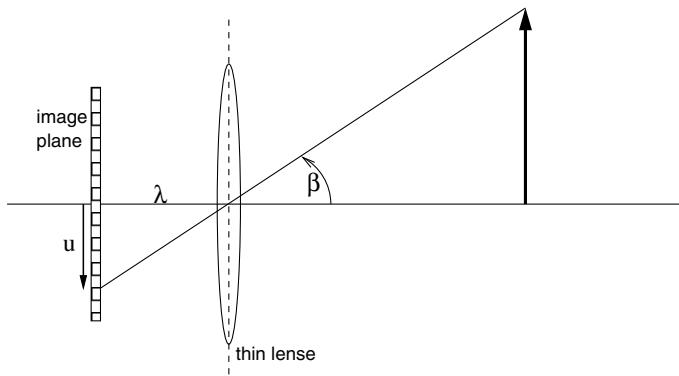


Fig. 1. Simple camera optical model showing bearing angle.

3 A Planar Formulation For Navigation

Mobile robot localization and navigation techniques are typically based on triangulation from beacons or landmarks (Borenstein et al., 1996), using sensors that measure bearing with respect to the robot. A standard camera also measures bearing angle to a point in the scene – it is this observation that connects visual servoing to navigation. In fact the camera geometry, Fig. 1, and simple optics shows that relationship is actually

$$\tan \beta = \frac{u}{\alpha \lambda}$$

where α is the pixel pitch (pixels/m) and λ is the focal length of the lens. Bearing angle is a simple non-linear function of the image plane coordinate u . Other sensors such as laser rangefinders and panospheric cameras provide bearing angle directly. For a camera, the elevation angle is treated similarly. In fact a conventional camera measures two bearing angles simultaneously: azimuth in the horizontal direction and elevation in the vertical direction, but for the planar navigation case only azimuth angle is required.

Navigation is typically formulated in terms of bearing angles to a number of beacons or landmarks. Triangulation based methods that reconstruct the robot's pose from three or more bearing measurements are the navigation equivalent of PBVS. As with PBVS, various algorithmic approaches can be used such as analytic formulations, iterative methods and Kalman filters.

This leads naturally to problem of finding the IBVS equivalent for navigation. Consider just a planar “camera” with a 2π field of view. The geometry of the situation is shown in Fig. 2. The “image feature” is the bearing angle β_i with respect to

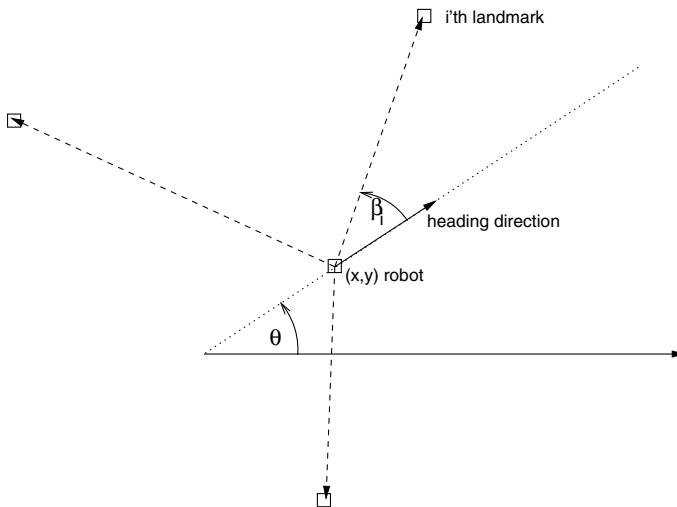


Fig. 2. Notation for planar navigation

the robot's heading direction. The velocity relationship for the i 'th feature is now

$$\dot{\beta}_i = \left[\frac{\sin(\theta + \beta_i)}{z_i} \quad \frac{\cos(\theta + \beta_i)}{z_i} \quad -1 \right] \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (4)$$

where the “image Jacobian” is now a 1×3 matrix.¹ Like its counterpart in visual servoing we note that the terms corresponding to translation rates have feature range, z , in the denominator.

For three or more features we can “stack” the equations

$$\dot{\beta} = J\dot{\zeta}$$

where $\zeta = [x \ y \ \theta]$ is the robot's pose. We then solve using the inverse (for 3 landmarks) or the Moore-Penrose pseudo-inverse (for more than 3 landmarks)

$$\dot{\zeta} = J^+ \dot{\beta}$$

which gives an explicit relationship for robot velocity in terms of feature velocity. If we choose the feature velocity to be proportional to feature error

$$\dot{\beta} = \Gamma (\beta^* \ominus \beta) \quad (5)$$

where Γ is a positive-definite gain matrix, β^* is the desired feature vector and β is the current feature vector, and \ominus denotes smallest angular difference on the unit circle in the range $[-\pi, \pi]$. This control law will move the vehicle such that the feature error decreases monotonically. The stability proof is as for the full IBVS case.

¹ In fact Weiss's original term “feature sensitivity matrix” (Sanderson et al., 1987) is more appropriate than “image Jacobian” in this context.

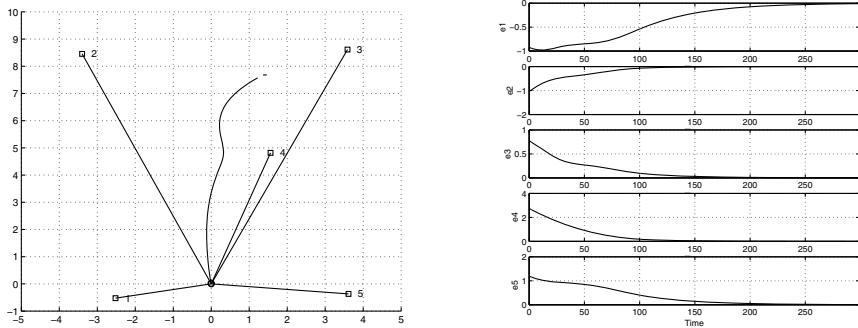


Fig. 3. Navigation results. Left is the XY navigation plane showing beacons and a typical robot path. Right shows the feature error versus time.

Simulation results are shown in Fig. 3. The robot moves from an initial starting point toward the goal at $(0, 0)$ and the feature (bearing angle) errors are shown to be monotonically decreasing. Just as is the case for IBVS the Cartesian trajectory is not a straight line – this is sub-optimal in terms of time and distance and admits the possibility of collision. Recent work (Corke, Hutchinson, 2000) shows how localized potential fields (borrowed from the robot obstacle avoidance literature) can be integrated with IBVS control so as to keep image features within the field of view. Similarly, local collision avoidance behaviours could be integrated with the IBVS navigation method – IBVS providing global control toward the goal with collision avoidance performed by localized potential fields.

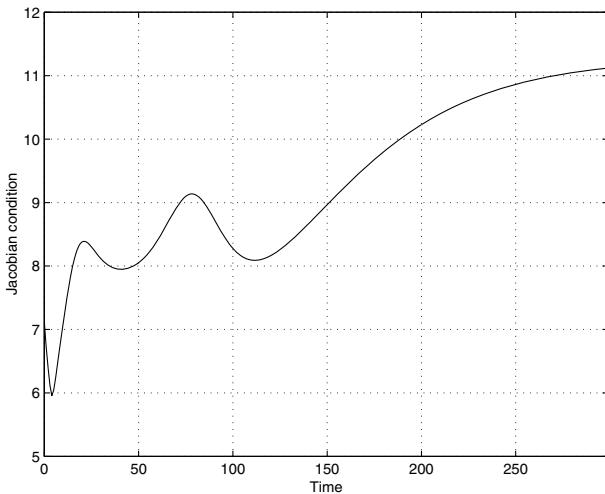


Fig. 4. Jacobian condition number along the trajectory

The condition number of the feature Jacobian is shown in Fig. 4 and is a function of the spatial location of landmarks. It will become singular when two or more landmarks become collinear with the robot.

This simulation assumes that the vehicle is capable of arbitrary motion in the pose space, a feature of many indoor robots. For a conventionally steered vehicle with non-holonomic kinematics it is well known that a fixed gain controller is insufficient, but suitable visual servoing techniques have been demonstrated by others (Das, Fierro, Kumar, Southall, Spletzer, Taylor, 2001; Orosez, Devy, 1997; Rives, Pissard-Gibollet, Kapellos, 1993) and could be employed in this case.

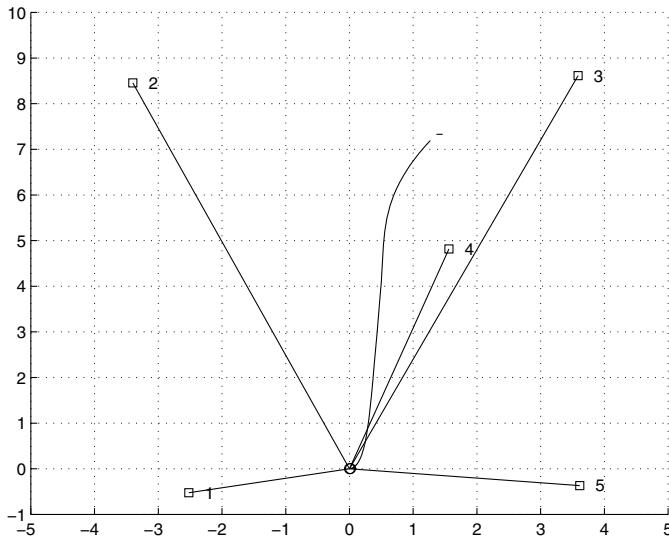


Fig. 5. Path for assumed landmark range of 5m

The simulation of Fig. 3 further assumes that the distance to the landmarks is known which may not be true in practice. This is similar to the problem faced in IBVS and the solution is to assume some constant value of z_i . Over or underestimation of z_i will change the path but within generous bounds does not affect convergence. In Fig. 5 the simulation is repeated but this time it is assumed that $z_i = 5, \forall i$. The path is close to the original one and the navigation approach inherits IBVS's robustness with respect to feature distance. In Fig. 6 the simulation is repeated but with significant Gaussian noise added to the bearing angles at each time step. Once again the system converges on the goal.

Figure 7 shows the velocity vectors for various initial robot locations. Since the method is based on linearization it is a local, not global, controller and for initial conditions outside its basin of attraction it will fail. In general, for starting points within the polygon formed by the landmarks, control performance is quite satisfactory, but the paths are frequently divergent outside this region. An adverse conse-

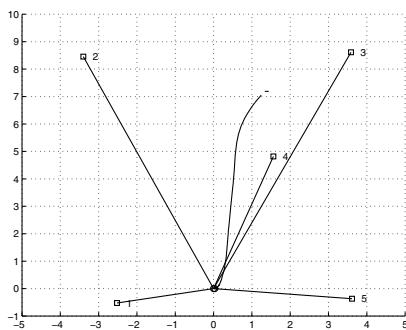


Fig. 6. Navigation results for assumed landmark range and with 5deg variance Gaussian noise on the measured bearing angles

quence of the non-straight paths is that it may be possible for a landmark's bearing angle to change by more than π along the path. This causes a control failure due to the simplistic way in which the angular difference is used to provide the demand in (5).

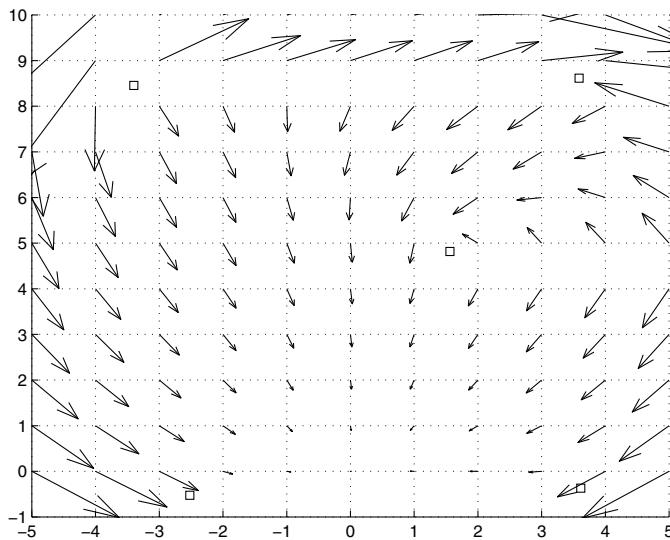


Fig. 7. Attraction field showing initial velocity vectors for various initial locations

4 What Can We Learn From The Visual Servoing Literature?

The IBVS approach was initially proposed by Sanderson and Weiss (Sanderson, Weiss, 1980) and many results have followed over the subsequent two decades that are applicable to the navigation case.

The IBVS method was generalized for a variety of image features besides points, for example straight lines and circles by Chaumette (Chaumette, 1990). This extended feature set may be applicable to the navigation problem using features such as detected wall segments (the analog of image line features).

Feddema (Feddema, Lee, Mitchell, 1991) describes an algorithm to select which subset of the available features give the best conditioned square Jacobian. Others (Hashimoto et al., 1991) have shown that there are advantages in using a larger number of features and using a pseudo-inverse to solve for velocity.

A number of researchers have proposed methods for dealing with the unknown landmark depth, z_i in (2). The classical solution is to use standard computer vision techniques to estimate the value for z (Feddema, Mitchell, 1989); however, this approach amounts to performing a 3D reconstruction of the scene, and brings with it the same drawbacks faced by position-based visual servo schemes. For mobile robot navigation this is analogous to solving for robot location using triangulation. An alternative approach is to estimate the value of z online, as demonstrated by Papanikolopoulos et. al (Papanikolopoulos, Khosla, 1993) using adaptive control techniques. This involves rewriting (4) in identification form where the landmark ranges are the unknown

$$\dot{\beta}_i - \dot{\theta} = [\dot{x} \sin \beta_i - \dot{y} \cos \beta_i] \frac{1}{z_i} \quad (6)$$

and then using observed bearing angle changes and proprioceptive velocity estimates to solve for inverse range by means of a recursive update technique. Persistence of excitation problems can arise and it may be necessary to impose some random motion component to obtain sufficient innovation. Simulation results in Fig. 8 have been derived from the results of Fig. 6 in which navigation is based on an assumed constant landmark distance. Convergence is fast, but become increasingly noisy after $t = 80$ as the vehicle velocity approaches zero.

An alternative to estimating the range of individual landmarks is to estimate the Jacobian itself. Hosoda (Hosoda, Asada, 1994), Jägersand (Jägersand, Fuentes, Nelson, 1996) and Piepmeyer (Piepmeyer, McMurray, Lipkin, 1999) have shown how the Jacobian matrix itself can be estimated online from measurements of robot and image motion. These techniques are based on Broyden, or rank 1, update techniques borrowed from numerical optimization.

Knowledge of landmark bearing angle plus range allows for online creation of approximate Cartesian maps. There are interesting parallels with the recently presented work of (Deans, Hebert, 2000) which finds a link between simultaneous localization and mapping (SLAM) and structure from motion (bundle adjustment) methods used in computer vision. The technique described in this paper is not SLAM but perhaps simultaneous navigation and mapping. There are also interesting

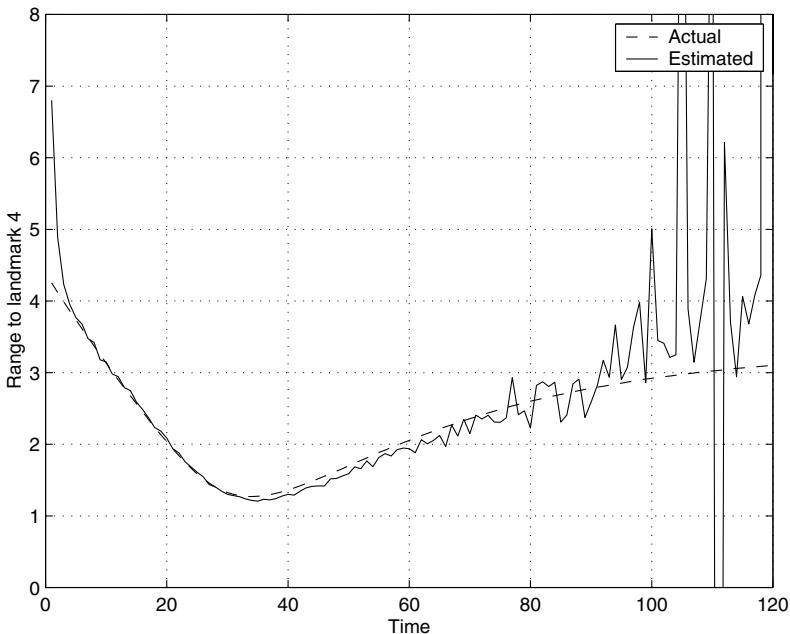


Fig. 8. Estimation of distance to landmark

parallels with the topic of “visual homing” which describes how insects use current and desired views of landmarks to position themselves within the world (Lambrinos, Moller, Labhart, Pfeifer, Wehner, 2000).

5 Conclusions

The establishment of a connection between fields is valuable in that it allows techniques from each of the fields to benefit the other. This paper has shown a connection between visual servo techniques and mobile robot navigation. Classical landmark localization methods combined with trajectory planning are shown to be the planar analog of position-based visual servoing. Adapting the fundamental ideas from image-based visual servoing allows the landmark bearing information to be used directly to form a velocity demand to the robot. Such a navigation method does not involve localization, the robot simply performs a gradient descent toward its goal in the bearing error space.

The IBVS control method is now two decades old and its characteristics are well understood. Its greatest virtue is robustness to errors in feature measurement and feature range and Section 3 indicates that the planar form has inherited at least some of these virtues. Further work is required to investigate this technique experimentally and to extend it to long motions, perhaps using some sort of visual task switching as discussed by (Oropeza, Devy, 1997).

Acknowledgements The author is grateful to Seth Hutchinson and Henrik Christensen for constructive comments on an early draft of this paper.

References

- Aloimonos J, Tsakiris DP (1991) On the mathematics of visual tracking. *Image and Vision Computing*, 9:235–251.
- Andersson RL (1988) A robot ping-pong player. In *Experiment in Real-Time Intelligent Control*. MIT Press, Cambridge, MA.
- Borenstein J, Everett H, Feng L (1996) Navigating mobile robots : systems and techniques. A K Peters, Wellesley, Mass.
- Chaumette F (1990) La relation vision-commande: théorie et application à des tâches robotiques. PhD thesis, L'Univesité de Rennes I.
- Corke PI, Hutchinson S (2000) A new partitioned approach to image-based visual servo control. In Proc. 39th CDC, pp 2521–2526, Sydney.
- Das A, Fierro R, Kumar V, Southall J, Spletzer J, Taylor C (2001) Real-time vision based control of a nonholonomic mobile robot. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea.
- Deans M, Hebert M (2000) Experimental comparison of techniques for localization and mapping using a bearings-only sensor. Preprints Int. Symp. Experimental Robotics.
- Espiau B, Chaumette F, Rives P (1992) A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8:313–326.
- Feddema J, Mitchell O (1989) Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. Robot. Autom.*, 5:691–700.
- Feddema JT, Lee CSG, Mitchell OR (1991) Weighted selection of image features for resolved rate visual feedback control. *IEEE Trans. Robot. Autom.*, 7:31–47.
- Ganapathy S (1984) Real-time motion tracking using a single camera. Technical report, AT&T Bell Laboratories. Technical Memorandum 11358-841105-21-TM.
- Haralick RM, Shapiro LG (1993) Computer and robot vision. Addison Wesley.
- Hashimoto K, Kimoto T, Ebine T, Kimura H (1991) Manipulator control with image-based visual servo. In Proc. IEEE Int. Conf. Robotics and Automation, pp 2267–2272.
- Hosoda K, Asada M (1994) Versatile visual servoing without knowledge of true jacobian. In Proc. IROS.
- Hutchinson S, Hager G, Corke P (1996) A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12:651–670.
- Jägersand M, Fuentes O, Nelson R (1996) Experimental evaluation of uncalibrated visual servoing for precision manipulation. In Proc. IEEE Int. Conf. Robotics and Automation.
- Lambrinos D, Moller R, Labhart T, Pfiefer R, Wehner R (2000) A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30:39–64.
- Lei M, Ghosh BK (1992) Visually-guided robotic motion tracking. In Proc. Thirtieth Annual Allerton Conference on Communication, Control, and Computing, pp 712–721.
- Oropeza RS, Devy M (1997) Visually-guided navigation of a mobile robot in a structured environment. In Proc. Symp. Intelligent Robotic Systems, Stockholm.
- Papanikopoulos NP, Khosla PK (1993) Adaptive robot visual tracking: Theory and experiments. *IEEE Transactions on Automatic Control*, 38:429–445.
- Papanikopoulos NP, Khosla PK, Kanade T (1993) Visual tracking of a moving target by a camera mounted on a robot: A combination of vision and control. *IEEE Transactions on Robotics and Automation*, 9(1):14–35.

- Piepmeyer J, McMurray G, Lipkin H (1999) A dynamic quasi-newton method for uncalibrated visual servoing. In Proc. IEEE Int. Conf. Robotics and Automation, pp 1595–1600.
- Rives P, Pissard-Gibollet R, Kapellos K (1993) Development of a reactive mobile robot using real-time vision. Preprints of 3rd Int. Symp. Experimental Robotics.
- Rizzi A, Koditschek D (1991) Preliminary experiments in spatial robot juggling. In Proc. 2nd International Symposium on Experimental Robotics, Toulouse, France.
- Sanderson AC, Weiss LE (1980) Image-based visual servo control using relational graph error signals. In Proc. IEEE, pp 1074 – 1077.
- Sanderson AC, Weiss LE, Neuman CP (1987) Dynamic sensor-based control of robots with visual feedback. IEEE Trans. Robot. Autom., RA-3:404–417.
- Sharma R, Hervé JY, Cucka P (1992) Dynamic robot manipulation using visual tracking. In Proc. of the 1992 IEEE International Conference on Robotics and Automation, pp 1844–1849.
- Skaar SB, Brockman WH, Jang WS (1990) Three-dimensional camera space manipulation. International Journal of Robotics Research, 9(4):22–39.
- Weiss LE, Sanderson AC, Neuman CP (1987) Dynamic sensor-based control of robots with visual feedback. IEEE Journal of Robotics and Automation, 3:404–417.
- Westmore DB, Wilson WJ (1991) Direct dynamic control of a robot using an end-point mounted camera and (Kalman) filter position estimation. In Proc. IEEE Int. Conf. Robotics and Automation, pp 2376 – 2384.
- Yoshimi B, Allen PK (1994) Active, uncalibrated visual servoing. In Proc. IEEE International Conference on Robotics and Automation, pp 156–161, San Diego, CA.

Landing Strategies in Honeybees, and Applications to UAVs

M. V. Srinivasan¹, S. W. Zhang¹, J. S. Chahl^{1,2}, and M. A. Garratt¹

¹ Centre for Visual Science, Australian National University, P.O. Box 475, Canberra, A.C.T. 2601, Australia

² Weapons Systems Division, Defence Science and Technology Organisation, P.O. Box 1500, Salisbury, S.A. 5108, Australia

Abstract. Insects, being perhaps more reliant on image motion cues than mammals or higher vertebrates, are proving to be an excellent organism in which to investigate how information on optic flow is exploited to guide locomotion and navigation. This paper describes one example, illustrating how bees perform grazing landings on a flat surface. A smooth landing is achieved by a surprisingly simple and elegant strategy: image velocity is held constant as the surface is approached, thus automatically ensuring that flight speed is close to zero at touchdown. No explicit knowledge of flight speed or height above the ground is necessary. The feasibility of this landing strategy is tested by implementation in a robotic gantry. We also outline our current efforts at exploring the applicability of this and related techniques to the guidance of UAVs.

1 Introduction

Unlike vertebrates, insects have immobile eyes with fixed-focus optics. Therefore, they cannot infer the distances to objects or surfaces from the extent to which the directions of gaze must converge to view the object, or by monitoring the refractive power that is required to bring the image of the object into focus on the retina. Furthermore, compared with human eyes, the eyes of insects are positioned much closer together, and possess inferior spatial acuity (Horridge, 1977). Therefore, the precision with which insects could estimate range through binocular stereopsis would be much poorer and restricted to relatively small distances, even if they possessed the requisite neural apparatus (Srinivasan, 1993). Not surprisingly, then, insects have evolved alternative strategies for dealing with the problems of visually guided flight. Many of these strategies rely on using image motion, generated by the insect's own motion, to infer the distances to obstacles and to control various manoeuvres (Horridge, 1987; Srinivasan, 1993, 1998).

Here we describe how honeybees use image motion cues to perform smooth landings on a flat surface, and explore the feasibility of this and related strategies for the guidance of autonomous aerial vehicles.

2 How Bees Perform Smooth Landings

The seminal work of Gibson (Gibson, 1950) has highlighted the optic-flow cues that can be brought to bear in controlling the landing of an aircraft. Studies of landing

behaviour in flies have revealed that, as a surface is approached, the expansion of the image of the surface provides strong cues that are used to control deceleration and trigger extension of the legs in preparation for contact (Borst, Bahde, 1988; Eckert, Hamdorf, 1980; Goodman, 1960; Wagner, 1982). There is also evidence that the rate of expansion of the image is used to infer the time to contact the surface, even if the insect does not possess explicit information about the speed of its flight or the distance to the surface (Wagner, 1982).

However, when an insect makes a grazing landing on a flat surface, cues derived from image expansion are relatively weak. This is because the dominant pattern of image motion is then a translatory flow in the front-to-back direction. Given that flying insects often make grazing landings on flat surfaces, what are the processes by which such landings are orchestrated?

Recently, this question was investigated by video-filming trajectories, in 3 dimensions, of bees landing on a flat, horizontal surface (Srinivasan et al., 2000).

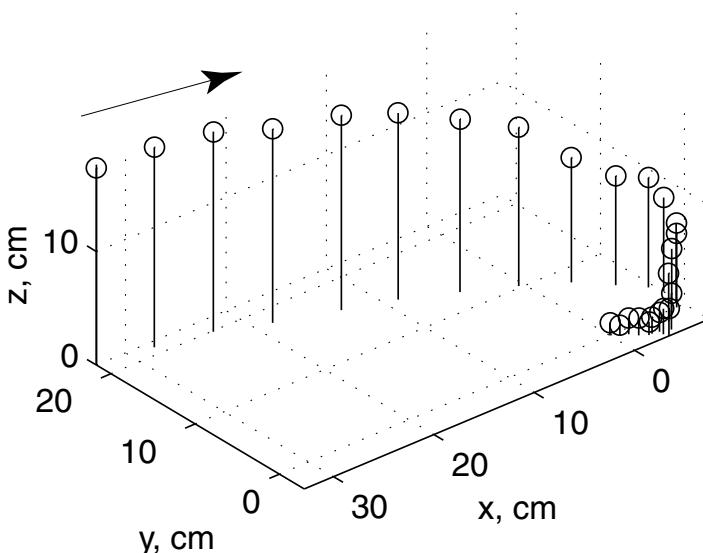


Fig. 1. 3-D reconstruction of a typical landing trajectory, from video film. Vertical lines depict height above surface.

An example of a landing trajectory, reconstructed from the data, is shown in Fig. 1. A number of such landing trajectories were analysed to examine the variation of the instantaneous height above the surface (h), instantaneous horizontal (forward) flight speed (V_f), instantaneous descent speed (V_d) and descent angle ($\arctan[\frac{V_d}{V_f}]$). These variables are illustrated in Fig. 2.

Analysis of the landing trajectories revealed that the descent angles were indeed quite shallow. The average value measured in 26 trajectories was ca. 28 deg (Srinivasan et al., 2000).

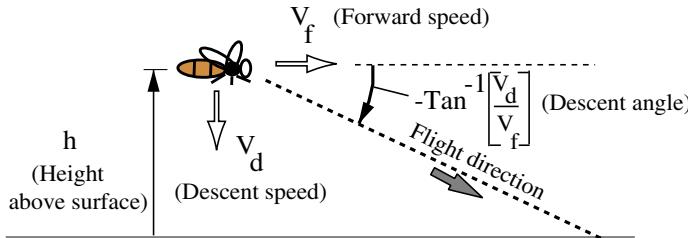


Fig. 2. Illustration of some of the variables analysed to investigate the control of landing. h (cm): height above surface; V_d (cm/sec): horizontal (forward) flight speed; V_f (cm/sec): vertical (descent) speed; $\arctan \left[\frac{V_d}{V_f} \right]$ (deg or rad): descent angle.

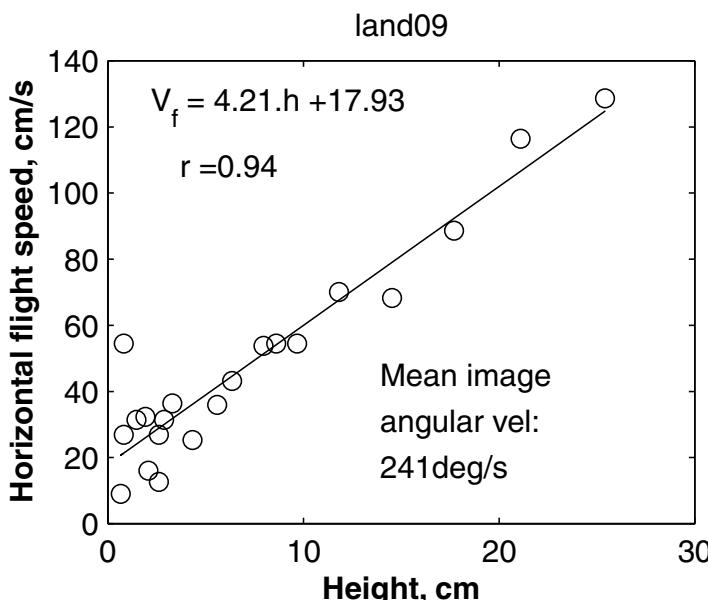


Fig. 3. Variation of horizontal flight speed (V_f) with height (h) above the surface, for a landing trajectory. The straight line is a linear regression through the data, as represented by the equation. r denotes the regression coefficient. Adapted from (Srinivasan et al., 2000).

Fig. 3 shows the variation of flight speed with height above the surface, analysed for a landing trajectory. Analysis of a number of landings in this way reveals one of the most striking and consistent observations with regard to landing bees: Horizontal speed is roughly proportional to height, as indicated by the linear regression on the data. When a bee flies at a horizontal speed of V_f cm/sec at a height of h cm, the angular velocity ω of the image of the surface directly beneath the eye is given by $\omega = \frac{V_f}{h}$ rad/sec. From this relationship it is clear that, if the bee's horizontal flight speed is proportional to her height above the surface (as shown by the data), then the angular velocity of the image of the surface, as seen by the eye, must be constant as the bee approaches it. This angular velocity is given by the slope of the regression line. The angular velocity of the image varies from one trajectory to another, but is maintained at an approximately constant value in any given landing. An analysis of 26 landing trajectories revealed a mean image angular velocity of ca. 500 deg/sec (Srinivasan et al., 2000).

These results reveal two important characteristics. First, bees landing on a horizontal surface tend to approach the surface at a relatively shallow descent angle. Second, landing bees tend to hold the angular velocity of the image of the ground constant as they approach it.

What is the significance of holding the angular velocity of the image of the ground constant during landing? One important consequence is that the horizontal speed of flight is automatically reduced as the height decreases. In fact, by holding the image velocity constant, the horizontal speed is regulated to be proportional to the height above the ground, so that when the bee finally touches down (at zero height), her horizontal speed is zero, thus ensuring a smooth landing. The attractive feature of this simple strategy is that it does not require explicit measurement or knowledge of the speed of flight, or the height above the ground. Thus, stereoscopic methods of measuring the distance of the surface (which many insects probably do not possess) are not required. What is required, however, is that the insect be constantly in motion, because the image motion resulting from the insect's own motion is crucial in controlling the landing.

The above strategy ensures that the bee's horizontal speed is zero at touchdown, but it does not regulate the descent speed. How is the descent speed controlled? Plots of descent speed versus height reveal a linear relationship between these two variables, as well. An example is shown in Fig. 4. This finding implies that landing bees (a) control their forward flight speed to hold the image velocity of the ground constant and (b) control the descent speed to be proportional to the forward speed, so that the descent speed decreases with the forward speed and also becomes zero at touchdown. (In flying *Drosophila*, for example, there is good evidence that lift and thrust co-vary (Goetz, Wandel, 1984)). The ratio of descent speed to forward speed determines the descent angle, which, in effect, is held constant during the landing process. The two rules described above, operating together, ensure a smooth landing.

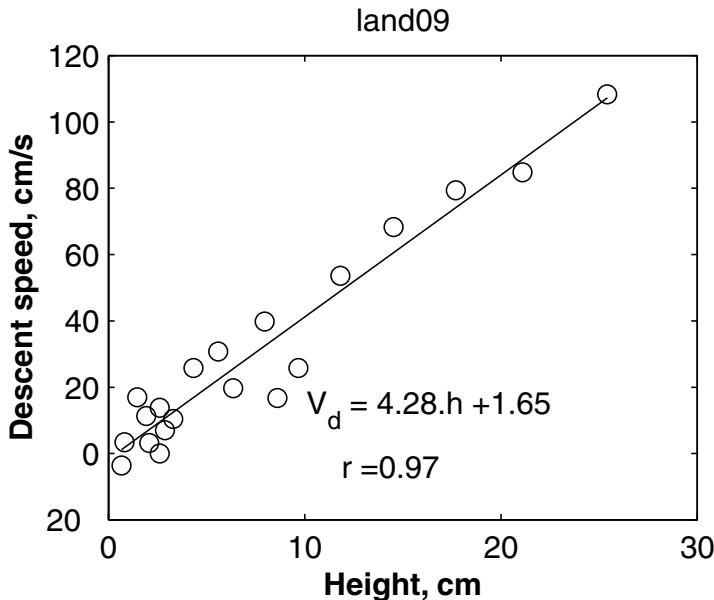


Fig. 4. Variation of descent speed (V_d) with height (h) above the surface, for a landing trajectory. The straight line is a linear regression through the data, as represented by the equation. r denotes the regression coefficient. Adapted from (Srinivasan et al., 2000).

3 Tests on a Robotic Gantry

The feasibility of the landing strategy described above has been tested by implementation in a computer-controlled gantry robot carrying a visual system (Srinivasan et al., 2000). Vision is provided by a video camera mounted on the gantry head, which can be translated in three dimensions (x,y and z). For the purpose of implementing the landing strategy, translatory motion of the camera is restricted to the forward (x) and downward (-z) directions. There is no rotary motion about the z axis.

The system is placed under closed-loop control by using a computer to analyse the motion in the image sequence captured by the camera, and to control the motion of the gantry. A view of the gantry and camera is shown in Fig. 5. The floor, defined to be the landing surface, is covered with a spatially random, black-and-white visual texture. The camera faces downwards and views the floor. The velocity of image motion is measured by using an Image Interpolation algorithm, details of which are given in (Srinivasan, 1994).

Landing is controlled as follows. The system is required to maintain a constant descent angle ($\arctan[\frac{V_d}{V_f}]$) and a constant image angular velocity, ω_{set} , as it descends. In the first time step, the gantry moves the camera head along the direction of descent at an arbitrarily chosen initial speed. The image velocity is measured during this step, using the image interpolation algorithm. Let us denote the measured



Fig. 5. View of robotic gantry, showing camera head and visual texture on the floor. Adapted from (Srinivasan et al., 2000).

image velocity by ω_{meas} . In the next step, the speed of motion of the head is increased or decreased, depending upon whether the measured image velocity is lower or greater than the set image velocity. Specifically, the forward speed of the camera $V_f(i+1)$ during the next step is related to the current speed $V_f(i)$ by

$$V_f(i+1) = V_f(i) \cdot \frac{\omega_{set}}{\omega_{meas}} \quad (1)$$

The speed of descent is also corrected by the same factor, since the forward and descent speeds are proportional to each other and linked by the desired angle of descent. This speed correction ensures that the image velocity during the next step will have the desired value ω_{set} , provided the camera maintains its present altitude. However, since the camera continues to descend during the new step, the forward speed in the following step would have to be reduced further. Thus, both the forward and descent speeds decrease continuously as the camera descends, reaching very low values when the camera is close to the ground.

Landing trajectories generated by the gantry, using this procedure are shown in Fig. 6 for three different descent angles. The image velocities maintained during

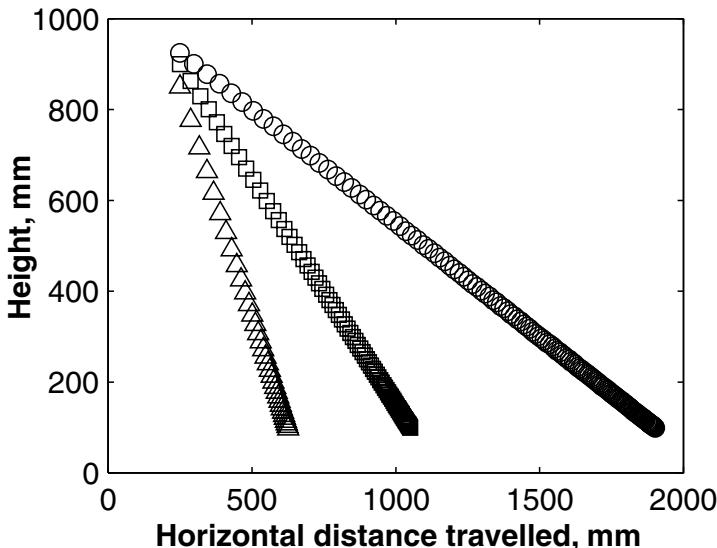


Fig. 6. Landing trajectories generated by the robotic gantry, showing height versus distance travelled for three different descent angles: -26.5 deg (circles), -4 deg (squares) and -63.5 deg (triangles).

these three landings are shown in Fig. 7. It is clear that the image velocity, though somewhat noisy, is held approximately constant. The height of the camera decreases exponentially with time (Fig. 8), as do the forward speed and the speed of descent (Fig. 9). These behaviours are as expected: a quantitative model of the landing strategy predicts and verifies that these variables do indeed vary exponentially with time (Srinivasan et al., 2000).

The results with the robotic gantry suggest that the strategy proposed is a feasible one for landing on flat surfaces, provided the surface carries visual texture that will enable the measurement of image motion. In undulating terrain, the system reduces the forward and descent speeds when the ground rises toward the camera, and increases them when the ground falls away. This is obviously a desirable feature, but has limitations in that the system cannot cope with a situation in which the ground in front rises abruptly to a level above the camera's current height.

A little reflection will reveal that the landing strategy described here can be used by an aerial vehicle to dock with any flat surface, regardless of its orientation: horizontal, vertical or oblique. All that is required is that the vehicle approach the surface in a straight line, and hold the image velocity constant during the approach. This will automatically ensure that the vehicle's speed decreases as the surface is approached, ensuring smooth docking. In the special case in which the surface is approached perpendicularly, the image velocity will be zero in the "straight ahead" direction: the flow field has a pole there. However, the strategy can still be implemented by holding constant the image speed in an annular region surrounding the

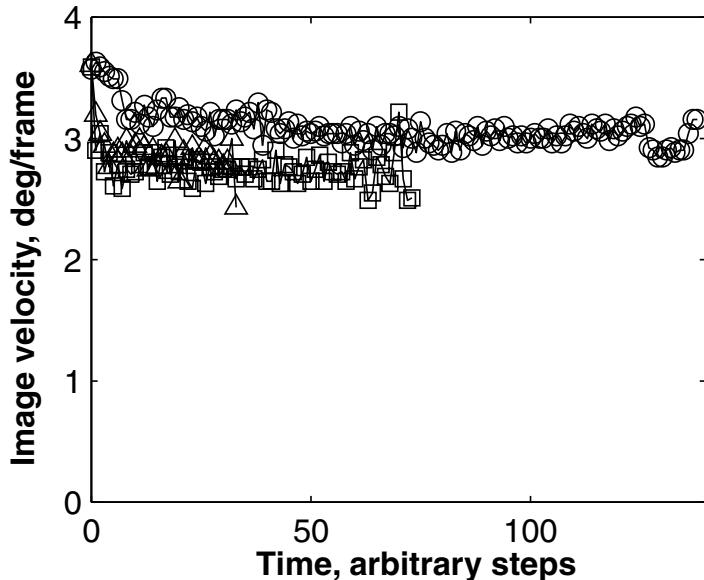


Fig. 7. Landing trajectories generated by the robotic gantry, showing variation of image angular velocity as a function of time. The symbols refer to the three different descent angles, as in Fig. 6.

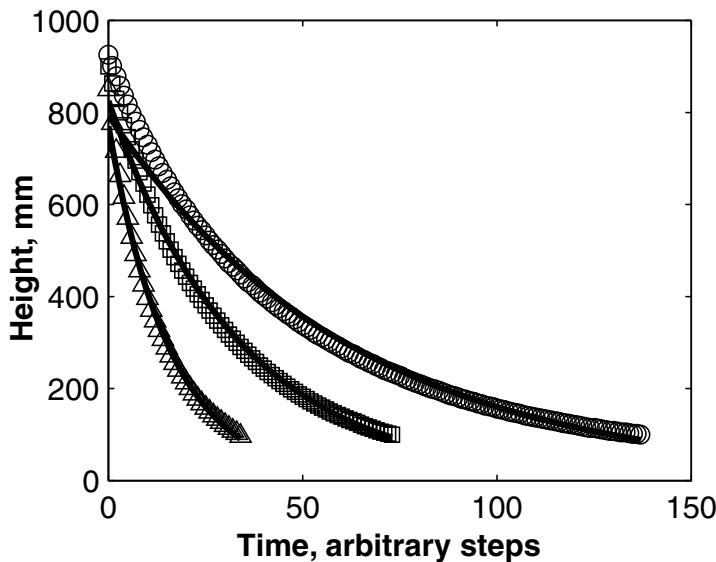


Fig. 8. Landing trajectories generated by the robotic gantry, showing variation of height with time. The symbols refer to the three different descent angles, as in Fig. 6. The line curves depict least-squares fits of exponential functions to the data.

pole, or in a large region centered on the pole. While the present study does not reveal whether bees are actually “aware” of the orientation of the surface in relation to the direction of their approach, it is clear from the above discussion that this information is not necessary for executing the landing process.

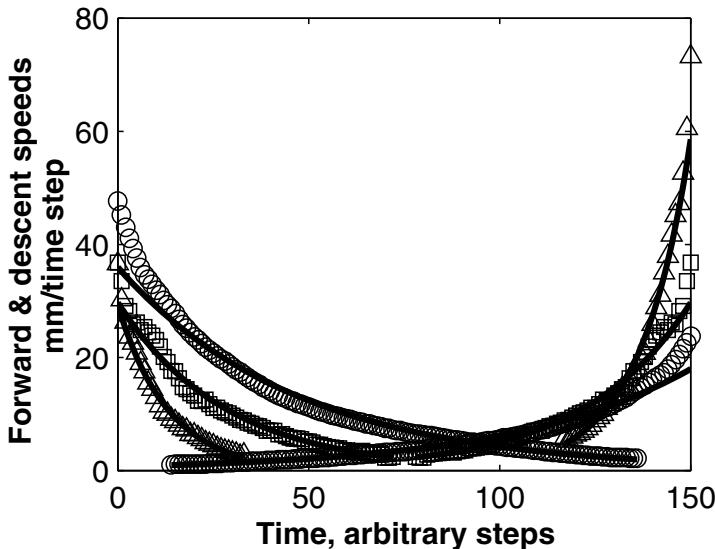


Fig. 9. Landing trajectories generated by the robotic gantry, showing variation of forward speed (left) and descent speed (right) with time. For clarity, descent speeds are plotted with the time axis running backwards. The symbols refer to the three different descent angles, as in Fig. 6. The line curves depict least-squares fits of exponential functions to the data.

4 Implementation in flying vehicles

We are preparing to implement the landing strategy described above, as well as other insect based strategies for visual flight control, in fixed-wing and rotary-wing craft. An overview of one rotary-wing craft is shown in Fig. 10. It is a Hirobo Eagle-X helicopter with a fuselage length of 1.5 m, a weight of 7 kg and a payload capability of 3 kg. The craft carries an imaging system which comprises either a downward-looking video camera, or a panoramic imaging system that uses camera viewing a specially shaped reflecting surface, as shown in Fig. 10e (Chahl, Srinivasan, 1997). Other sensors include a custom-designed, home-built inertial measurement unit comprising three accelerometers and three gyros (Fig. 10d), a three-axis magnetometer, and a GPS system. Video, GPS and other sensor signals are transmitted to a ground-based computer. Here the signals are processed to generate the appropriate flight control commands, which are relayed back to the craft via its wireless control

link. The advantage of this system is that it offers substantial payload savings, since the bulk of the signal analysis and control is done on the ground.

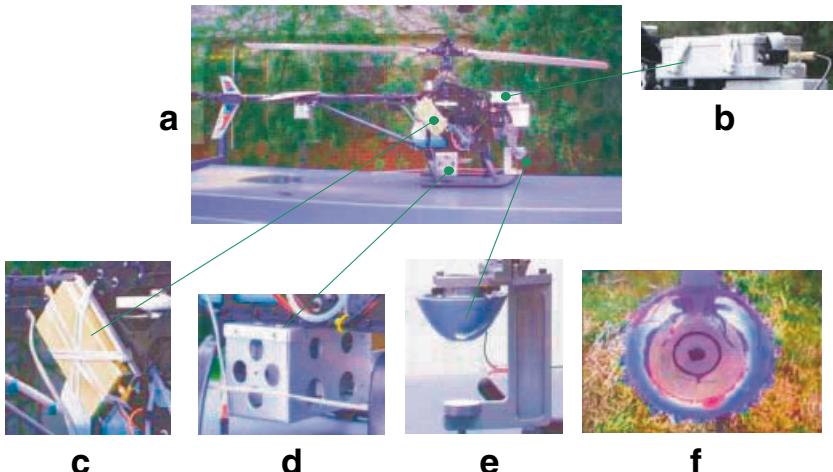


Fig. 10. Overview of helicopter system showing (a) craft, (b) flight computer, (c) video and telemetry transmitter, (d) custom designed inertial sensor housing (e) custom designed and patented panoramic optical imaging system and (f) in-flight panoramic image

The instantaneous height above the ground is computed from knowledge of the craft's ground speed (using GPS information) and from the optic flow generated by the image of the ground. This optic flow is computed at a frame rate of 50 Hz by using an Image Interpolation algorithm (Srinivasan, 1994) to analyse the video image sequence acquired by a downward-looking camera. An example of real-time height estimation using this technique is shown in Fig. 11. The height signal can be used in a feedback loop to regulate height above the ground, and to achieve terrain following. In June 2001, autonomous control of forward flight was first successfully achieved. The craft flew for a distance of approximately 2 km without human intervention, using optic flow measurements to measure and regulate height, and to detect and correct sideways drift. The inertial and magnetometer signals were used to control attitude and to maintain a prescribed heading.

The next phases of this work will involve developing and testing techniques for (a) landing, using the strategy described above and (b) autonomous, visually mediated hover, either with or without the aid of predefined visual landmarks.

5 Conclusions

Analysis of vision in simple natural systems, such as those found in insects, can often point to novel ways of tackling tenacious problems in autonomous navigation. This is probably because insects, with their "stripped down" nervous systems, have

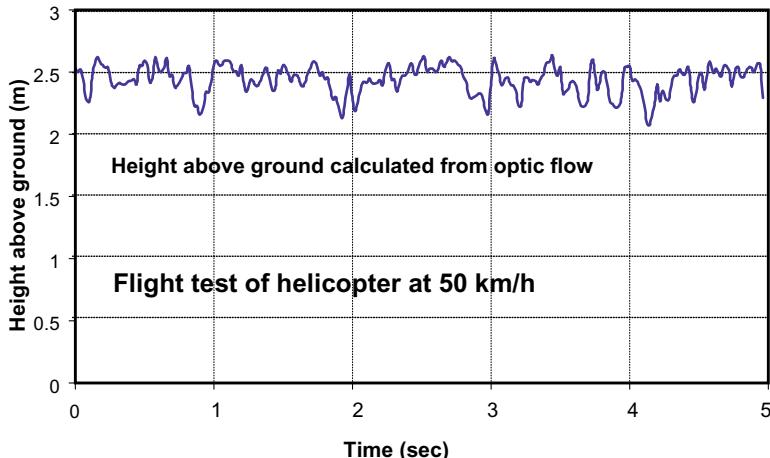


Fig. 11. Height above ground as a function of time, estimated from analysis of ground-induced optic flow. In this experiment the ground speed of the helicopter was 50 km/h.

been forced to evolve ingenious strategies to cope with visual challenges in their environment. This article has outlined a surprisingly simple way in which insects use motion cues to perform smooth landings on flat surfaces. We are now beginning to investigate whether this and other biologically inspired principles can be used to advantage in the design of visually-based control systems for autonomously flying vehicles.

Acknowledgments This research was supported partly by a grant from the Australian Defence Science and Technology Organisation, Salisbury, Grant RG 84/97 from the Human Frontiers in Science Program, and Grant N00014-99-1-0506 from the U.S. Defence Advanced Research Projects Agency and the Office of Naval Research.

References

- Borst A, Bahde S (1988) Visual information processing in the fly's landing system. *J. Comp. Physiol. A*, 163:167–173.
- Chahl JS, Srinivasan MV (1997) Reflective surfaces for panoramic imaging. *Applied Optics*, 36:8275–8285.
- Eckert H, Hamdorf K (1980) Excitatory and inhibitory response components in the landing response of the blowfly, *Calliphora erythrocephala*. *J. Comp. Physiol.*, 138:253–264.
- Gibson JJ (1950) The perception of the visual world. Houghton Mifflin, Boston.
- Goetz KG, Wandel U (1984) Optomotor control of the force of flight in *Drosophila* and *Musca*. II. Covariance of lift and thrust in still air. *Biol. Cybernetics*, 51:135–139.

- Goodman LJ (1960) The landing responses of insects. i. the landing response of the fly, *Lucilia sericata*. *J. Exp. Biol.*, 37:854–878.
- Horridge GA (1977) Insects which turn and look. *Endeavour N.S.*, 1:7–17.
- Horridge GA (1987) The evolution of visual processing and the construction of seeing systems. In *Proc. R. Soc. Lond. B*, vol 230, pp 279–292.
- Srinivasan MV (1993) How insects infer range from visual motion. In *In Visual motion and its role in the stabilization of gaze*, pp 139–156. Elsevier, Amsterdam.
- Srinivasan MV (1994) An image-interpolation technique for the computation of optic flow and egomotion. *Biol. Cybernetics*, 71:401–416.
- Srinivasan MV (1998) Insects as Gibsonian animals. *Ecol. Psychol.*, 10:251–270.
- Srinivasan MV et al. (2000) How honeybees make grazing landings on flat surfaces. *Biol. Cybernetics*, 83:171–183.
- Wagner H (1982) Flow-field variables trigger landing in flies. *Nature*, 297:147–148.

Visual navigation in a plane using the conformal point

Richard Hartley and Chanop Silpa-Anan

System Engineering Department, RSISE, Australian National University

Abstract. A simple construction is given for measuring angles in a plane from a perspective image. The method requires the identification of the horizon line of the plane in the image, as well as a point called the *conformal point*. The angle between two lines is measured by extending the two image lines until they meet the horizon, then connecting back to the conformal point. The angle between the resulting lines is the angle between the original lines in the plane.

The position of the conformal point is computed by calibrating the camera. Alternatively, knowledge of angles in the plane allows the conformal point to be located without explicit calibration. In pure planar motion, where the horizon and the conformal point of the ground plane are preserved, applications such as motion estimation can be applied for robot navigation and measurement of road angles.

1 Motion in a plane

We consider an image of a plane π (the *ground plane*) in which it is possible to determine the horizon line. This may be because the horizon is visible or may be determined by finding horizontal vanishing lines. We assume a standard pinhole camera model, with square pixels.

The task we propose is to measure angles in the plane from their appearance in the image. Various ways of doing this, previously proposed, involve rectification of the image by applying a projective transformation, or else involving Laguerre's formula, or some equivalent algebraic method. None of these methods facilitates any intuitive understanding of the nature of the problem, or how the measured angle depends on assumed values of the focal length, position of the principal point or location of the horizon in the image.

In this paper, however, it is shown that angles may be measured by a very simple geometric construction. This method has the advantage that the true values of the angles seen in the image may be quickly estimated by eye or by computer. The construction depends on the existence of a certain point in the image plane (though sometimes outside of the visible portion of the image) called the *conformal point*. This name is chosen because the imaging process preserves angles at the conformal point; the value of an angle at the conformal point in the image is the same as the actual angle in the plane. Details of the construction are given (without proof) in Fig. 1.

The conformal point is a point in the image located on the line perpendicular to the horizon passing through the principal point (see Fig. 4). It is located at a distance $(f^2 + d^2)^{1/2}$ from the horizon. In this formula, d is the distance of the principal point from the horizon, and f is the focal length of the camera, measured

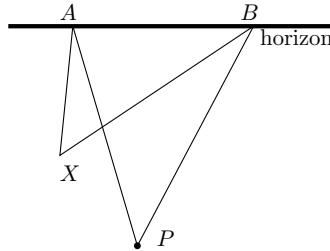


Fig. 1. To measure an angle between two lines in the image, the lines are extended to meet the horizon at points A and B. The true angle between the two lines is equal to the angle $A\widehat{P}B$, where P is the conformal point. The position of the conformal point is explained in the text.

in pixels. Another way of visualising f is that it is the radius of the circle in the image comprising the projection of points located at an angle of 45° from the optical axis – the 45° cone.

Knowledge of the conformal point can be used to find the mapping between the ground plane and the image plane. This mapping can be found up to scale with only knowledge of the conformal point and the horizon. The application of the conformal point to motion estimation is demonstrated at the end of this paper.

2 Proof of the angle construction

In this section, we prove the validity of the method for computing angles in an image of a plane. The proof is contained in the captions to figures 2, 3 and 4.

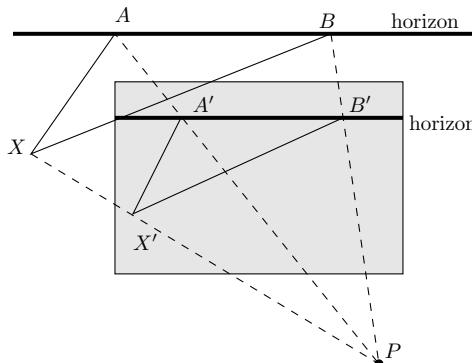


Fig. 2. Two lines meeting at X at an angle θ are extended to points A and B at the line at infinity on the ground plane. These points are projected into the image plane to points X' , A' and B' , forming an angle θ' (not equal to θ) at X' . Point P is the camera centre. Since lines PA and XA meet at infinity, they are parallel; similarly PB and XB are parallel. Consequently $A\widehat{X}B = A\widehat{P}B = A'\widehat{P}B' = \theta$, which is the angle we wish to measure.

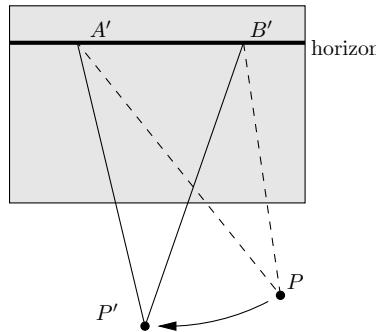


Fig. 3. The triangle $A'B'P$ is rotated around the line $A'B'$ (the horizon line in the image) until it lies in the image plane. Point P is rotated to point P' , and $A'\widehat{P}'B' = A'\widehat{P}B' = \theta$. Point P' is the conformal point. The construction of point P' is independent of the two points A' and B' , since it is obtained simply by rotation of P about the horizon line of the image.

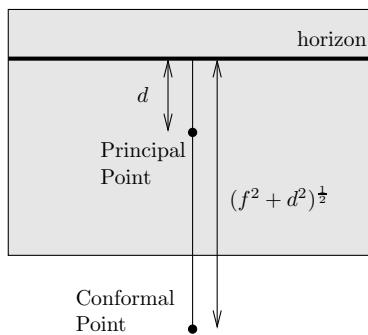


Fig. 4. By Pythagoras' theorem, the distance of P (and hence P') from the horizon line in the image is $(f^2 + d^2)^{1/2}$, where f is the perpendicular distance from P to the image plane (the focal length) and d is the distance from the foot of this perpendicular (the principal point) to the horizon. Hence the conformal point is located a distance $(f^2 + d^2)^{1/2}$ from the horizon along the perpendicular through the principal point.

3 Determining the position of the conformal point

If the principal point, the horizon and the focal length of the camera are known, then the position of the conformal point is easily determined by direct construction. Thus, for a calibrated camera, only the position of the horizon needs to be found in the image. This can be done in many ways. The horizon may be directly visible, or it may be constructed by finding two horizontal vanishing points. A further method for determining the horizon from two or more images will be discussed later. For now, we will assume that the horizon can be found.

In general there are always two conformal points, one on each side of the horizon. It is convenient to choose the one on the same side of the horizon as the prin-

pal point, but the choice is not important. In any case, in the following we consider conformal points on one side of the horizon line only.

Known principal point, plus one known angle. Suppose that the horizon is known, and a right-angle is visible in the image. Let the two lines forming the right angle be extended to the horizon, to points A' and B' . If P' is the conformal point, then the angle $A'P'B'$ must be a right-angle. By elementary geometry P' must lie on a semi-circle with $A'B'$ as diameter. Intersecting this circle with the perpendicular to the horizon through the principal point provides the conformal point.

In the case where the known angle θ is not a right-angle, a similar construction holds. In this case, the line $A'B'$ must subtend an angle θ at the conformal point P' . This constrains P' to lie on an arc of a circle through A' and B' , which is easily constructed directly. Once more, intersecting this circle with the perpendicular to the horizon provides the position of the conformal point. In the case where θ is an acute angle, the circular arc will be greater than a semicircle, and it is possible that there are two intersection points with the perpendicular. This ambiguity may often be removed by observing that the conformal point must be further from the horizon than the principal point.

Two known angles. In the case where the principal point is unknown, the conformal point may be found as the intersection of two circular arcs, one defined by each of the two angles in the manner described above. If one of the angles is acute, then there may be two possible solutions.

Three equal but unknown angles. Given three equal angles in an image, defining vanishing points (A, B) , (A', B') and (A'', B'') at the horizon, one may construct circular arcs through each of the vanishing point pairs corresponding to any angle θ . As θ varies, these arcs grow, until for the correct value of θ they will all intersect in a common point, which must be the conformal point.

4 Homography from conformal point

The location of the conformal point and the horizon in the image implicitly encapsulate the mapping between the ground plane and the image plane. This mapping can be determined using only knowledge of the location of the conformal point and the line representing the horizon in the image.

The homography from the ground plane to the image plane will be specified relative to a specific normalised coordinate system for the ground plane. In turn, this coordinate system will be defined in terms of a normalised coordinate system for the *image*, which we will define first.

The normalised image coordinate system is a Euclidean coordinate system with origin at the conformal point. The y -axis of this coordinate system is the line through the conformal point perpendicular to the horizon, and the x -axis is a line through the conformal point parallel to the horizon. The scale of the coordinate system is such

that the horizon lies at unit distance from the conformal point. In this coordinate system, the horizon line is the line $y = 1$, represented¹ by a homogeneous vector $(0, -1, 1)^T$.

Next we define the normalised *ground* coordinate system. This coordinate system depends on (and moves with) the camera, and is defined in terms of the normalised coordinate system in the image. It is a Euclidean (rectilinear) coordinate system defined by the following three considerations.

1. The origin of the ground coordinate system is the point that projects to the conformal point (that is the origin of the normalised image coordinates).
2. The orientation of the ground coordinates system is defined by the condition that the two axes of the ground coordinate system project to the two axes of the normalised image coordinate system.
3. The scale of the ground coordinate system is set as follows: the point with coordinates $(x_\pi, y_\pi)^T = (1, 0)_\pi^T$ on the ground projects to the point $(x, y)^T = (1, 0)^T$ in the image.

Note that the second condition is possible only because angles are preserved at the conformal point. This condition ensures that a pair of rectangular axes in the image corresponds to a pair of rectangular axes on the ground.

So far, we have defined two correspondences between ground and image points, namely points $(0, 0)^T$ and $(1, 0)^T$ on the ground correspond to points with the same coordinates in the image. In order to define a homography, we need to find two other point correspondences (since four correspondences define a homography). Refer to Fig. 5. In the image coordinate system, the point $(0, 1)^T$ lies on the horizon. Lines joining this point to $(0, 0)^T$ and to $(1, 0)^T$ must therefore correspond to parallel lines on the ground. The first of these ground lines is the axis $x_\pi = 0$, and so the second one is the line $x_\pi = 1$. In particular the line $x + y = 1$ in the image corresponds to the line $x_\pi = 1$ on the ground. Next, since angles (and hence slopes) are preserved at the conformal point the image line $x = y$ must correspond to a line $x_\pi = y_\pi$ on the ground. Intersecting these two line pairs, we see that image point $(0.5, 0.5)^T$ corresponds to point $(1, 1)_\pi^T$ on the ground. By similar arguments, the point $(0, 0.5)^T$ corresponds to the ground point $(0, 1)_\pi^T$.

Now we can construct a fixed homography between these two planes which will map points $(0, 0)_\pi^T, (1, 0)_\pi^T, (1, 1)_\pi^T$, and $(0, 1)_\pi^T$ on the ground plane to points $(0, 0)^T, (1, 0)^T, (0.5, 0.5)^T$, and $(0, 0.5)^T$ on the image plane respectively. Fig. 5 shows the diagram of this fixed mapping.

Therefore, we can write this fixed homography (H_f) that maps points on the ground plane (\mathbf{x}_π) to normalised points on the image plane ($\tilde{\mathbf{x}}$) as

$$H_f = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, [H_f : \mathbf{x}_\pi \mapsto \tilde{\mathbf{x}}].$$

¹ The line with equation $a_0x + a_1y + a_2 = 0$ is represented by a homogeneous vector $\mathbf{a} = (a_0, a_1, a_2)^T$. The condition for a point represented by homogeneous vector $\mathbf{x} = (x, y, 1)^T$ to lie on this line is that $\mathbf{a}^T \mathbf{x} = 0$.

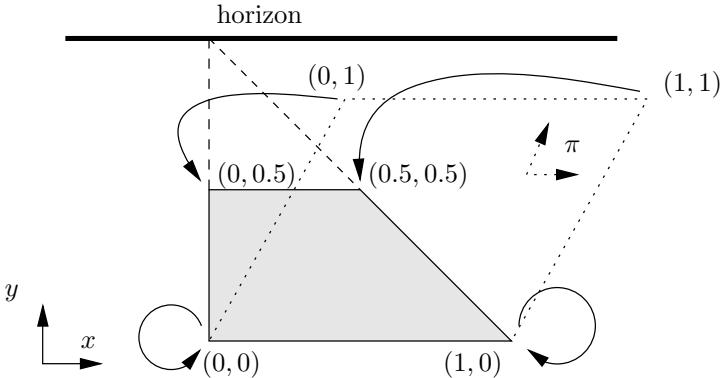


Fig. 5. A fixed homography maps a square on the *ground plane* π (dotted line) to a quadrilateral in the normalised image (solid line). Two parallel sides of the square meet at the horizon on the image. Both lines $y = 0$ on the ground plane and in the normalised image, have the same scale. In other words, this homography maps points $(0, 0)_\pi^T$, $(1, 0)_\pi^T$, $(1, 1)_\pi^T$, and $(0, 1)_\pi^T$ on the ground plane to points $(0, 0)^T$, $(1, 0)^T$, $(0.5, 0.5)^T$, and $(0, 0.5)^T$ in the image.

This is the homography that maps normalised ground coordinates to normalised image coordinates. Generally however, points and lines in an image are measured in pixel coordinates, or some other Euclidean coordinate system. We need to relate normalised image coordinates to the measured image coordinates. To do this, we assume that the conformal point in image coordinates is the point $\mathbf{p}^{cp} = (p_x^{cp}, p_y^{cp})^T$ in the image. Secondly, we suppose that the horizon line in image coordinates is represented by the equation $h_0x + h_1y + h_2 = 0$, or more succinctly as a homogeneous vector $\mathbf{h} = (h_0, h_1, h_2)^T$.

Our task is to find the similarity (scaled Euclidean) transformation H_e that maps point $(0, 0)^T$, to $(p_x^{cp}, p_y^{cp})^T$, and line $(0, -1, 1)^T$ to $(h_0, h_1, h_2)^T$. Without going into the details of the derivation, we observe that

$$H_e = \begin{bmatrix} sh_1 & sh_0 & p_x^{cp} \\ -sh_0 & sh_1 & p_y^{cp} \\ 0 & 0 & 1 \end{bmatrix} \quad \text{where } s = \frac{-\mathbf{h}^T \mathbf{p}^{cp}}{h_0^2 + h_1^2}. \quad (1)$$

Indeed, one easily verifies that $H_e(0, 0, 1)^T = (p_x^{cp}, p_y^{cp}, 1)^T$, so H_e maps the conformal point correctly. Similarly, a general point $(x, 1, 1)^T$ on the horizon line $(0, -1, 1)^T$ maps to point $(sh_1x + sh_0 + p_x^{cp}, -sh_0x + sh_1 + p_y^{cp}, 1)^T$. This point lies on the line $\mathbf{h} = (h_0, h_1, h_2)^T$ if and only if its inner product with \mathbf{bfh} is zero. Expanding, we see that this occurs if and only if $s = -\mathbf{h}^T \mathbf{p}^{cp} / (h_0^2 + h_1^2)$, as required. Thus, H_e maps the horizon line $(0, -1, 1)^T$ in normalised image coordinates to the required line $\mathbf{h} = (h_0, h_1, h_2)$. Consequently, H_e is the desired similarity transformation.

Combining H_e with H_f gives the transformation that maps points in the ground domain to points in the image domain according to $\mathbf{x} = H_{cp} \mathbf{bf} \mathbf{x}_\pi$. The complete

transformation is

$$H_{cp} = H_e H_f = \begin{bmatrix} sh_1 & sh_0 & p_x^{cp} \\ -sh_0 & sh_1 & p_y^{cp} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} sh_1 & sh_0 + p_x^{cp} & p_x^{cp} \\ -sh_0 & sh_1 + p_y^{cp} & p_y^{cp} \\ 0 & 1 & 1 \end{bmatrix}$$

where s is given by (1).

5 Camera moving in a plane

Several papers ((Armstrong, Zisserman, Hartley, 1996; Faugeras, Quan, Sturm, 1998; Montiel, Zisserman, 2001)) have been concerned with *planar motion* of a camera. This means that the camera moves in a plane, and any rotation of the camera is about an axis perpendicular to the plane. For instance, a camera attached rigidly to a roaming robot moving in a plane undergoes this type of motion. A notable feature of this type of motion is that the principal ray of the camera remains inclined at the same angle to the ground, and the horizon line of the ground plane remains fixed in the image. Similarly, the conformal point in the image remains fixed. The discussion in the previous part of this paper relating to angles seen in a single image may be applied to features seen in two or more images related by planar motion.

For instance, imagine a line on the ground plane, visible in an image. Suppose now that the camera undergoes a planar motion to a new position, but the line remains visible in the new image. By superimposing the two images of the line, one on top of the other, an angle is formed. The motion of the robot over the stationary ground plane is equivalent to fixing the robot and moving the ground underneath it. The angle between the two positions of the ground line, may be measured by the conformal point construction. This angle is equal to the angle that the robot has rotated during this motion. Hence, this method gives a very rapid way of tracking the orientation of the robot.

It is not necessary to find real lines in the image of the ground in order to use this method. It is enough to find two feature correspondences between the two images. The virtual line between the two ground features serves as a line for the purposes of this algorithm, and the change of orientation of this line from one image to the next measures the rotation of the robot.

Finding points on the ground plane Points may be tracked from image to image to obtain point-tracks through an image sequence. From this point track, the orientation of the robot throughout the image sequence may be inferred. In doing this, however, it is necessary to distinguish tracked points in the plane from other points. This may be done using a robust statistical technique, such as RANSAC (Fischler, Bolles, 1981). The idea depends on the fact that two tracked points are sufficient to determine the angle of rotation. If two tracked points (reference points) are selected at random, then an angle may be computed. This angle which may be seen as a hypothesized rotation angle, θ . Consider a further point x . This point supports or conflicts with this hypothesis, depending on whether the angle of rotation of the line

from x to each of two basis points is equal to θ (within error bounds). In this way, each pair of chosen reference points receives support from some number of other tracked points. Further randomly chosen reference points are selected, and the pair of points that receives the most support is assumed to be a correctly tracked pair of points on the motion plane (a pair of good points). The process of selecting random points is continued until it is statistically near-certain that at least one pair of good points are selected. Any tracked point that does not support the finally chosen pair of reference points is assumed to be an outlier – either an incorrectly tracked point, or a point not in the plane. A detailed discussion of the RANSAC algorithm is contained in (Hartley, Zisserman, 2000).

Determination of the horizon. The horizon line may be determined from two or more images taken by a robot undergoing planar motion by computing fundamental matrices and epipoles. As shown in (Armstrong et al., 1996), the horizon line passes through all the epipoles, and hence may be calculated by computing the epipoles between all images, and fitting a line to them.

If the robot can be controlled to make a purely translatory motion (no rotation), then there is an even simpler way of computing the horizon line. If a line (real or virtual) is seen in two images, then the intersection of the two superimposed image lines must lie on the horizon. The horizon in the image may be computed by matching two or more lines between the two images, thus obtaining several points on the horizon.

Determining the conformal point. How to compute the conformal point from three equal but unknown angles was discussed in 3. This computation applies naturally to planar motion. Suppose three lines (possibly virtual) are selected in an image, the camera undergoes a planar motion, and the same three lines are identified in a new image. The angle between the original and final positions of each of the lines is equal to the angle of rotation of the robot, and hence is equal for each of the three lines.

6 Motion estimation in planar motion

We can measure the angle between views easily by finding a line correspondence or two point correspondences between two images and measuring the angle between the two lines. This has solved one third of the planar motion estimation problem. The rest is to estimate the translation on the plane (two degrees of freedom). As mentioned previously, a homography H_{cp} that maps the ground plane to the image plane can be constructed from the location of the conformal point and the horizon in the image. This will help us find the motion on the ground plane which may be written in a Euclidean homography form as

$$H_\pi = \begin{bmatrix} R(\theta) & t_x \\ t_y & 1 \\ \boldsymbol{\theta}^T & 1 \end{bmatrix}, [H_\pi : x_\pi \mapsto x'_\pi].$$

For a point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ on the image plane, we can write a homography that maps point \mathbf{x} to point \mathbf{x}' in terms of the homography H_{cp} and H_π as

$$\begin{aligned}\mathbf{x}' &= H\mathbf{x}, \\ \mathbf{x}' &= H_{cp}H_\pi H_{cp}^{-1}\mathbf{x}.\end{aligned}$$

In the above equation, the rotation matrix $R(\theta)$ can be estimated using the angle found by line correspondence and the conformal point. The translation t_x and t_y are linear in this equation, hence, we can use a DLT-like algorithm ((Hartley, Zisserman, 2000)) to solve for t_x and t_y . We may improve our estimation of the three parameters of H_π (θ , t_x , and t_y) by using iterative minimisation method such as bundle-adjustment using the Levenberg–Marquardt algorithm ((Hartley, Zisserman, 2000)). A cost function may be defined in the image plane as a one-sided transfer cost function

$$\sum_{i=1}^n ||\mathbf{x}'_i - H_{cp}H_\pi H_{cp}^{-1}\mathbf{x}_i||,$$

with θ , t_x , and t_y as parameters for minimisation. We can use the values obtained previously from the linear method to initialise the iterative minimisation. We choose to use the cost function in the image domain because of the assumed uniform isotropic uncertainty in measuring the locations of points in the image. Due to the homography mapping H_{cp} , points' locations in the ground domain do not have a uniform isotropic uncertainty. Other types of cost functions can be used as well.

It is important to note here that the motion estimated by this method is the motion of the normalised ground coordinate frame over the ground as the camera moves, and not the actual motion of the camera. Thus, if the camera rotates without moving, then the normalised coordinate frame origin (projection of the conformal point) moves in an arc about the rotation centre. Nevertheless, the actual motion of the camera centre is easily determined, since at all times it lies a fixed distance below the conformal point along the y_π axis.

7 Examples

Included here are some examples of constructing the conformal points and measure angles in the scene.

Known principal point, plus one known angle Figure 6 shows the construction of the conformal point given a known principal point and a known angle. The horizon is estimated using the vanishing points from parallel lines. Using a known right-angle (one corner in the tennis court), the conformal point must lie on the semi-circle joining the two intersections of the right angle and the horizon.

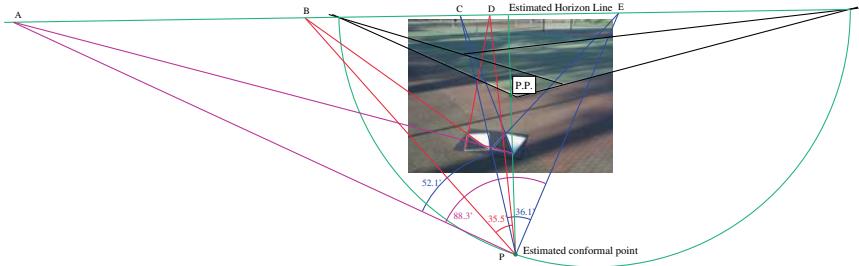


Fig. 6. The conformal point point P is constructed from a known 90° angle (one corner in the tennis court) and a known principal point. The triangle in the scene is a $35^\circ/55^\circ$ right triangle. The angle $A\hat{P}E$ is measured to be 88.3° , 1.7° off from a known 90° . Other measurements are also close to true value e.g. $35.5^\circ/35^\circ$, $36.1^\circ/35^\circ$, and $16.1^\circ/55^\circ$.

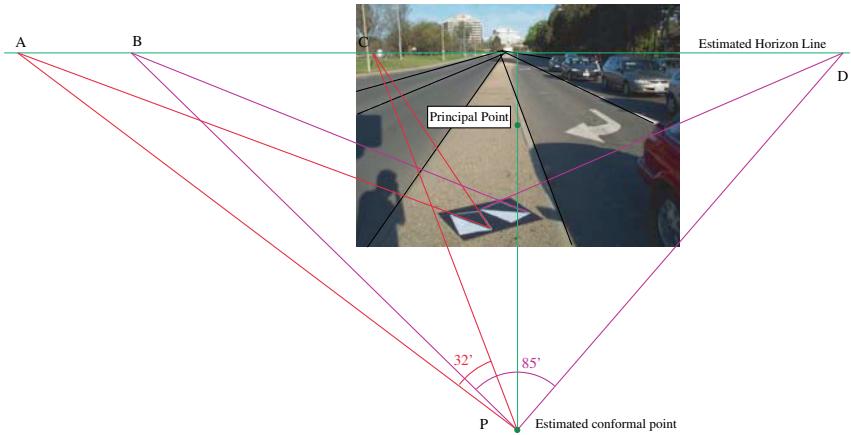


Fig. 7. The conformal point P is constructed from a known principal point and a known focal length. The angle $A\hat{P}C$ is 32° from the measure. The angle $B\hat{P}D$ is 85° from the measure. Both angles have a true value of 35° and 90° respectively. Additionally, the shape of the arrow in the road-marking on the right were measured using this method. The three angles of the triangular arrow-head were measured to be 115.7° , 30.5° and 33.7° , compared with the true values (measured on the road) of 116° , 32° and 32° .

Known principal point, plus known focal length Figure 7 shows the estimation of the conformal point using a known principal point and a known focal length. The horizon is estimated using the vanishing point of lines from the road.

Image sequences. Figure 8 shows the estimation of the conformal point using a known principal point and a known focal length. The horizon is estimated using the vanishing point of lines from the road from three images. Rotation of the camera can be estimated using the conformal point and a line in the image sequences.

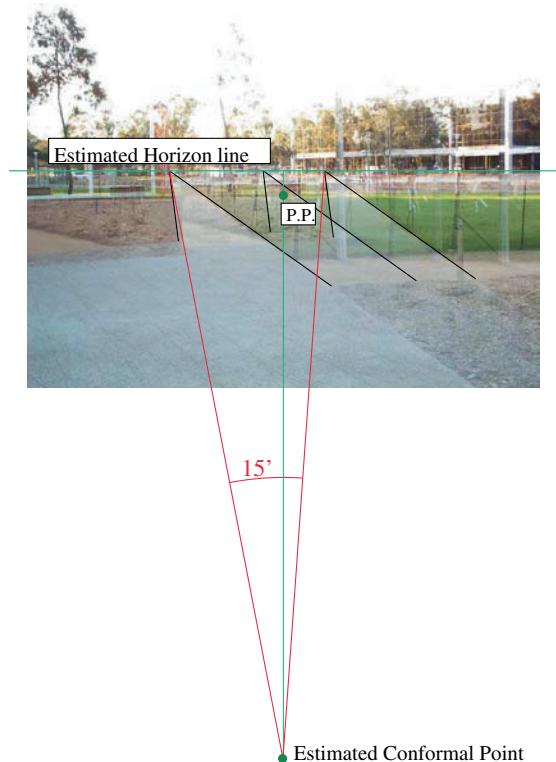


Fig. 8. The conformal point P is constructed from a known principal point and a known focal length. Three images are superimposed on top of each other. The vanishing point of the road from each image is used to estimate the horizon. The camera was rotated about 15° between the first and the last image. Estimation of the camera rotation is found to be 15° .

Automatic point tracking and RANSAC weeding. Points were automatically tracked on an image sequence taken with a robot indoors, moving along a corridor. The image tracked points included many false matches and also tracked points not on the ground plane. Bad point matches were weeded out using the method described above. The results are shown in Fig. 9.

Swivelling Motion. Figure 10 shows the motion estimation of the conformal point. In this set-up the camera is rotated about 15° to the left and the right on a tripod. The heading of the camera and the motion of the conformal point are estimated.

Motion Estimation. Figure 11 shows the motion estimation of the conformal point. In this set up the camera travels forward about a meter then turns 45° to the right and travels forward again. The graphs show a good estimation of both heading and motion. Figure 12 shows the motion estimation of a camera travelling in a loop and coming back to the original spot.

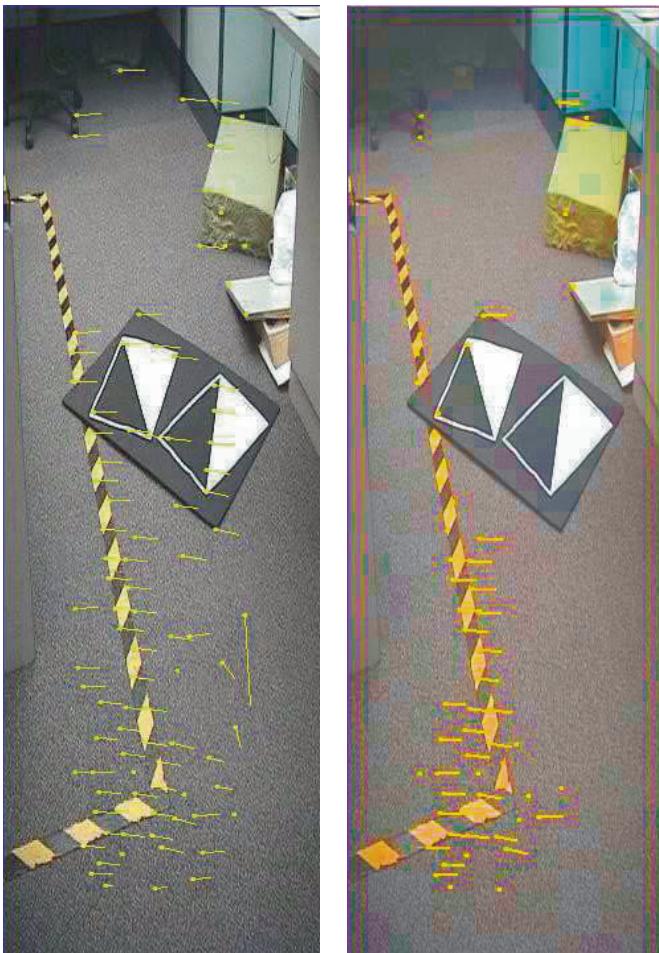


Fig. 9. The image on the left shows a set of points tracked between two frames of an image sequence. Lines attached to points indicate the motion of the point between frames. As may be seen, some false matches were present (indicated by lines directed across the prevailing flow). After RANSAC weeding (right image) all remaining points may be used to obtain an accurate estimate of image rotation.

8 Conclusion

The method described in this paper for estimating angles in a plane is extremely rapid, robust and intuitive. It has a significant advantage that it does not require a complete calibration of the camera. It is sufficient only to determine the conformal point and horizon for the plane being viewed. This avoids the need for exact determination of the principal point and focal length of the camera, which are known to be hard to compute stably. Applications in robot navigation are promising. A fur-

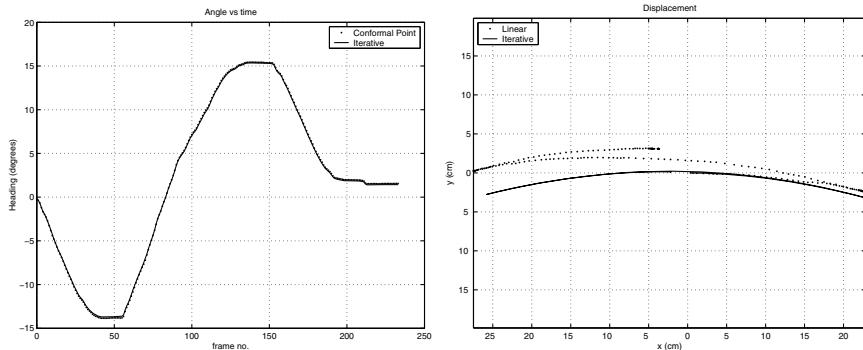


Fig. 10. The camera is rotated on a tripod to the left and right for about 15° each side. For the angle estimation (left figure), both estimations from conformal point and its iterative solutions are similar. The right figure shows the difference in displacement estimation of the conformal point. The iterative solution (solid line) shows a much better estimation than the linear solution (dotted line). Since the camera is stationary the motion of the conformal point is expected to be on the same curve for this pure rotation.

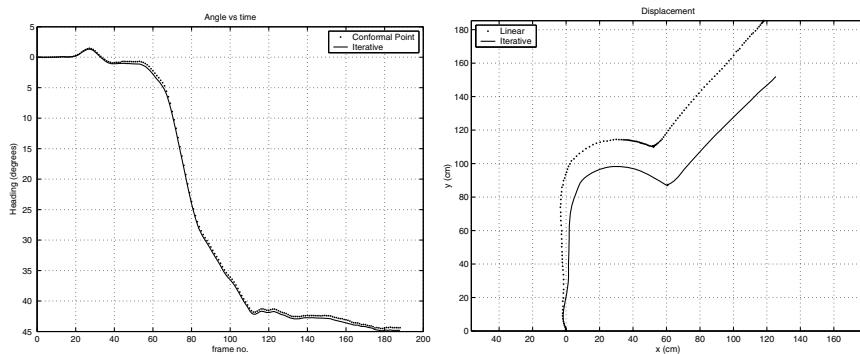


Fig. 11. The camera travels forward for about a meter then turns 45° to the right and travels forward again. The left figure shows the heading estimation and the right figure shows the displacement. In both figures, the dotted line shows the closed-form or linear estimate and the solid line is an iterative estimate. During the 45° turning, the displacement graph shows extra lateral motion to the right because the estimation is the motion of the projection of the conformal point onto the ground, and not the motion of the camera centre itself. The conformal point lies in front of the camera centre and has a lateral motion whenever the camera rotates. The iterative solution gives a smaller estimate which is similar to the case of the swivelling camera.

ther obvious application is to autonomous vehicle navigation, for real-time rapid determination of vehicle orientation and angles of curves in the road.

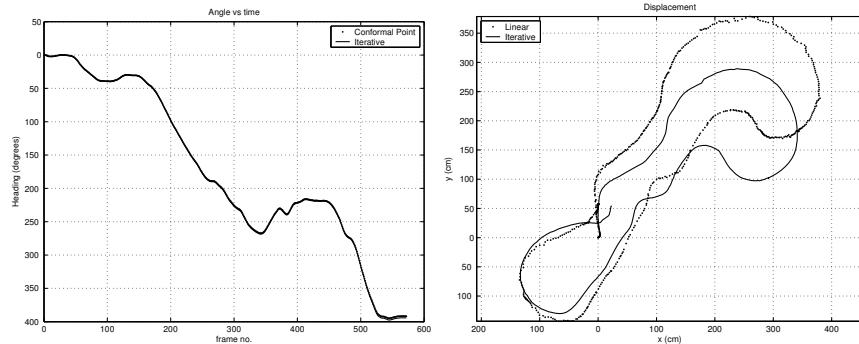


Fig. 12. The camera is set to travel in a loop and come around back to the original location. For this motion, the heading should be approximately 0° or 360° at the end of the motion and should also come back to $(0, 0)$ coordinates. The left figure shows the heading estimation and the right figure shows the displacement. In both figures, the dotted line is the closed-form or linear estimate, and the solid line is an iterative estimate. The heading estimates from both conformal point and iterative solutions are similar. In the displacement estimation, the iterative method again gives a more compact estimate.

References

- Armstrong M, Zisserman A, Hartley R (1996) Self-calibration from image triplets. In Proc. 4th European Conference on Computer Vision, vol LNCS 1064/5, pp 3–16, Cambridge. Springer-Verlag.
- Faugeras OD, Quan L, Sturm P (1998) Self-calibration of 1D projective camera and its application to the self-calibration of a 2D projective camera. In Proc. 5th European Conference on Computer Vision, pp 36–52, Freiburg, Germany.
- Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. Assoc. CComp. Mach., 24(6):381–395.
- Hartley RI, Zisserman A (2000) Multiple view geometry in computer vision. Cambridge University Press.
- Montiel JMM, Zisserman A (2001) Automated architectural acquisition from a camera undergoing planar motion. In Int. Symp. on Virtual and Augmented Architecture (VAA01).

Part 9

Planning and Modeling

Session Summary

Bob Bolles
Chair

SRI International, EK260, 333 Ravenswood Avenue, Menlo Park, California, U.S.A. 94025

This session includes one paper on path planning and three on modeling and control. In the planning paper, Sanchez and Latombe describe an efficient probabilistic roadmap planner for handling complex multi-robot maneuvers. The control papers include an interactive system for drawing and modifying 3-D animations developed at the University of Tokyo by Yamane and Nakamura, plus two theoretical papers describing aspects of contact and compliance. Huang and Kao derive the Conservative Congruence Transformation, which models robot stiffness under external loads, and Featherstone describes a theory for controlling a robot operating in an environment with unknown dynamics.

Sanchez and Latombe present a new route planner that is designed to minimize collision testing, because they made two observations. First, probabilistic roadmap (PRM) planners spend 90% of their time checking for possible collisions. And second, the most expensive path segments to test are the ones that don't have a collision. The new planner is 4 to 40 times as fast as previous PRM planners, depending on the complexity of the environment and the number of degrees of freedom of the "robot." The planner is a Single-query, Bi-directional, Lazy collision checking (SBL) planner. It is single-query in the sense that it explores each path problem from scratch as opposed to generating a persistent representation of free space that is used to answer several queries. It is bi-direction because it explores two trees in parallel, one rooted at the starting configuration and one at the ending configuration. It is lazy in the sense that it postpones collision detection until after a path has been found, and then it checks the segments that are most likely to have occlusions first.

Yamane and Nakamura present a pin-and-drag method for generating and editing animation sequences of three-dimensional figures, such as people. The technique runs in real time on a PC with models that contain 50 degrees of freedom, making it possible for an animator to interactively design motions by pinning some joints, constraining others, and then dragging one joint into a desired position. To achieve this fast and natural interaction, Yamane and Nakamura prioritize the joints in the model and introduce a feedback controller into the differential kinematics. They use a singularity-robust (SR) inverse procedure instead of a pseudoinverse routine in order to avoid problems near singularities. They also incorporate methods to handle spherical joints, such as shoulders, that have limited ranges of motion. And finally, they extend the basic pin-and-drag system so that an animator can edit dynamic sequences.

Huang and Kao present a derivation of the Conservative Congruence Transformation, and then explain how it correctly models compliant robot behavior under external loads. They use Screw Theory to highlight key insights in the derivation. In addition, they briefly discuss three common errors that arise in mathematical models

of stiffness that are caused by incorrect assumptions about finite versus infinitesimal quantities and static versus dynamic quantities.

In the fourth paper, Featherstone presents a new theoretical approach for designing a hybrid motion/force controller for a six degree of freedom robot to operate in an environment having unknown dynamics. The approach produces motion and force subsystems that are instantaneously decoupled. The idea is to handle non-instantaneous effects as slowly-varying disturbances. Experimental validation of this theory is being planned.

A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking

Gildardo Sánchez¹ and Jean-Claude Latombe²

¹ ITESM, Campus Cuernavaca, Cuernavaca, México

² Computer Science Department, Stanford University, Stanford, CA, USA

Abstract. This paper describes a new probabilistic roadmap (PRM) path planner that is: (1) single-query – instead of pre-computing a roadmap covering the entire free space, it uses the two input query configurations as seeds to explore as little space as possible; (2) bi-directional – it explores the robot’s free space by concurrently building a roadmap made of two trees rooted at the query configurations; (3) adaptive – it makes longer steps in opened areas of the free space and shorter steps in cluttered areas; and (4) lazy in checking collision – it delays collision tests along the edges of the roadmap until they are absolutely needed. Experimental results show that this combination of techniques drastically reduces planning times, making it possible to handle difficult problems, including multi-robot problems in geometrically complex environments.

1 Introduction

Probabilistic roadmaps (PRM) have proven to be an effective tool to solve path-planning problems with many degrees of freedom (dofs) (Hsu, 2000; Kavraki, 1994; Kavraki, Svestka, Latombe, Overmars, 1996) and/or complex admissibility constraints (e.g., kinodynamic, stability, and visibility constraints) (Casal, 2001; Hsu, 2000; Kindel, 2001; Kuffner, 1999). A PRM planner samples the configuration space at random and retains the collision-free points as *milestones*. It connects pairs of milestones by simple paths (straight segments in configuration space) and retains the collision-free ones as *local paths*. The milestones and local paths form the *probabilistic roadmap*. The motivation is that, while it is often impractical to compute an explicit representation of the collision-free subset of a configuration space (the *free space*), algorithms exist that efficiently test if a given configuration or a local path is collision-free (Barraquand, Kavraki, Latombe, Li, Motwani, Raghavan, 1997; Gottschalk, Lin, Manocha, 1996). Under some assumptions, the probability that a PRM planner finds a collision-free path, if one exists, goes to 1 exponentially in the number of milestones (Hsu, 2000; Hsu, Latombe, Motwani, 1997). Hence, random sampling provides a convenient incremental path-planning scheme.

PRM planners spend most of their time performing collision checks. Several approaches are possible to reduce the overall cost of collision checking:

- *Design faster collision checkers.* However, several efficient algorithms already exist. Among them, hierarchical algorithms pre-compute a multi-level bounding approximation of every object in an environment (Gottschalk et al., 1996;

Quinlan, 1994). For each collision query, they use this approximation to quickly rule out large subsets of the objects that cannot collide. They scale up well to complex environments (Hsu et al., 1997).

- *Design sampling strategies yielding smaller roadmaps.* For example, the strategy in (Kavraki, 1994) produces a first roadmap by sampling configuration space uniformly; next, it picks additional milestones in neighborhoods of existing milestones with no or few connections to the rest of the roadmap. Other strategies generate a greater density of milestones near the boundary of the free space, as the connectivity of narrow regions is more difficult to capture than that of wide-open regions (Amato, Bayazit, Dale, Jones, Vallejo, 2002; Boor, Overmars, van der Strappen, 1999).
- *Delay collision checks until they are absolutely needed.* The planner in (Bohlin, Kavraki, 2000) first generates a network by distributing points at random in configuration space. It initially assumes that all points and connections between them are collision-free. It then computes the shortest path in this network between two query configurations and tests it for collision. If a collision is detected, the node and/or segment where it occurs are erased, and a new shortest path is computed and tested; and so on.

We think that delaying collision tests is a promising approach, but its potential has only been partially exploited in (Bohlin, Kavraki, 2000). One must decide in advance how large the network should be. If it is too coarse, it may fail to contain a solution path. But, if it is too dense, time will be wasted checking similar paths for collision. The focus on shortest paths may be costly when obstacles force the robot to take long detours.

In this paper, we present a new PRM planner – called SBL, for Single-query, Bi-directional, Lazy in collision checking – that tries to better exploit delayed collision checking, in particular by combining it with single-query, bi-directional, and adaptive sampling techniques, some of which were introduced in (Hsu, 2000; Hsu et al., 1997). SBL incrementally constructs a network of milestones made of two trees rooted at the query configurations, hence focusing its attention to the subset of the free space that is reachable from these configurations. It also locally adjusts the sampling resolution to take longer steps in opened regions of the free space and shorter ones in narrow regions. It does not immediately test connections between milestones for collision. Only when a sequence of milestones joining the two query configurations is found, the connections between milestones along this path are tested. This test is performed at successive points ordered according to their likelihood of revealing a collision. No time is wasted testing connections that are not on a candidate path and relatively little time is spent checking connections that are not collision-free. On a 1-GHz Pentium III processor, the planner reliably solves problems with 6-dof robots in times ranging from a small fraction of a second to a few seconds. Comparison with a similar planner using a traditional collision-checking strategy shows that lazy collision checking cuts planning times by factors from 4 to 40 in the environments of Fig. 1. SBL also solves multi-robot problems reliably and efficiently, like the one in Fig. 5 (36 dofs).

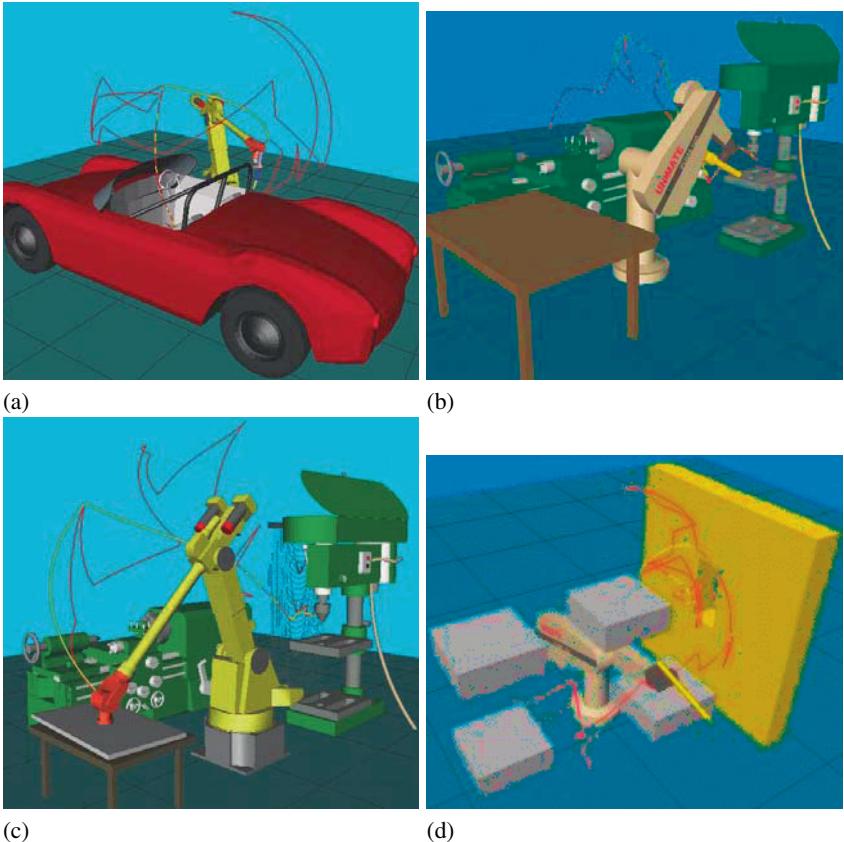


Fig. 1. Path planning environments

2 Definitions and Notations

Let C denote the configuration space of a robot and $F \subseteq C$ its free space. We normalize the range of values of each dof to be $[0,1]$ and we represent C as $[0, 1]^n$, where n is the number of dofs of the robot. We define a metric d over C . For any $q \in C$, the neighborhood of q of radius r is the subset $B(q, r) = \{q' \in C \mid d(q, q') < r\}$. With $d = L_\infty$ – the metric used by SBL – it is an n -D cube.

No explicit geometric representation of F is computed. Instead, given any configuration $q \in C$, a collision checker returns whether $q \in F$. A path τ in C is considered collision-free if a series of points on τ , such that every two successive points are closer apart than some ε , are all collision-free. A rigorous test (eliminating the need for ε) is possible when a distance-computation algorithm is used instead of a pure collision checker (Barraquand et al., 1997).

A path-planning *query* is defined by two *query configurations*, q_{init} and q_{goal} . If these configurations lie in the same connected component of F , the planner should

return a collision-free path between them; otherwise, it should indicate that no such path exists. There are two main classes of PRM planners: *multi-query* and *single-query*. A multi-query planner pre-computes a roadmap, which it later uses to process multiple queries (Kavraki, 1994; Kavraki et al., 1996). To deal with any possible query, the roadmap must be distributed over the entire free space. Instead, a single-query planner computes a new roadmap for each query (Hsu et al., 1997). The less space it explores to find a solution path, the better. Single-query planners are more suitable in environments with frequent changes.

A single-query planner either grows one tree of milestones from either q_{init} or q_{goal} , until a connection is found with the other query configuration (single-directional sampling), or grows two trees concurrently, respectively rooted at q_{init} and q_{goal} , until a connection is found between the two trees (bi-directional sampling) (Hsu, 2000). In both cases, milestones are iteratively added to the roadmap. Each new milestone m' is selected in a neighborhood of a milestone m already installed in a tree T , and is connected to m by a local path (hence, m' becomes a child of m in T). Bi-directional planners are usually more efficient than single-directional ones.

SBL is a single-query, bi-directional PRM planner. Unlike previous such planners, it does not immediately test the connections between milestones for collision. Therefore, rather than referring to the connection between two adjacent nodes in a roadmap tree as a *local path*, we will call it a *segment*.

3 Experimental Foundations

The design of SBL was suggested by experiments that we performed with the single-query PRM planner described in (Hsu, 2000). To study the impact of collision checking on the running time, we modified the planner's code by removing collision checks for connections between milestones. As we expected, the planner was faster by two to three orders of magnitude, but surprisingly a significant fraction of the generated paths were actually collision-free.

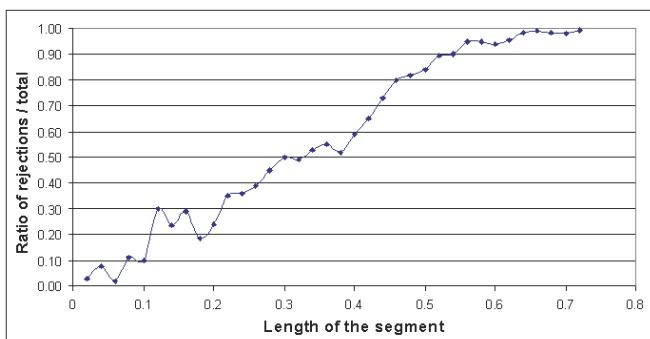


Fig. 2. Collision ratios

Every segment created by the planner of (Hsu, 2000) is relatively short. Thus, the above observation suggested that if two configurations picked at random are both collision-free and close to each other, then the straight-line segment between them has high prior probability of being collision-free. To verify this analysis, we generated 10,000 segments at random with L_∞ lengths uniformly distributed between 0 and 1 (recall that the L_∞ diameter of C is 1). This was done by picking 100 collision-free configurations in C uniformly at random and connecting each such configuration q to 100 additional collision-free configurations obtained by randomly sampling neighborhoods of q of different radii. We then tested each of the 10,000 segments for collision. The chart of Fig. 2 (generated for the environment of Fig. 1(a)) displays the ratio of the number of segments that tested collision-free, as a function of the lengths of these segments. Here, a segment shorter than 0.25 has probability greater than 0.6 of being collision-free. Similar charts were obtained with the other environments of Fig. 1. There is a simple explanation for this result. Since the robot and the obstacles are “thick” along all or most directions, the obstacle regions in C are also thick in most directions. Hence, a short colliding segment with collision-free endpoints is necessarily almost tangential to an obstacle region in C , an event that has small probability.

The above test and other tests led to making the following observations:

- Most local paths in a probabilistic roadmap are not on the final path. Using the planner of (Hsu, 2000) on the examples of Fig. 1, we measured that the ratio of milestones on the final path varies between 0.1 and 0.001.
- The test of a connection is most expensive when it is actually collision-free. Indeed, the test ends as soon as a collision is detected, but is carried down to the finest resolution when there is no collision.
- A short connection between two milestones has high prior probability of being collision-free. Thus, testing connections early is likely to be both useless and expensive.
- If a connection between milestones is colliding, its midpoint has high probability to be in collision; hence, this point should be tested next (a choice that can be applied recursively).

SBL’s lazy collision-checking strategy derives from these observations.

4 Description of SBL

SBL is given two parameters: s – the maximum number of milestones that it is allowed to generate – and ρ – a distance threshold. Two configurations are considered “close” to one another if their L_∞ distance is less than ρ . In our implementation, ρ is typically set between 0.1 and 0.3.

4.1 Overall algorithm

Algorithm PLANNER($q_{\text{init}}, q_{\text{goal}}$)

1. Install q_{init} and q_{goal} as the roots of T_{init} and T_{goal} , respectively
 2. Repeat s times
 - 2.1. EXPAND-TREE
 - 2.2. $\tau \leftarrow \text{CONNECT-TREES}$
 - 2.3. If $\tau \neq \text{nil}$ then return τ
 3. Return *failure*
-

The planner builds two trees of milestones, T_{init} and T_{goal} , respectively rooted at q_{init} and q_{goal} . At each loop of Step 2, EXPAND-TREE adds a milestone to one of the two trees, while CONNECT-TREES connects the two trees. The planner returns *failure* if it has not found a solution path after s iterations. If the planner returns *failure*, either no collision-free path exists between q_{init} and q_{goal} , or the planner actually failed to find one.

4.2 Tree expansion

Algorithm EXPAND-TREE

1. Pick T to be either T_{init} , or T_{goal} , each with probability 1/2
 2. Repeat until a new milestone q has been generated
 - 2.1. Pick a milestone m from T at random, with probability $\pi(m)$
 - 2.2. For $i = 1, 2, \dots$ until a new milestone q has been generated
 - 2.2.1. Pick a configuration q uniformly at random from $B(m, \rho/i)$
 - 2.2.2. If q is collision-free then install it in T as a child of m
-

Each expansion of the roadmap consists of adding a milestone to one of the two trees. The algorithm first selects the tree T to expand. Next, a milestone m is picked from T with probability $\pi(m)$ inverse to the current density of milestones of T around m . (Implementation details will be given in Subsection 4.5.) Finally, a collision-free configuration q is picked at distance less than ρ from m . This configuration is the new milestone. The use of the probability distribution $\pi(m)$ at Step 2.1 was introduced in (Hsu et al., 1997) to avoid over-sampling regions of F . It guarantees that the distribution of milestones eventually diffuses through the subsets of F reachable from q_{init} and q_{goal} . This condition is needed to prove that the planner finds a path with high probability, when one exists (Hsu, 2000; Hsu et al., 1997). The alternation between the two trees prevents any tree from eventually growing much bigger than the other, as the advantages of bi-directional sampling would then be lost.

Step 2.2 implements an adaptive sampling strategies, by selecting a series of milestone candidates, at random, from successively smaller neighborhoods of m , starting with a neighborhood of radius ρ . When a candidate q tests collision-free, it is retained as the new milestone. On the average, the jump from m to q is greater

in wide-open regions of F than in narrow regions. Note that the collision test of the segment from m to q is not done here; it will be done later if and when this segment belongs to a candidate path.

4.3 Tree connection

Algorithm CONNECT-TREES

1. $m \leftarrow$ most recently created milestone
2. $m' \leftarrow$ closest milestone to m in the tree not containing m
3. If $d(m, m') < \rho$ then
 - 3.1. Connect m and m' by a bridge w
 - 3.2. $\tau \leftarrow$ path connecting q_{init} and q_{goal}
 - 3.3. Return TEST-PATH(τ)
4. Return *nil*

Let m now denote the milestone that was just added by EXPAND-TREE. Let m' be the closest milestone to m in the other tree. The two trees are connected by a segment, called a *bridge*, joining m and m' if these two milestones are less than ρ apart. The bridge creates a path τ joining q_{init} and q_{goal} in the roadmap. The segments along τ , including the bridge, are now tested for collision. TEST-PATH returns *nil* if it detects a collision.

4.4 Path testing

SBL associates a collision-check index $\kappa(u)$ with each segment u between milestones (including the bridge). This index takes an integer value indicating the resolution at which u has already been tested. If $\kappa(u) = 0$, then only the two endpoints of u (which are both milestones) have been tested collision-free. If $\kappa(u) = 1$, then the two endpoints and the midpoint of u have been tested collision-free. More generally, for any $\kappa(u)$, $2^{\kappa(u)} + 1$ equally distant points of u have been tested collision-free. Let $\lambda(u)$ denote the length of u . If $2^{-\kappa(u)}\lambda(u) < \varepsilon$, then u is marked *safe*. The index of every new segment is initialized to 0.

Let $\sigma(u, j)$ designate the set of points in u that must have already been tested collision-free in order for $\kappa(u)$ to have the value j . The algorithm TEST-SEGMENT(u) increments $\kappa(u)$ by 1:

Algorithm TEST-SEGMENT(u)

1. $j \leftarrow \kappa(u)$
2. For every $q \in \sigma(u, j+1) \setminus \sigma(u, j)$ if q is in collision then return *collision*
3. If $2^{-(j+1)}\lambda(u) < \varepsilon$ then mark u *safe*, else $\kappa(u) \leftarrow j + 1$

For every segment u that is not marked *safe*, the current value of $2^{-\kappa(u)}\lambda(u)$ is cached in the data structure representing u . The smaller this value, the greater the probability that u is collision-free.

Let u_1, u_2, \dots, u_p denote all the segments in the path τ that are not already marked *safe*. TEST-PATH(τ) maintains a priority queue U of these segments sorted in decreasing order of $2^{-\kappa(u_i)}\lambda(u_i)$ ($i = 1$ to p). A similar technique has been previously used in (Nielsen, Kavraki, 2000).

Algorithm TEST-PATH(τ)

1. While U is not empty do
 - 1.1. $u \leftarrow \text{extract}(U)$
 - 1.2. If TEST-SEGMENT(u) = *collision* then
 - 1.2.1. Remove u from the roadmap
 - 1.2.2. Return *nil*
 - 1.3. If u is not marked *safe* then re-insert u into U
 2. Return τ
-

Each loop of Step 1 results in increasing the index of the segment u that is in first position in U . This segment is first removed from U . It is later re-inserted into U if TEST-SEGMENT(u) at Step 1.2 neither detects a collision, nor marks u *safe*. If u is re-inserted into U , it may not be in first position, since the quantity $2^{-\kappa(u')}\lambda(u)$ has been divided by 2. TEST-PATH terminates when a collision is detected – then the colliding segment is removed from the roadmap – or when all segments have been marked *safe* (i.e., U is empty) – then the path τ is returned.

The removal of a segment u disconnects again the roadmap into two trees. If u is the bridge that CONNECT-TREES created to connect the two trees, then the two trees return to their previous state (except for the collision-check indices of some segments, whose values may have increased). Otherwise, the removal of u results in a transfer of milestones from one tree to the other. Assume that u is in T_{goal} , as illustrated in Fig. 3(a), where $w \neq u$ denotes the bridge added by CONNECT-TREES. The milestones m_1, \dots, m_r between u and w ($r = 3$ in Fig. 3) and their children in T_{goal} are transferred to T_{init} as shown in Fig. 3(b). The parent-child connections between the transferred milestones remain the same, except those between m_1, \dots, m_r , which are inverted. So, no milestone is ever removed from the roadmap and the collision-checking work done along the segments is saved in their indices. If one of these segments later lies on another candidate path, then the tests previously done are not repeated.

4.5 Implementation details

SBL's collision checker is PQP (Gottschalk et al., 1996). Each obstacle and robot link is described by a collection of triangles representing its surface. PQP pre-computes a bounding hierarchical representation of each object using oriented-bounding boxes. No other pre-computation is done by the planner.

The planner spatially indexes every milestone of T_{init} (resp. T_{goal}) in an h -dimensional ($h = 2$ or 3) array A_{init} (resp. A_{goal}). Both arrays partition the subspace defined by h dimensions of C (in our implementation, $h = 2$) into the same grid of

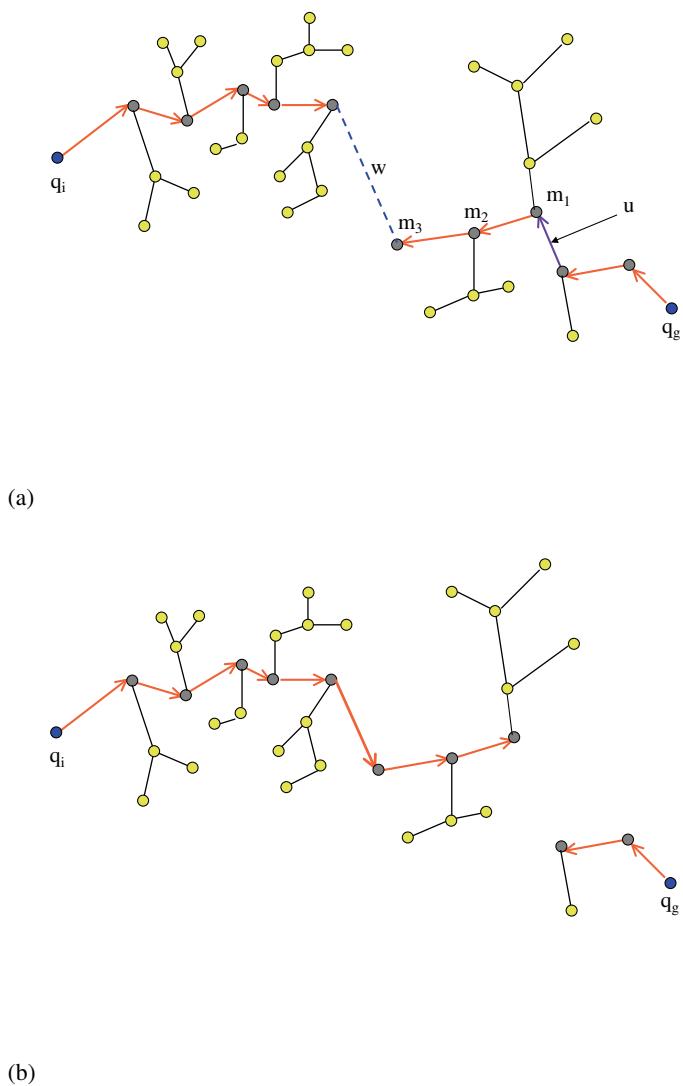


Fig. 3. Transfer of milestones from one tree to the other

equally sized cells. Whenever a new milestone q is installed in a tree, the appropriate cell of the corresponding array is updated to contain q . When a milestone is transferred from one tree into the other, the two arrays are updated accordingly. A_{init} and A_{goal} are used at Step 2.1 of EXPAND-TREE, where we pick a milestone m from one tree T with a probability distribution $\pi(m)$. Rather than maintaining the density of samples around each milestone, we do the following. Assume that $T = T_{\text{init}}$. We first pick a non-empty cell of A_{init} , then a milestone from this cell. Hence, the probability to pick a certain milestone is greater if this milestone lies in a cell of A_{init} containing fewer milestones. This technique is fast and results in a good diffusion of milestones in F along the h selected dimensions. To ensure diffusion along all dimensions of C , we periodically change the h dimensions. Each change requires re-constructing the arrays A_{init} and A_{goal} , but the total cost of this operation is negligible relative to collision tests.

Step 2 of CONNECT-TREES also uses A_{init} and A_{goal} to identify the milestone m' that will be connected to the newly added milestone m . Our implementation of CONNECT-TREES tries two connections: first, instead of selecting m' as the closest milestone to m in the other tree, it picks m' to be the closest milestones in the same cell as m , but in the other array (m and m' are then only guaranteed to be close to each other along h dimensions); then, it picks m' uniformly at random in the other tree. Our experiments have shown that this technique is faster on average than connecting m to the closest milestone. (The “closest-milestone” heuristic often delays the finding of some easy connections.)

Finally, we added a simple path optimizer to SBL to remove blatant jerks from paths. This optimizer takes a path τ as input and performs the following operation several times: pick two points q and q' in τ at random and, if the straight-line segment connecting them tests collision-free, replace the portion of τ between q and q' by this segment.

5 Experimental Results

SBL is written in C++. The running times reported below were obtained on a 1-GHz Pentium III processor with 1 GB of main memory running Linux. The distance threshold ρ was set to 0.15 and the resolution ε to 0.01. Each array A_{init} and A_{goal} had size 10×10 . The two dimensions of these arrays were changed every 50 milestones.

Fig. 1 displays some of the single-robot examples we used to test SBL. In each example, the robot is a 6-dof arm; the dark curve is traced by the center-point of its end-effector for one non-optimized path generated by SBL and the light curve is defined in the same way for the optimized path. The numbers of triangles in the robot and the obstacles, n_{rob} and n_{obs} , are indicated in Table 1. The geometrically simpler example of Fig. 1(d) was intended to test SBL when the free space contains narrow passages, a notorious difficulty for PRM planners. The time used by PQP to pre-compute the bounding hierarchies goes from 0.19s for the environment with

the fewest triangles (Fig. 1(d)) to 3.9s for the environment with the most triangles (Fig. 1(c)). It is not included in the running times of SBL given below.

Table 1. Number of triangles in robots and obstacles

1(a)		1(b)		1(c)		1(d)	
n_{rob}	n_{obs}	n_{rob}	n_{obs}	n_{rob}	n_{obs}	n_{rob}	n_{obs}
5,000	21,000	3,000	50,000	5,000	83,000	3,000	50

5.1 Basic performance evaluation

Table 2 gives statistics – average running time, standard deviation of running time, average time spent in collision checking, average number of milestones in final roadmaps, average number of milestones on generated paths, average number of collision checks, average number of checks on generated paths – over 100 runs of SBL on each of the four examples of Fig. 1. In all 400 runs, SBL found a path in small amount of time; there was no failure (the maximal number of milestones s was set to 10,000). A large fraction of the collision checks were made on the solution paths. The running times of Table 2 do not include path optimization, which in all cases took an additional 0.1 to 0.2s.

Table 2. Results on the examples of Fig. 1 (times are in seconds)

ex.	time	std	cc-t	mil	mil-p	#cc	#cc-p
1(a)	0.60	0.38	0.58	159	13	1483	342
1(b)	0.17	0.07	0.17	33	10	406	124
1(c)	4.42	1.86	4.17	1405	24	7267	277
1(d)	6.99	3.55	6.30	4160	44	12228	447

The chart of Fig. 4 was generated by running SBL many times on the example of Fig. 1(c) with increasing values of the maximum number s of milestones, from very small ones to larger ones (horizontal axis). For each value of s , we ran SBL 200 times with different seeds, and we counted the number of failures (vertical axis). When s is very small, SBL fails consistently. When it is sufficiently large, its success rate is 100%. The transition between consistent failure and consistent success is quite fast, which is coherent with the theoretical result that a PRM planner has fast convergence rate in the number of milestones (Hsu et al., 1997).

On several examples, we tried different values of ρ between 0.1 and 0.3, as well as indexing arrays of resolutions other than 10×10 , including 3-D arrays, but performance results were not significantly different.

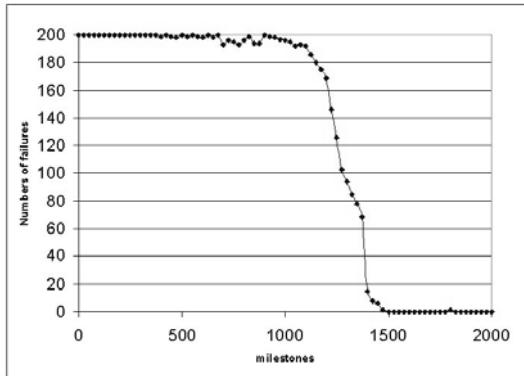


Fig. 4. Experimental convergence rate of SBL on the example of Fig. 1(c)

5.2 Comparative performance evaluation

To assess the efficiency of our lazy collision-checking strategy, we implemented a version of the planner that fully tests every segment between a milestone and a milestone candidate before inserting the later in the roadmap. This planner is similar to the one presented in (Hsu, 2000). Note, however, that our two planners do not exactly generate the same milestones, even when they use the same seed for the random number generator. Indeed, while SBL considers any collision-free configuration q picked in the neighborhood of a milestone m as a new milestone (Step 2.2 of EXPAND-TREE), the second planner requires in addition that the segment joining m and q be collision-free. Moreover, in the second planner no milestone is ever transferred from one tree to the other.

Table 3. Results with full-collision-check planner

ex.	time	std	cc-t	mil	mil-p	#cc	#cc-p
1(a)	2.82	3.01	2.81	22	5	7425	173
1(b)	1.03	0.70	1.02	29	9	2440	123
1(c)	18.46	15.34	18.35	771	16	38975	219
1(d)	293.77	122.75	292.40	6737	24	666084	300

Table 3 shows the results obtained with the full-collision-check planner, on the same four examples as above. The maximal number of milestones s was set to 10,000 and the results are statistics over 100 runs. The average running times (and numbers of collision checks) for SBL are smaller than for the full-collision-check planner by factors ranging from slightly over 4 for the problem of Fig. 1(c) to over 40 for the problem of Fig. 1(d). These results cannot be compared to those in

(Bohlin, Kavraki, 2000), where the improvement was measured relative to a *multi-query* planner, which must pre-compute a large roadmap to cover the entire free space.

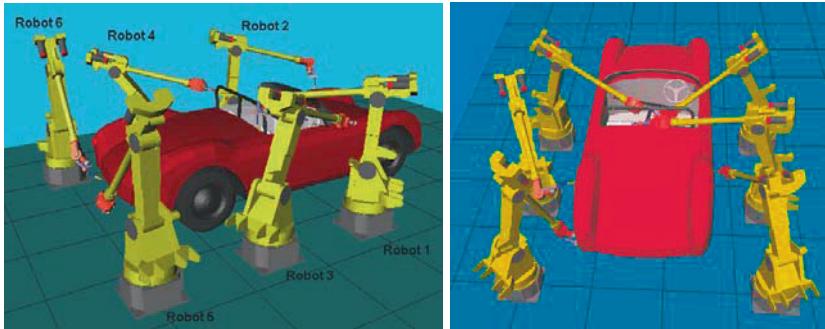


Fig. 5. Multi-robot problem

5.3 Multi-robot examples

We ran SBL on several problems in the environment of Fig. 5, which represents a welding station found in automotive body shops. This environment contains 6 robot arms with 6 dofs each. SBL treats them as a single robot with 36 dofs (centralized planning) and checks collisions between arms.

Table 4. Average running times (in seconds) on 9 multi-robot examples

PI-2	PI-4	PI-6	PII-2.	PII-4	PII-6	PIII-2	PIII-4	PIII-6
0.26	3.97	28.91	0.25	3.94	59.65	2.44	30.82	442.85

Table 4 gives the average running times on 9 problems, over 100 runs for each problem. Fig. 5 shows the initial and goal configurations for the problem named PIII-6. PIII-2 and PIII-4 are the same problem, but reduced to robots 1 and 2, and robots 1 through 4, respectively. The problems PI-2/4/6 and PII-2/4/6 are simpler, with one of the query configurations being the rest configuration of the arms. In all $100 \times 3 \times 3 = 900$ runs, SBL successfully found a path in a satisfactory amount of time. (The maximum number s of milestones for each run was set to 10,000.) The increase in running time when the number of arms goes from 2 to 4 to 6 is caused by both the growth in the number of pairs of bodies that must be tested at each collision-checking operation and by the greater difficulty of the problems due to the constraints imposed by the additional arms upon the motions of the other arms. A more thorough description and analysis of the application of SBL to multi-robot problems can be found in (Sánchez, Latombe, 2002a,b)

6 Conclusion

This paper shows that a PRM planner combining a lazy collision-checking strategy with single-query, bi-directional, and adaptive sampling techniques can solve path-planning problems of practical interest (i.e., with realistic complexity) in times ranging from fractions of a second to a few seconds for a single or few robots, and from seconds to few minutes for multiple robots. Improvements are still possible. For example, we could use PQP in its distance-computation mode to reduce the number of calls to this program (Barraquand et al., 1997).

Our main goal is to extend SBL to facilitate the programming of multi-robot spot-welding stations in automotive body shops. In particular, each robot must perform several welding operations, but the ordering of these locations is not fully specified. Hence, the planner will have to compute an optimized tour of the welding locations. This is a variant of the Traveling Salesman Problem where multiple “salesmen” visit “cities” concurrently and where the distance between any two “cities” is not given and, instead, must be computed by the planner.

Acknowledgements. This research was funded in part by grants from General Motors Research and ABB. G. Sánchez’s stay at Stanford was partially supported by ITESM (Campus Cuernavaca) and a fellowship from CONACyT. This work has greatly benefited from discussions with H. González-Baños, C. Guestrin, D. Hsu, L. Kavraki, and F. Prinz. PQP was developed in the Computer Science Dept. at the U. of North-Carolina.

References

- Amato NM, Bayazit OB, Dale LK, Jones C, Vallejo D (2002) OBPRM: An obstacle-based PRM for 3D workspace. In *Robotics: The Algorithmic Perspective*, pp 155–168. A.K. Peters, Natick, MA.
- Barraquand J, Kavraki LE, Latombe JC, Li TY, Motwani R, Raghavan P (1997) A random sampling scheme for path planning. *Int. J. of Robotics Research*, 16(6):759–774.
- Bohlin R, Kavraki LE (2000) Path planning using lazy PRM. In Proc. IEEE Int. Conf. Robotics & Autom., San Francisco, CA.
- Boor V, Overmars MH, van der Strassen AF (1999) The gaussian sampling strategy for probabilistic roadmap planners. In IEEE Int. Conf. Robotics & Autom., pp 1018–1023, Detroit, MI.
- Casal A (2001) Reconfiguration planning for modular self-reconfigurable robots. PhD thesis, Aeronautics & Astronautics Dept., Stanford U., CA.
- Gottschalk S, Lin M, Manocha D (1996) OBB-Tree: A hierarchical structure for rapid interference detection. In Proc. ACM SIGGRAPH’96, pp 171–180.
- Hsu D (2000) Randomized single-query motion planning in expansive spaces. PhD thesis, Computer Science Dept., Stanford University, CA.
- Hsu D, Latombe JC, Motwani R (1997) Path planning in expansive configuration spaces. In Proc. IEEE Int. Conf. Robotics & Autom., pp 2719–2726.
- Kavraki LE (1994) Random networks in configuration space for fast path planning. PhD thesis, Computer Science Dept., Stanford University, CA.
- Kavraki LE, Svestka P, Latombe JC, Overmars MH (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robotics & Autom.*, 12 (4):566–580.

- Kindel R (2001) Motion planning for free-flying robots in dynamic and uncertain environments. PhD thesis, Aeronaut. & Astr. Dept., Stanford University, CA.
- Kuffner JJ (1999) Autonomous agents for real-time animation. PhD thesis, Computer Science Dept., Stanford University, CA.
- Nielsen C, Kavraki LE (2000) A two-level fuzzy PRM for manipulation planning. In Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.
- Quinlan S (1994) Efficient distance computation between non-convex objects. In Proc. Int. IEEE Conf. Robotics & Autom., pp 3324–3329.
- Sánchez G, Latombe JC (2002a) On delaying collision checking in PRM planning – application to multi-robot coordination. To appear in Int. J. of Robotics Research.
- Sánchez G, Latombe JC (2002b) Using a PRM planner to compare centralized and decoupled planning for multi-robot systems. In Proc. Int. IEEE Conf. Robotics & Autom., Washington, D.C.

Geometrical Interpretation of the CCT Stiffness Mapping for Serial Manipulators

Chintien Huang¹ and Imin Kao²

¹ Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan

² Department of Mechanical Engineering, State University of New York, Stony Brook, NY 11794, USA

Abstract. Recent research results suggested a conservative transformation to correct the well-known congruence transformation between Cartesian and joint stiffness matrices of a serial manipulator. This paper utilizes screw geometry to interpret the conservative congruence transformation (CCT). The analysis using screw theory provides better geometric insights into the CCT. The effective geometrical stiffness matrix, due to the change of manipulator geometry under stiffness control in the presence of external force, is confirmed. This paper also points out several erroneous assumptions that may have led to the incorrect formulation of the conventional congruence transformation.

1 Introduction

In practice, a robot will deviate from its nominal position when exerting some force and moment on its environment. The deviation depends on the stiffness of the robot. Stiffness control strategies account for stiffness in the feedback control of robot manipulators and can be utilized to respond to unexpected external disturbance. The congruence transformation between the Cartesian stiffness matrix and the corresponding joint stiffness matrix is an essential operator for stiffness control of serial manipulators. The conventional congruence transformation for serial manipulators was developed by Salisbury (Salisbury, 1980, 1982) and has been well accepted by researchers until recently. Then Chen and Kao (Chen, Kao, 1999, 2000) discovered that the conventional congruence transformation was not conservative (Kao, Ngo, 1999). Namely, there is a discrepancy between the work done in the joint and Cartesian spaces when conventional formulation is applied to robot stiffness control.

Chen and Kao (Chen, Kao, 1999, 2000) proposed a conservative congruence transformation (CCT) by considering the change of Jacobian matrix (geometry) due to compliance. The effect of the change in geometry under external loads is captured by a matrix denoted by K_g . This paper aims to give geometrical insights into the conservative congruence transformation and to provide a foundation for the new formulation of stiffness control. We utilize a very concise but powerful tool in kinematic geometry–screw theory (Ball, 1986; Hunt, 1978; Roth, 1984). The geometrical meaning of K_g is illustrated by using the screw-based Jacobian matrix (Tsai, 1999; Waldron, Wang, Bolin, 1985) and the principle of virtual work. A planar 3R example is used to illustrate the concepts.

This paper also discusses the erroneous derivations found in several textbooks in robotics. As will be shown in this paper, these derivations are usually based

on mathematically or geometrically incorrect assumptions, which lead to the non-conservative mapping.

2 Screw-Based Jacobian Matrices of Serial Manipulators

The Plücker coordinates of a line are the combination of two Cartesian vectors:

$$\hat{\mathbf{L}} = \begin{bmatrix} \hat{\mathbf{q}} \\ \mathbf{r} \times \hat{\mathbf{q}} \end{bmatrix}$$

where $\hat{\mathbf{q}}$ denotes the direction-cosine vector of the line, and \mathbf{r} is the position vector of a point on the line. In this paper, we use the symbol \wedge over a boldface lower case letter to denote a unit vector. A screw is a line with an associated pitch. The coordinates of a screw are

$$\hat{\mathbf{S}} = \begin{bmatrix} \hat{\mathbf{q}} \\ \mathbf{r} \times \hat{\mathbf{q}} + p\hat{\mathbf{q}} \end{bmatrix}$$

where p denotes the pitch of the screw. Screws are powerful geometrical and mathematical tools in kinematics and statics.

In instantaneous kinematics, the first-order motion of a rigid body can be regarded as a velocity screw, usually referred to as an instantaneous twist. The instantaneous motion of the body can be thought of as rotating about, and translating along the screw axis. The coordinates of an instantaneous twist are

$$\hat{\mathbf{V}} = \omega\hat{\mathbf{S}} = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v}_p \end{bmatrix}$$

where $\boldsymbol{\omega}$ denotes the angular velocity vector of the body, and \mathbf{v}_p is the velocity of the point coincident with the origin of the coordinate system. The magnitude of $\boldsymbol{\omega}$ is denoted by ω . For example, as shown in Fig. 1, the velocity state of the end effector of a manipulator is described as a twist \mathbf{V} , and point P is the reference point on the end effector.

In statics, a force system applied on a body can be regarded as a force screw, usually referred to as a wrench. A wrench is the combination of a force acting along a screw axis and a torque acting about the same screw axis. The radial coordinates of a wrench are

$$\mathbf{W}_r = f\hat{\mathbf{S}} = \begin{bmatrix} \mathbf{f} \\ \mathbf{m}_p \end{bmatrix}$$

where \mathbf{f} is the force, and \mathbf{m}_p is the moment taken about the origin of the coordinate system. The magnitude of \mathbf{f} is denoted by f . In order to take advantage of vector operations, we can interchange the two vectors and use the axial coordinates of the wrench:

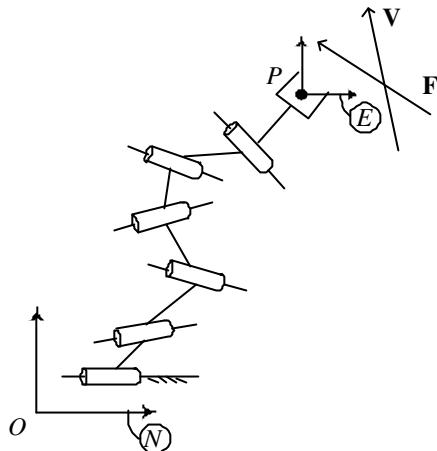


Fig. 1. Twist and wrench associated with a manipulator

$$\mathbf{F} = \begin{bmatrix} \mathbf{m}_p \\ \mathbf{f} \end{bmatrix}$$

As shown in Fig. 1, the wrench \mathbf{F} applied on the end effector is represented in the reference frame whose origin is at point P . The principle of virtual work for a body under static equilibrium can be expressed as the inner product of \mathbf{F} and \mathbf{V} , i.e.,

$$\mathbf{F} \cdot \mathbf{V} = \mathbf{F}^T \mathbf{V} = 0$$

where \mathbf{F}^T denotes the transpose of \mathbf{F} .

The Jacobian matrix of a serial manipulator relates its joint velocities $\dot{\theta}$ to the velocity field of its end effector as

$$\mathbf{V} = \mathbf{J}_\theta \dot{\theta}$$

In terms of infinitesimal displacements, we have

$$d\mathbf{X} = \mathbf{J}_\theta d\theta \quad (1)$$

The above equation can be derived in any coordinate system, such as the reference frame N , shown in Fig. 1. The Jacobian matrix defined in (1) is usually referred to as the conventional Jacobian matrix (Tsai, 1999).

According to screw theory, the Jacobian matrix is composed of joint screws, i.e.,

$$\mathbf{J}_\theta = (\hat{\mathbf{S}}_1 \hat{\mathbf{S}}_2 \hat{\mathbf{S}}_3 \hat{\mathbf{S}}_4 \hat{\mathbf{S}}_5 \hat{\mathbf{S}}_6) \quad (2)$$

where $\hat{\mathbf{S}}_i$, $i = 1, 2, \dots, 6$ denote the unit screw coordinates (column vectors) of the joint axes. The Jacobian matrix defined in (2) is called the screw-based Jacobian

matrix (Tsai, 1999). In order to make the Jacobian matrices given in (1) and (2) identical, the screws must be represented in the coordinate system whose origin is at the reference point P . And the two reference frames, E and N must be parallel. Applying the principle of virtual work to the analysis of the manipulator, the joint torques τ and the wrench applied on the end effector are related as follows:

$$\tau = J_\theta^T F \quad (3)$$

3 The Conservative Congruence Transformation

The joint stiffness matrix of a serial manipulator, K_p relates the infinitesimal joint torque changes to the infinitesimal joint displacements. That is,

$$d\tau = K_\theta d\theta \quad (4)$$

where K_θ is usually treated as a diagonal matrix. The links of the manipulator are assumed to be rigid. The stiffness of a manipulator affects the position accuracy of its end effector. The Cartesian stiffness matrix K_p of a manipulator relates the infinitesimal change of a wrench applied on the end effector to the infinitesimal displacement of the end effector at an equilibrium position. That is,

$$dF = K_p dX \quad (5)$$

Note that K_p is configuration-dependent and needs to be calculated from K_θ . The mapping between the Cartesian stiffness matrix and the joint stiffness matrix is a congruence transformation. The widely accepted congruence transformation (Salisbury, 1980, 1982) is

$$K_\theta = J_\theta^T K_p J_\theta$$

Chen and Kao (Chen, Kao, 1999, 2000) showed that the above transformation is valid only when there is no external load. Since it is of no practical use to talk about stiffness control in robotics without considering external loads, the conventional congruence transformation is incorrect. We will elaborate on this in the succeeding sections.

The derivation of the conservative congruence transformation is as follows. Differentiating (3) gives

$$d\tau = dJ_\theta^T F + J_\theta^T dF \quad (6)$$

Substituting (4) and (5) into the above equation gives

$$\mathbf{K}_\theta d\boldsymbol{\theta} = d\mathbf{J}_\theta^T \mathbf{F} + \mathbf{J}_\theta^T \mathbf{K}_p d\mathbf{X} \quad (7)$$

The first term in the right-hand side of (7), $d\mathbf{J}_\theta^T \mathbf{F}$ makes the conservative congruence transformation different from the conventional congruence transformation. It accounts for the change of the Jacobian matrix due to joint compliance. Note that

$$d\mathbf{J}_\theta^T = \sum_{i=1}^6 \frac{\partial \mathbf{J}_\theta^T}{\partial \theta_i} d\theta_i \quad (8)$$

Substituting (1) and (8) into (7) gives

$$\mathbf{K}_\theta d\boldsymbol{\theta} = \mathbf{K}_g d\boldsymbol{\theta} + \mathbf{J}_\theta^T \mathbf{K}_p \mathbf{J}_\theta d\boldsymbol{\theta} \quad (9)$$

where

$$\begin{aligned} \mathbf{K}_g &= \left[\frac{\partial \mathbf{J}_\theta^T}{\partial \theta_1} \mathbf{F} \quad \frac{\partial \mathbf{J}_\theta^T}{\partial \theta_2} \mathbf{F} \quad \dots \quad \frac{\partial \mathbf{J}_\theta^T}{\partial \theta_6} \mathbf{F} \right] \\ &= \left[\begin{array}{cccccc} \frac{\partial \hat{\mathbf{S}}_1^T}{\partial \theta_1} \mathbf{F} & \frac{\partial \hat{\mathbf{S}}_1^T}{\partial \theta_2} \mathbf{F} & \dots & \frac{\partial \hat{\mathbf{S}}_1^T}{\partial \theta_6} \mathbf{F} \\ \frac{\partial \hat{\mathbf{S}}_2^T}{\partial \theta_1} \mathbf{F} & \frac{\partial \hat{\mathbf{S}}_2^T}{\partial \theta_2} \mathbf{F} & \dots & \frac{\partial \hat{\mathbf{S}}_2^T}{\partial \theta_6} \mathbf{F} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial \hat{\mathbf{S}}_6^T}{\partial \theta_1} \mathbf{F} & \frac{\partial \hat{\mathbf{S}}_6^T}{\partial \theta_2} \mathbf{F} & \dots & \frac{\partial \hat{\mathbf{S}}_6^T}{\partial \theta_6} \mathbf{F} \end{array} \right] \end{aligned}$$

From (9), we have

$$\mathbf{K}_\theta = \mathbf{K}_g + \mathbf{J}_\theta^T \mathbf{K}_p \mathbf{J}_\theta$$

The inverse of the above transformation is the conservative congruence transformation:

$$\mathbf{K}_p = \mathbf{J}_\theta^{-T} (\mathbf{K}_\theta - \mathbf{K}_g) \mathbf{J}_\theta^{-1}$$

4 The Interpretation of \mathbf{K}_g

According to (3), the joint torque required to statically balance the external load is the scalar product of the wrench with the corresponding joint screw, i.e.,

$$\tau_i = \hat{\mathbf{S}}_i \cdot \mathbf{F} = \hat{\mathbf{S}}_i^T \mathbf{F}, \quad i = 1, 2, \dots, 6. \quad (10)$$

Without considering joint compliance, differentiating (3) yields the infinitesimal torques needed to balance an infinitesimal applied wrench:

$$d\boldsymbol{\tau} = \mathbf{J}_\theta^T d\mathbf{F}$$

or

$$d\tau_i = \hat{\mathbf{S}}_i^T d\mathbf{F}, \quad i = 1, 2, \dots, 6.$$

Note that in the above equation, the joint screws (and thus the Jacobian matrix) are treated as constants. When considering joint compliance, differentiating (3) should give

$$d\boldsymbol{\tau} = d\boldsymbol{\tau}_g + \mathbf{J}_\theta^T d\mathbf{F}$$

In the above equation, the term $d\boldsymbol{\tau}_g$ is the extra joint torque needed to balance the external load when joint compliance is considered. As discussed in the previous section,

$$d\boldsymbol{\tau}_g = \mathbf{K}_g d\boldsymbol{\theta}$$

Expanding the above equation gives the following expression of individual joint torque change:

$$d\tau_{ig} = \frac{\partial \hat{\mathbf{S}}_i^T}{\partial \theta_1} \mathbf{F} d\theta_1 + \frac{\partial \hat{\mathbf{S}}_i^T}{\partial \theta_2} \mathbf{F} d\theta_2 + \cdots + \frac{\partial \hat{\mathbf{S}}_i^T}{\partial \theta_6} \mathbf{F} d\theta_6, \quad i = 1, 2, \dots, 6.$$

When the external load is applied to the end effector, the joint angles deviate from their nominal positions, and the joint screws change too. The deviation of any single joint angle will alter the screw coordinates of all six joints. From (10), we know that a joint torque is the inner product of the applied wrench with the corresponding joint screw. To calculate the extra joint torque at any one joint, we need to account for changes in its joint screw coordinates due to the deviations of all six joint angles. Therefore $d\tau_{ig}$ is the sum of the inner products of the applied wrench with the changes in the i^{th} joint screw coordinates due to the deviations of all six joint angles.

5 Example of the Planar 3R Manipulator

In this section, a planar 3R example is used to illustrate the concepts discussed in the previous sections. As shown in Fig. 2, the origin of the fixed reference frame is located at the fixed pivot, O . A reference point on the end effector, P , is chosen to describe the velocity state of the end effector. Let ω denote the angular velocity of the end effector, and (v_x, v_y) denotes the velocity vector of point P . The velocity analysis of the manipulator gives the Jacobian matrix:

$$\begin{pmatrix} \omega \\ v_x \\ v_y \end{pmatrix} = \mathbf{J}_\theta \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} \quad (11)$$

where

$$\mathbf{J}_\theta = \begin{pmatrix} 1 & 1 & 1 \\ -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{c123} & l_2 c_{12} + l_3 c_{c123} & l_3 c_{c123} \end{pmatrix}$$

and

$$s_{12} = \sin(\theta_1 + \theta_2)$$

$$c_{12} = \cos(\theta_1 + \theta_2)$$

$$s_{123} = \sin(\theta_1 + \theta_2 + \theta_3)$$

$$c_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$$

Let the external force applied on the end effector be described by a force vector (f_x, f_y) passing through point P with an associated moment m_p

The Jacobian matrix given in (11) is the conventional Jacobian matrix, derived by the velocity analysis of the manipulator in reference frame N. In order to use screw geometry, we have to derive a screw-based Jacobian matrix. As mentioned before, one main difference between the conventional Jacobian matrix and the screw-based Jacobian matrix is the choice of the origin of the reference frame. To make the screw-based Jacobian matrix consistent with the conventional Jacobian matrix of the 3R manipulator, we need to choose a reference frame, E, whose origin is at point P. The two reference frames, E and N, are parallel. The screws (represented in E) for the R joints are derived as follows.

$$\hat{\mathbf{S}}_1^T = (0, 0, 1, -l_1 s_1 - l_2 s_{12} - l_3 s_{123}, l_1 c_1 + l_2 c_{12} + l_3 c_{c123}, 0)$$

$$\hat{\mathbf{S}}_2^T = (0, 0, 1, -l_2 s_{12} - l_3 s_{123}, l_2 c_{12} + l_3 c_{c123}, 0)$$

$$\hat{\mathbf{S}}_3^T = (0, 0, 1, l_3 s_{123}, l_3 c_{c123}, 0)$$

The external force is expressed as a wrench (represented in E):

$$\mathbf{F}^T = (0, 0, m_p, f_x, f_y, 0)$$

According to the discussion given in the previous sections, by omitting the zeros, we have

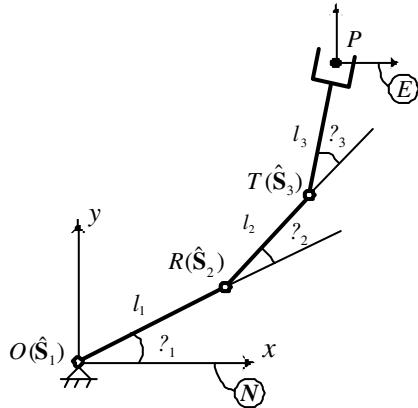


Fig. 2. A planar 3R manipulator

$$\mathbf{J}_\theta^T = \begin{pmatrix} \hat{\mathbf{S}}_1^T \\ \hat{\mathbf{S}}_2^T \\ \hat{\mathbf{S}}_3^T \end{pmatrix} = \begin{pmatrix} 1 & -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ 1 & -l_2 s_{12} - l_3 s_{123} & l_2 c_{12} + l_3 c_{123} \\ 1 & -l_3 s_{123} & l_3 c_{123} \end{pmatrix}$$

which is consistent with the conventional Jacobian matrix.

The partial derivative of the Jacobian matrix with respect to θ_1 is the change of joint screw coordinates due to the infinitesimal change of θ_1 :

$$\frac{\partial \mathbf{J}_\theta^T}{\partial \theta_1} = \begin{pmatrix} \frac{\partial \hat{\mathbf{S}}_1^T}{\partial \theta_1} \\ \frac{\partial \hat{\mathbf{S}}_2^T}{\partial \theta_1} \\ \frac{\partial \hat{\mathbf{S}}_3^T}{\partial \theta_1} \end{pmatrix} = \begin{pmatrix} 0 & -l_1 c_1 - l_2 c_{12} - l_3 c_{123} & -l_1 s_1 - l_2 s_{12} - l_3 s_{123} \\ 0 & -l_2 c_{12} - l_3 c_{123} & -l_2 s_{12} - l_3 s_{123} \\ 0 & -l_3 c_{123} & -l_3 s_{123} \end{pmatrix} \quad (12)$$

As shown in Fig. 3, the joint angle θ_1 is deviated from its nominal position by $\delta\theta_1$. The reference point is displaced from P to P' . The new joint screw coordinates have to be calculated with respect to a parallel reference frame, E' , whose origin is located at P' . The change of $\hat{\mathbf{S}}_1$ due to the change of θ_1 is

$$\begin{aligned} \delta \mathbf{S}_1 &= (\hat{\mathbf{S}}_1)_{E'} - (\hat{\mathbf{S}}_1)_E = \left(\mathbf{r}_{P'O} \times \mathbf{q} \right) - \left(\mathbf{r}_{PO} \times \mathbf{q} \right) \\ &= \left((\mathbf{r}_{P'O} - \mathbf{r}_{PO}) \times \mathbf{q} \right) = \left(\mathbf{r}_{P'P} \times \mathbf{q} \right) \end{aligned}$$

where

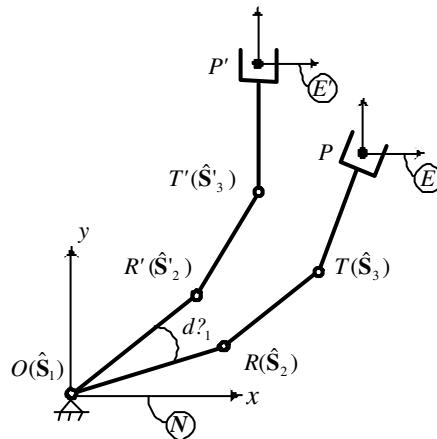


Fig. 3. Displacement due to the compliance of the first joint

$$\mathbf{q} = (0, 0, 1)^T$$

$$\mathbf{r}_{P'P} = \mathbf{r}_{OP} \times \delta\theta_1 \mathbf{q}$$

Here we use the notation $\mathbf{r}_{OP}(r_{opx}, r_{opy})$ to denote the vector measured from point O to point P . Therefore,

$$\frac{\partial \mathbf{S}_1}{\partial \theta_1} = \frac{\delta \mathbf{S}_1}{\delta \theta_1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -r_{opx} \\ -r_{opy} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -l_1 c_1 - l_2 c_{12} - l_3 c_{123} \\ -l_1 s_1 - l_2 s_{12} - l_3 s_{123} \\ 0 \end{pmatrix}$$

If we omit the zeros, the result is identical to that in (12). Similarly, the changes of $\hat{\mathbf{S}}_2$ and $\hat{\mathbf{S}}_3$ due to the change of θ_1 are, respectively,

$$\begin{aligned} \delta \mathbf{S}_2 &= (\hat{\mathbf{S}}'_2)_{E'} - (\hat{\mathbf{S}}_2)_E = \left(\begin{array}{c} \mathbf{q} \\ \mathbf{r}_{P'R'} \times \mathbf{q} \end{array} \right) - \left(\begin{array}{c} \mathbf{q} \\ \mathbf{r}_{PR} \times \mathbf{q} \end{array} \right) \\ &= \left(\begin{array}{c} \boldsymbol{\theta} \\ (\mathbf{r}_{P'R'} - \mathbf{r}_{PR}) \times \mathbf{q} \end{array} \right) = \left(\begin{array}{c} \boldsymbol{\theta} \\ (\mathbf{r}_{P'P} - \mathbf{r}_{R'R}) \times \mathbf{q} \end{array} \right) \\ \delta \mathbf{S}_3 &= (\hat{\mathbf{S}}'_3)_{E'} - (\hat{\mathbf{S}}_3)_E = \left(\begin{array}{c} \mathbf{q} \\ \mathbf{r}_{P'T'} \times \mathbf{q} \end{array} \right) - \left(\begin{array}{c} \mathbf{q} \\ \mathbf{r}_{PT} \times \mathbf{q} \end{array} \right) \\ &= \left(\begin{array}{c} \boldsymbol{\theta} \\ (\mathbf{r}_{P'T'} - \mathbf{r}_{PT}) \times \mathbf{q} \end{array} \right) = \left(\begin{array}{c} \boldsymbol{\theta} \\ (\mathbf{r}_{P'P} - \mathbf{r}_{T'T}) \times \mathbf{q} \end{array} \right) \end{aligned}$$

The partial derivatives of the Jacobian matrix with respect to θ_2 and θ_3 are, respectively,

$$\frac{\partial \mathbf{J}_\theta^T}{\partial \theta_2} = \begin{pmatrix} \frac{\partial \hat{\mathbf{S}}_1^T}{\partial \theta_2} \\ \frac{\partial \hat{\mathbf{S}}_2^T}{\partial \theta_2} \\ \frac{\partial \hat{\mathbf{S}}_3^T}{\partial \theta_2} \end{pmatrix} = \begin{pmatrix} 0 & -l_2 c_{12} - l_3 c_{123} & -l_2 s_{12} - l_3 s_{123} \\ 0 & -l_2 c_{12} - l_3 c_{123} & -l_2 s_{12} - l_3 s_{123} \\ 0 & -l_3 c_{123} & -l_3 s_{123} \end{pmatrix} \quad (13)$$

$$\frac{\partial \mathbf{J}_\theta^T}{\partial \theta_3} = \begin{pmatrix} \frac{\partial \hat{\mathbf{S}}_1^T}{\partial \theta_3} \\ \frac{\partial \hat{\mathbf{S}}_2^T}{\partial \theta_3} \\ \frac{\partial \hat{\mathbf{S}}_3^T}{\partial \theta_3} \end{pmatrix} = \begin{pmatrix} 0 & -l_3 c_{123} & -l_3 s_{123} \\ 0 & -l_3 c_{123} & -l_3 s_{123} \\ 0 & -l_3 c_{123} & -l_3 s_{123} \end{pmatrix} \quad (14)$$

The effectiveness geometrical stiffness matrix \mathbf{K}_g is determined as follows:

$$\mathbf{K}_g = \begin{bmatrix} K_{g11} & K_{g12} & K_{g13} \\ K_{g21} & K_{g22} & K_{g23} \\ K_{g31} & K_{g32} & K_{g33} \end{bmatrix}$$

where

$$K_{g11} = (-l_1 c_1 - l_2 c_{12} - l_3 c_{123}) f_x + (-l_1 s_1 - l_2 s_{12} - l_3 s_{123}) f_y$$

$$K_{g12} = (-l_2 c_{12} - l_3 c_{123}) f_x + (-l_2 s_{12} - l_3 s_{123}) f_y$$

$$K_{g13} = (-l_3 c_{123}) f_x + (-l_3 s_{123}) f_y$$

$$K_{g22} = (-l_2 c_{12} - l_3 c_{123}) f_x + (-l_2 s_{12} - l_3 s_{123}) f_y$$

$$K_{g23} = -l_3 c_{123} f_x - l_3 s_{123} f_y$$

$$K_{g33} = -l_3 c_{123} f_x - l_3 s_{123} f_y$$

$$K_{g21} = K_{g12}$$

$$K_{g31} = K_{g13}$$

$$K_{g32} = K_{g23}$$

Since the joint axis directions of a planar manipulator do not change, the applied moment does not contribute to stiffness matrix. Note that \mathbf{K}_g for the planar 3R manipulator is symmetric. However, \mathbf{K}_g is, in general, asymmetric for spatial manipulators.

6 Discussion

The congruence transformation of stiffness matrices derived by Salisbury (Salisbury, 1980, 1982) has been well known for two decades. Many textbooks in

robotics (Asada, Slotine, 1986; Mckerrow, 1991; Schilling, 1990; Tsai, 1999) have adopted Salisbury's formulation. In this section, we discuss how researchers might have been misled by erroneous assumptions when deriving the congruence transformation for stiffness control.

Here we summarize Salisbury's formulation of the congruence transformation. First the Cartesian and joint stiffness matrices are defined as follows:

$$\mathbf{F} = \mathbf{K}_p \delta \mathbf{X} \quad (15)$$

$$\boldsymbol{\tau} = \mathbf{K}_\theta \delta \boldsymbol{\theta} \quad (16)$$

where $\delta \mathbf{X}$ and $\delta \boldsymbol{\theta}$ denote virtual (infinitesimal) displacements of the end effector and the joints, respectively. Substituting the above equations into $\boldsymbol{\tau} = \mathbf{J}_\theta^T \mathbf{F}$ (3) yields

$$\mathbf{K}_\theta = \mathbf{J}_\theta^T \mathbf{K}_p \mathbf{J}_\theta$$

Note that (15) and (16) are mathematically incorrect because a finite quantity \mathbf{K}_p or \mathbf{K}_θ multiplied by an infinitesimal quantity $\delta \mathbf{X}$ (or $\delta \boldsymbol{\theta}$) should give an infinitesimal quantity, not a finite quantity \mathbf{F} (or $\boldsymbol{\tau}$). Eqs. (4) and (5) should have been used for the definitions of stiffness matrices.

Another misleading derivation that is commonly found in many robotics textbooks is as follows. The stiffness matrices are usually defined correctly as

$$\mathbf{F} = \mathbf{K}_p \Delta \mathbf{X} \quad (17)$$

$$\boldsymbol{\tau} = \mathbf{K}_\theta \Delta \boldsymbol{\theta} \quad (18)$$

where $\Delta \mathbf{X}$ and $\Delta \boldsymbol{\theta}$ denote a small (but finite) displacements. These definitions of stiffness matrices are correct because $\Delta \mathbf{X}$ and $\Delta \boldsymbol{\theta}$ are finite quantities, provided that the stiffness matrix (\mathbf{K}_p or \mathbf{K}_θ) remains constant over the range of the finite displacement. Then Eqs. (17) and (18) are substituted into the definition of Jacobian matrix $d\mathbf{X} = \mathbf{J}_\theta d\boldsymbol{\theta}$ (1) to obtain

$$\mathbf{K}_p^{-1} \mathbf{F} = \mathbf{J}_\theta \mathbf{K}_\theta^{-1} \boldsymbol{\tau}$$

Substituting $\boldsymbol{\tau} = \mathbf{J}_\theta^T \mathbf{F}$ (3) into the above equation yields

$$\mathbf{K}_\theta = \mathbf{J}_\theta^T \mathbf{K}_p \mathbf{J}_\theta$$

Note that the finite quantities ΔX and $\Delta \theta$ have been approximated by the infinitesimal quantities dX and $d\theta$ respectively. One may use dX and $d\theta$ to approximate and when compliance is not considered; however, we cannot ignore small deviations when dealing with compliance and stiffness. The deflection of a robot due to compliance, though small, cannot be considered infinitesimal. Otherwise it is not useful to discuss the small displacement due to joint compliance of the robot.

Yet another erroneous derivation can result from using the following differentiation of the equilibrium equation $\tau = J_\theta^T F$ (3). Taking the derivative of the equation and considering the Jacobian matrix constant, we have

$$d\tau = J_\theta^T dF \quad (19)$$

Substituting the definitions of stiffness matrices, (4) and $dF = K_p dX$ (5), into the above equation gives

$$K_\theta = J_\theta^T K_p J_\theta$$

Note that here the Jacobian matrix has been considered constant. (19) simply gives the change of joint torque required to balance the change of external force at an equilibrium position. Therefore stiffness (or compliance) is not considered. On the other hand, in $d\tau = dJ_\theta^T F + J_\theta^T dF$ (6), the term $(dJ_\theta^T F)$ accounts for the change of geometry due to compliance.

7 Conclusion

This paper presents the geometrical interpretation of the conservative congruence transformation of stiffness matrices, initially proposed by Chen and Kao, for serial manipulators. First, the screw-based Jacobian matrix was introduced. Appropriate coordinate systems were chosen to make the screw-based Jacobian matrix consistent with the conventional Jacobian matrix. Second, the geometrical meanings of the stiffness matrix K_g which accounts for the change in geometry due to compliance in the presence of external loads, were provided according to screw theory. Third, the analysis of the planar 3R manipulator was provided to illustrate these concepts. Finally, several derivations that may have led to the erroneous congruence transformation were discussed in detail. We hope that this paper can serve as a helpful supplement to better understand the conservative congruence transformation of stiffness matrices and to correct the erroneous formulations widely used in the stiffness control of manipulators.

Acknowledgement.

The financial support of the National Science Council of Taiwan (NSC 89-2212-E006-171) and the NSF grant IIS-9906890 is gratefully acknowledged.

References

- Asada H, Slotine JE (1986) Robot analysis and control. John Wiley & Sons, New York.
- Ball RS (1986) A treatise on the theory of screws. Cambridge University Press, Cambridge.
- Chen SF, Kao I (1999) The conservative congruence transformation of stiffness control in robotic grasping and manipulation. In Proceedings of the 9th International Symposium of Robotics Research, pp 7–14, Snowbird.
- Chen SF, Kao I (2000) Conservative congruence transformation for joint and cartesian stiffness matrices of robotic hands and fingers. International Journal of Robotics Research, 19: 835–847.
- Hunt KH (1978) Kinematic geometry of mechanisms. Clarendon Press, Oxford.
- Kao I, Ngo C (1999) Properties of grasp stiffness matrix and conservative control strategy. International Journal of Robotics Research, 18:159–167.
- Mckerrow PJ (1991) Introduction to robotics. Addison-Wesley, New York.
- Roth B (1984) Screws, motors, and wrenches that cannot be bought in a hardware store. In Proceedings of the 1st International Symposium of Robotics Research, pp 679–693.
- Salisbury JK (1980) Active stiffness control of a manipulator in cartesian coordinates. In Proceedings of the 19th IEEE Conference on Decision and Control, pp 87–97, Albuquerque.
- Salisbury JK (1982) Kinematic and force analysis of articulated hands. PhD thesis, Stanford University.
- Schilling RJ (1990) Fundamentals of robotics. Prentice-Hall, Upper Saddle River, New Jersey.
- Tsai LW (1999) Robot analysis: the mechanics of serial and parallel manipulators. John Wiley & Sons, New York.
- Waldron KJ, Wang SL, Bolin SJ (1985) A study of the jacobian matrix of serial manipulators. ASME Journal of Mechanisms, Transmissions, and Automation in Design, 107:230–238.

A Dynamic Model of Contact Between a Robot and an Environment with Unknown Dynamics

Roy Featherstone

Dept. Systems Engineering,
Australian National University,
Canberra, Australia

Abstract. This paper presents an analysis of frictionless contact between a rigid body belonging to a robot mechanism and one belonging to its environment. According to this analysis, it is possible to design a hybrid motion/force controller such that the motion and force subsystems are instantaneously independent of each other, and both are instantaneously independent of the environmental dynamics. A control system with this property should be able to operate in contact with an environment having unknown dynamics.

1 Introduction

This paper presents a mathematical analysis of the dynamics of a robot mechanism in which a single body is making frictionless contact with a body in its environment. The latter is assumed to be part of an arbitrary rigid-body system with unknown dynamics. The contact is modelled as a general holonomic constraint between the two bodies; and it is assumed that the degree of constraint is constant at the current instant.

It is also assumed that the contact constraints are known, and that the positions and velocities of the two participating bodies are known and are consistent with the contact constraints.

In addition to the above assumptions, the method used in this paper requires that the participating robot body have six degrees of motion freedom (DMF). Apart from this, the method is completely general and is applicable to any kind of robot, including serial and parallel robots arms, mobile robots and robots with non-rigid mechanisms.

The analysis employs articulated-body equations to model the dynamics of the two participating bodies, and the change-of-basis technique (Featherstone, Fijany, 1999) to resolve the equations into two independent subsystems. The analysis is conducted in a dual system of vector spaces, which assures the invariance and dimensional consistency of the resulting equations (Doty, Melchiorri, Bonivento, 1993; Selig, McAree, 1996).

The outcome is a pair of equations describing the behaviour of the robot body: one in the subspace of motion freedoms, and one in the subspace of contact forces. At the current instant, these equations are decoupled from each other, and the motion equation is independent of environmental dynamics. Furthermore, although the

actual contact force does depend on the environment, the sum of this force with that required to maintain contact with the moving environment does not.

It is therefore possible to design a hybrid motion/force controller such that the motion and force subsystems are instantaneously decoupled, and the environment has no instantaneous effect on the controlled behaviour of either subsystem. There will be non-instantaneous effects, of course, but these are all felt at the position and velocity levels, not the acceleration level, so it may be possible to treat them as slowly-varying disturbances for the control system to reject. Such a controller should be able to operate in contact with an environment having unknown dynamics.

The remainder of this paper is organized as follows. First, a little background material is presented in order to put this paper into context. This is followed by an introduction to the mathematics of systems of dual vector spaces, which forms the mathematical basis for all that follows. The remaining sections describe the contact model, the analysis and a possible control strategy.

2 Background

The main contribution of this paper concerns the degree to which the motion and force control subsystems of a hybrid motion/force controller can be decoupled when the robot is in contact with an unknown dynamic environment. The problems of dynamics compensation and decoupling have already received much attention; for example, see (Cai, Goldenberg, 1989; De Schutter, Torfs, Bruyninckx et al., 1997; Faessler, 1990; Jankowski, ElMaraghy, 1992; Khatib, 1987; McClamroch, Wang, 1988; Mills, Goldenberg, 1989; Stramigioli, 1998; Yoshikawa, Zheng, 1993). In particular, it is already known that a hybrid motion/force controller with full compensation for the robot mechanism's dynamics will exhibit decoupled behaviour when the robot is in contact with a rigid environment (Yoshikawa, 1987); and it is known that decoupling can also be achieved when the robot is in contact with a general dynamic environment, provided the controller has at least partial knowledge of the environmental dynamics (De Luca, Manes, 1994; De Schutter, Bruyninckx, 1996). This paper shows that the decoupling property extends to the case of contact with an unknown dynamic environment.

The subject matter of this paper is probably closest to that of (De Luca, Manes, 1994), since both papers tackle essentially the same problem. The differences lie in the analytical method, the results obtained, and the proposed control scheme. This paper uses articulated-body models for both the robot and the environment. The former is closely related to the operational-space approach (Khatib, 1987, 1995); and the latter is functionally equivalent to the second-order model in (De Luca, Manes, 1994) for instantaneous dynamics.

The analysis is carried out using the change-of-basis technique described in (Featherstone, Fijany, 1999). The nearest similar idea is the modal decoupling of (De Schutter et al., 1997), which involves an eigenvalue-based decomposition of the product of an inverse mass matrix and a stiffness matrix. The change-of-basis technique uses a different kind of decomposition that reduces mass matrices and

their inverses to block-diagonal form. Mathematically, the distinction is that modal decoupling works on a matrix that transforms vectors within a single vector space, whereas the change-of-basis decomposition applies to matrices that map vectors from one space to another.

Finally, this paper uses the concept of duality (or reciprocity) between motion and force vectors, in order to obtain an invariant formulation. It follows the formal approach advocated in (Selig, McAree, 1996), in which these vectors are assigned to two distinct vector spaces, each being the dual of the other. Many papers already use notions of duality and reciprocity, but often in a less explicit or less formal manner than here. (For example, generalized velocities and forces form a dual system.) More information on this topic can be found in (De Schutter, Bruyninckx, 1992; Doty et al., 1993; Duffy, 1990; Lipkin, Duffy, 1988; Stramigioli, 1998).

3 Dual Vector Spaces

A system of dual vector spaces, or ‘dual system’ for short, is a mathematical structure comprising two vector spaces of equal dimension and a scalar product (a nondegenerate bilinear form) that takes one argument from each space. Each vector space is considered to be the dual of the other.

A dual system can be defined by listing its constituent parts, so we shall use the expression $\langle U, V, \cdot \rangle$ to denote the dual system consisting of the vector spaces U and V and the scalar product ‘ \cdot ’. If $u \in U$ and $v \in V$ then the scalar product may be written either $u \cdot v$ or $v \cdot u$, both meaning the same, but the expressions $u \cdot u$ and $v \cdot v$ are not defined.

Dual systems arise naturally in the mathematical modelling of physical systems in which a scalar physical quantity equates with the scalar product of two different types of vector. In the case of rigid-body systems, the scalar is work (or power, virtual work, etc.) and the two types of vector are motions and forces. We therefore define a dual system $\langle M^n, F^n, \cdot \rangle$ in which M^n is a space of n -dimensional motion vectors, F^n is a space of n -dimensional force vectors, and the scalar product is the work done by a force vector acting on a motion vector. Examples of motion vectors include (generalized) velocities, accelerations, infinitesimal displacements and directions of motion freedom; and examples of force vectors include (generalized) forces, momenta and contact normals.

The same basic structure appears in bond graphs, where effort and flow vectors combine to form energy scalars, and also in tensor calculus, where covariant and contravariant vectors combine to form invariant scalars. In hybrid motion/force control, dual systems support invariant formulations (Selig, McAree, 1996; Stramigioli, 1998).

3.1 Reciprocal Coordinates

A basis on a dual system $\langle U, V, \cdot \rangle$ is a set of vectors, half of which form a basis on U , while the other half form a basis on V .

Let P be a basis on $\langle \mathbf{M}^n, \mathbf{F}^n, \cdot \rangle$, and let $P = \{P_{\mathbf{M}}, P_{\mathbf{F}}\}$ where $P_{\mathbf{M}} = \{\mathbf{d}_1, \dots, \mathbf{d}_n\} \subset \mathbf{M}^n$ is a basis on \mathbf{M}^n and $P_{\mathbf{F}} = \{\mathbf{e}_1, \dots, \mathbf{e}_n\} \subset \mathbf{F}^n$ is a basis on \mathbf{F}^n . In P coordinates, motion vectors are expressed using $P_{\mathbf{M}}$ and force vectors using $P_{\mathbf{F}}$.

If the elements of $P_{\mathbf{M}}$ and $P_{\mathbf{F}}$ satisfy

$$\mathbf{d}_i \cdot \mathbf{e}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

then $P_{\mathbf{M}}$ is said to be reciprocal to $P_{\mathbf{F}}$, and P defines a reciprocal coordinate system on $\langle \mathbf{M}^n, \mathbf{F}^n, \cdot \rangle$. (The term ‘dual coordinates’ is also used.) The special property of reciprocal coordinates is that the scalar product takes the form $\mathbf{m} \cdot \mathbf{f} = \mathbf{m}^T \mathbf{f}$ and is invariant with respect to any change of basis from one reciprocal coordinate system to another.

Reciprocal coordinates are the dual-system equivalent of orthonormal (or Cartesian) coordinates in a Euclidean vector space, but there is one important difference: there are n^2 freedoms to choose a reciprocal basis on a dual system, compared with only $n(n - 1)/2$ freedoms to choose an orthonormal basis on a Euclidean n -space. These extra freedoms encompass generalized coordinates, and they provide the change-of-basis technique with enough freedoms to work.

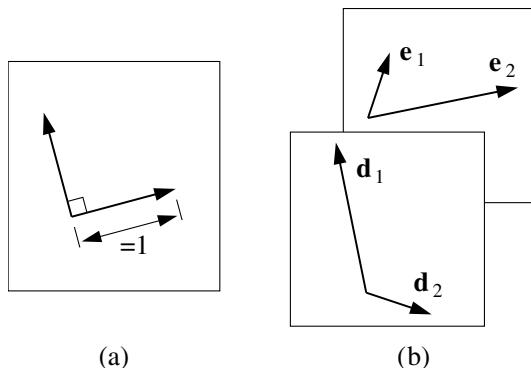


Fig. 1. The difference between an orthonormal basis on a 2D Euclidean space (a) and a reciprocal basis on a 2D system of dual vector spaces (b)

The difference is illustrated in Fig. 1. The orthonormal basis in Fig. 1(a) consists of two unit vectors at right angles. Since neither the lengths of the vectors nor the angle between them can be altered, the only parameter that can be varied freely is the overall orientation of the basis. In contrast, the reciprocal basis shown in Fig. 1(b) consists of four vectors in total, two in each space. In general, one may choose any two of these vectors freely, leaving the other two to be determined by the reciprocity conditions; so there are a total of four freedoms to choose a reciprocal basis on this dual system.

Although Fig. 1(b) uses the visual cues of angle and magnitude to suggest the reciprocity conditions (for example, \mathbf{d}_1 is shown at right angles to \mathbf{e}_2 to suggest $\mathbf{d}_1 \cdot \mathbf{e}_2 = 0$), it should be realized that the concepts of angle and magnitude are usually undefined in a dual system.

3.2 Coordinate Transforms

In general, motion and force vectors obey different coordinate transformation rules. If P and Q are two reciprocal bases on $(\mathbf{M}^n, \mathbf{F}^n, \cdot)$, and $\mathbf{m}_P, \mathbf{m}_Q, f_P$ and f_Q are coordinate vectors representing $\mathbf{m} \in \mathbf{M}^n$ and $f \in \mathbf{F}^n$ in P and Q coordinates, respectively, then

$$\mathbf{m}_Q = X_M \mathbf{m}_P, \quad f_Q = X_F f_P,$$

$$\mathbf{m}_P = X_M^{-1} \mathbf{m}_Q, \quad f_P = X_F^{-1} f_Q,$$

where X_M and X_F are the coordinate transformation matrices that perform the transformations from P_M to Q_M coordinates, and P_F to Q_F coordinates, respectively. The reciprocity property ensures that

$$X_F = (X_M^{-1})^T,$$

which is sufficient to guarantee the invariance of the scalar product.

Table 1. Transformation rules for linear mappings defined on a dual system

mapping	transform rule	transform type
$\mathbf{M}^n \mapsto \mathbf{M}^n$	$A_Q = X_M A_P X_M^{-1}$	similarity
$\mathbf{M}^n \mapsto \mathbf{F}^n$	$B_Q = X_F B_P X_M^{-1}$	congruence
$\mathbf{F}^n \mapsto \mathbf{M}^n$	$C_Q = X_M C_P X_F^{-1}$	congruence
$\mathbf{F}^n \mapsto \mathbf{F}^n$	$D_Q = X_F D_P X_F^{-1}$	similarity

The two vector spaces of a dual system give rise to four types of linear mapping: $\mathbf{M}^n \mapsto \mathbf{M}^n$, $\mathbf{M}^n \mapsto \mathbf{F}^n$, $\mathbf{F}^n \mapsto \mathbf{M}^n$ and $\mathbf{F}^n \mapsto \mathbf{F}^n$. They can all be represented using $n \times n$ matrices, but each has its own transformation rule, as shown in Table 1. (They are essentially the same as the rules for transforming the four types of dyadic tensor.) Observe that two mappings obey similarity transforms, which preserve eigenvalues, while the other two obey congruence transforms, which preserve symmetry and definiteness.

3.3 Subspaces

Any subspace of a vector space can be expressed as the range of a suitable matrix. If S is an m -dimensional subspace of an n -dimensional vector space then it can be

expressed as the range of an $n \times m$ matrix S whose columns are the coordinates of any m linearly-independent vectors that span S . These vectors form a basis on S , so S serves to define both a subspace and a basis. The matrix transforms like its column vectors. Any element of S can be expressed in the form $S\alpha$, where α is an $m \times 1$ vector of coordinates.

Subspaces can be used to define linear decompositions of the parent space. If a vector space V is the direct sum of two subspaces S_1 and S_2 (written $V = S_1 \oplus S_2$) then any vector $v \in V$ can be decomposed uniquely into $v = v_1 + v_2$ where $v_1 \in S_1$ and $v_2 \in S_2$. Any two subspaces will direct-sum to V provided they have no nonzero element in common and their dimensions sum to the dimensions of V .

The decomposition can also be written in the form

$$v = S_1 \alpha_1 + S_2 \alpha_2 = [S_1 \ S_2] \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}.$$

In this equation, $[S_1 \ S_2]$ is a square matrix that can be interpreted as a coordinate transform, and $[\alpha_1^T \ \alpha_2^T]^T$ is a representation of v in the coordinate system defined by the columns of $[S_1 \ S_2]$.

If two subspaces $S \subset M^n$ and $T \subset F^n$ have the property that $s \cdot t = 0$ for every $s \in S$ and $t \in T$ then we say that they are reciprocal, and write $S \perp T$. If, in addition, they satisfy $\dim(S) + \dim(T) = n$ (i.e., their dimensions sum to n) then we say that T is the *reciprocal complement* of S , and write $T = S^\perp$. Reciprocal complements are unique, and satisfy $(S^\perp)^\perp = S$.

Notice that the word ‘reciprocal’ has a different meaning in ‘reciprocal complement’ to that in ‘reciprocal basis’. In the former, it refers to the screw-theoretic definition of reciprocity between twists and wrenches; while in the latter, it refers to the fact that the product of the two sets of basis vectors is the identity matrix. In multilinear algebra, S^\perp would be called the annihilator of S .

3.4 M^6 and F^6

The dual system (M^6, F^6, \cdot) is ideally suited to describing rigid-body dynamics; and most existing 6D vector notations can easily be translated into a dual-system format so that existing formulas and equations can be re-used. In general, the translation process involves the following steps:

1. Formally assign each vector quantity to the appropriate vector space (M^6 or F^6), and classify mappings and dyadics as per Table 1.
2. Adopt a reciprocal coordinate system (see below).
3. Convert conventional accelerations to spatial accelerations (Featherstone, 1987), so that they behave like vectors.
4. Abandon certain concepts that are incompatible with duality (e.g. inner products, orthogonal complements and the common-screw relation between twists and wrenches).

Most 6-D vector notations use Plücker ray and/or axis coordinates (see (Lipkin, Duffy, 1988)). If ray coordinates are used for M^6 and axis coordinates for F^6 , or vice versa, then the coordinates are reciprocal.

4 Dynamic Model of Contact

This section presents a dynamic model of a general state of contact between a robot body B^r and an environment body B^e . The model consists of an equation of motion for each participating body (in the absence of contact) and a description of the contact constraint. The equations are:

$$\mathbf{a}^r = \Phi^r (\mathbf{f}^r - \mathbf{f}^c) + \mathbf{b}^r, \quad (1)$$

$$\mathbf{a}^e = \Phi^e \mathbf{f}^c + \mathbf{b}^e, \quad (2)$$

$$\mathbf{a}^r - \mathbf{a}^e = S^c \dot{\alpha} + \dot{S}^c \alpha \quad (3)$$

and

$$(S^c)^T \mathbf{f}^c = \mathbf{0}. \quad (4)$$

(See Fig. 2.)

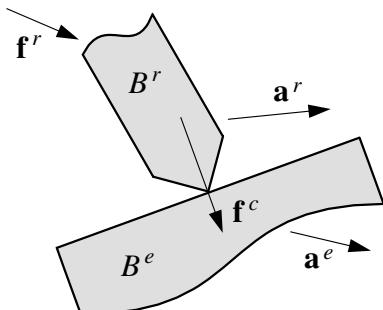


Fig. 2. Contact between robot body B^r and environment body B^e

Equations (1) and (2) express the dynamic behaviour of B^r and B^e in the form of articulated-body equations of motion (Featherstone, 1987). This type of equation allows each body to be a member of an arbitrary rigid-body system of unlimited size and complexity. Neither equation describes the dynamics of an entire rigid-body system, but each describes the totality of dynamic effects that are felt at the relevant body. They are therefore fully general for the task at hand.

\mathbf{a}^r and \mathbf{a}^e are the spatial accelerations of B^r and B^e , respectively; Φ^r and Φ^e are their articulated-body inverse inertias; and \mathbf{b}^r and \mathbf{b}^e are their bias accelerations. \mathbf{b}^r is defined as the acceleration that B^r would have if $\mathbf{f}^r - \mathbf{f}^c = \mathbf{0}$, and therefore accounts for the sum of all forces acting on B^r other than $\mathbf{f}^r - \mathbf{f}^c$; and \mathbf{b}^e is defined similarly. Φ^r and Φ^e are symmetric positive semidefinite matrices with ranks equal to the DMF of their respective bodies. At one extreme, if B^e had no freedom to move then the rank of Φ^e would be zero (which implies that $\Phi^e = \mathbf{0}$). At the other extreme, if B^r had a full 6 DMF then Φ^r would have full rank and hence be a symmetric positive definite (SPD) matrix.

\mathbf{f}^r is a force transmitted to B^r from other parts of the robot mechanism, and \mathbf{f}^c is the force transmitted from B^r to B^e through the contact. \mathbf{f}^r is assumed to contain all controllable forces acting on B^r , but its exact definition can be chosen to suit individual circumstances. Any force that is not included in \mathbf{f}^r will have its effects incorporated into \mathbf{b}^r instead.

$S^c \subset M^6$ is the subspace of instantaneous motion freedoms permitted by the contact constraints at the current configuration, and S^c is a matrix representing S^c . If the contact imposes d constraints on the relative motion of B^r and B^e then S^c has dimension $6 - d$ and S^c is a $6 \times (6 - d)$ matrix.

Equation (3) expresses the acceleration constraint imposed by the contact, which is simply the time-derivative of the velocity constraint equation: $\mathbf{v}^r - \mathbf{v}^e = S\boldsymbol{\alpha}$. At the acceleration level, all velocities are assumed to be known; so $\dot{\boldsymbol{\alpha}}$ is treated as a vector of unknown acceleration variables, while $\boldsymbol{\alpha}$ is treated as known. \dot{S}^c is also assumed to be known.

Finally, (4) expresses the fact that \mathbf{f}^c is a nonworking constraint force: it does no work in any direction of motion allowed by the motion constraints.

Equation (1) is capable of modelling the dynamics of any body in a general rigid-body system. It is therefore applicable to practically any robot, including mobile robots, parallel robots and so on. A couple of examples are shown in Fig. 3. Both are serial robot arms, and in both cases B^r is the end effector.

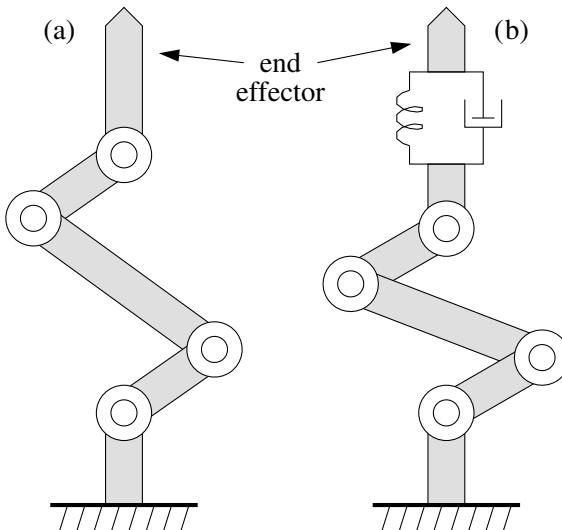


Fig. 3. A rigid robot mechanism (a) and a robot with a compliance between the end effector and the arm (b)

Figure 3(a) shows a robot with a rigid mechanism – one composed entirely of rigid bodies and ideal joints. In this example, (1) can be equated with the operational-space formulation of end-effector dynamics, such as (14) or (50) in (Khatib,

1987) or (3), (9) or (23) in (Khatib, 1995). If we take

$$\Lambda \ddot{\mathbf{x}} + \mu + \mathbf{p} = \mathbf{F}$$

to be a representative operational-space formulation, where Λ is the operational-space inertia of the end-effector and μ and \mathbf{p} contain velocity-product and gravitational terms, respectively, then $\Phi^r = \Lambda^{-1}$ and $\mathbf{b}^r = -\Lambda^{-1}(\mu + \mathbf{p})$.

The robot in Fig. 3(b) consists of an end-effector body connected to the end of a rigid robot arm via a generalized spring and damper. A system like this could be used to model a robot with a wrist-mounted 6-axis force sensor. The end effector in this example is kinematically independent of the arm, so (1) should be the equation of motion of just this one body. Φ^r should be the inverse of the end-effector's rigid-body inertia; \mathbf{f}^r should be the force transmitted to the end-effector through the spring and damper; and \mathbf{b}^r should account for gravity and velocity-product terms. Bear in mind that \mathbf{a}^r refers to the acceleration of the end-effector, not the end of the arm.

5 Analysis

This section derives an equation of motion for B^r , including the effect of the contact, using the change-of-basis technique described in (Featherstone, Fijany, 1999). This method requires Φ^r to be an SPD matrix, so it is applicable only to systems in which B^r has 6 DMF in the absence of the contact constraint. If B^r has fewer DMF then a different analytical procedure must be used, which is outside the scope of this paper.

The first step is to define four subspaces, S_1 , S_2 , T_1 and T_2 , that satisfy the following equations:

$$\begin{aligned} S_1 \oplus S_2 &= \mathbf{M}^6, & T_1 \oplus T_2 &= \mathbf{F}^6, \\ S_1 &= S^c, & T_1 &= (\Phi^r)^{-1} S_1, \\ S_2 &= \Phi^r T_2, & T_2 &= (S^c)^\perp. \end{aligned}$$

These spaces are defined uniquely by S^c and Φ^r , and they have the effect of decomposing $(\mathbf{M}^6, \mathbf{F}^6, \cdot)$ into a pair of dual subsystems, $\langle S_1, T_1, \cdot \rangle$ and $\langle S_2, T_2, \cdot \rangle$, which are aligned with the directions of freedom and constraint as specified by S^c and $(S^c)^\perp$.

The second step is to define matrices S_1 , S_2 , T_1 and T_2 to represent the above four subspaces and simultaneously satisfy

$$[S_1 \ S_2]^T [T_1 \ T_2] = \mathbf{I}_{6 \times 6}.$$

This condition ensures that the column vectors of the four matrices form a reciprocal basis on $(\mathbf{M}^6, \mathbf{F}^6, \cdot)$. In this special basis, the coordinates fall naturally into two groups: one associated with $\langle S_1, T_1, \cdot \rangle$ and one with $\langle S_2, T_2, \cdot \rangle$. Furthermore, Φ^r takes block-diagonal form, with one block in $\langle S_1, T_1, \cdot \rangle$ and one in $\langle S_2, T_2, \cdot \rangle$. If S_1 and T_2 are any two matrix representations of S_1 and T_2 then the reciprocity condition can be satisfied by choosing

$$T_1 = (\Phi^r)^{-1} S_1 (S_1^T (\Phi^r)^{-1} S_1)^{-1}$$

and

$$\mathbf{S}_2 = \Phi^r \mathbf{T}_2 (\mathbf{T}_2^T \Phi^r \mathbf{T}_2)^{-1}.$$

The next step is to transform (1–4) to the special basis. If we define \mathbf{X}_M and \mathbf{X}_F to be the coordinate transformation matrices for motion and force vectors, respectively, from the given basis to the special basis, then

$$\mathbf{X}_M = [\mathbf{T}_1 \mathbf{T}_2]^T, \quad \mathbf{X}_M^{-1} = [\mathbf{S}_1 \mathbf{S}_2],$$

$$\mathbf{X}_F = [\mathbf{S}_1 \mathbf{S}_2]^T, \quad \mathbf{X}_F^{-1} = [\mathbf{T}_1 \mathbf{T}_2].$$

If $[\mathbf{a}_1^r \mathbf{a}_2^r]^T$ is the coordinate vector representing \mathbf{a}^r in the special basis then

$$\begin{bmatrix} \mathbf{a}_1^r \\ \mathbf{a}_2^r \end{bmatrix} = \mathbf{X}_M \mathbf{a}^r = \begin{bmatrix} \mathbf{T}_1^T \mathbf{a}^r \\ \mathbf{T}_2^T \mathbf{a}^r \end{bmatrix}$$

and

$$\mathbf{a}^r = \mathbf{X}_M^{-1} \begin{bmatrix} \mathbf{a}_1^r \\ \mathbf{a}_2^r \end{bmatrix} = \mathbf{S}_1 \mathbf{a}_1^r + \mathbf{S}_2 \mathbf{a}_2^r.$$

All other motion vectors transform similarly; and force vectors obey similar equations with \mathbf{X}_F in place of \mathbf{X}_M .

Transforming (1) and (2) to the special basis produces

$$\begin{bmatrix} \mathbf{a}_1^r \\ \mathbf{a}_2^r \end{bmatrix} = \begin{bmatrix} \Phi_{11}^r & \mathbf{0} \\ \mathbf{0} & \Phi_{22}^r \end{bmatrix} \begin{bmatrix} f_1^r - f_1^c \\ f_2^r - f_2^c \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1^r \\ \mathbf{b}_2^r \end{bmatrix} \quad (5)$$

and

$$\begin{bmatrix} \mathbf{a}_1^e \\ \mathbf{a}_2^e \end{bmatrix} = \begin{bmatrix} \Phi_{11}^e & \Phi_{12}^e \\ \Phi_{21}^e & \Phi_{22}^e \end{bmatrix} \begin{bmatrix} f_1^c \\ f_2^c \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1^e \\ \mathbf{b}_2^e \end{bmatrix}, \quad (6)$$

where $\Phi_{ii}^r = \mathbf{T}_i^T \Phi^r \mathbf{T}_i$ and $\Phi_{ij}^e = \mathbf{T}_i^T \Phi^e \mathbf{T}_j$, $i, j \in \{1, 2\}$, from the appropriate formula in Table 1.

Transforming (3) to the special basis produces

$$\begin{bmatrix} \mathbf{a}_1^r - \mathbf{a}_1^e \\ \mathbf{a}_2^r - \mathbf{a}_2^e \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \dot{\alpha} + \begin{bmatrix} \mathbf{b}_1^c \\ \mathbf{b}_2^c \end{bmatrix}$$

where $\mathbf{b}^c = \dot{\mathbf{S}}^c \boldsymbol{\alpha}$. (Note the special form of \mathbf{S}^c in the special basis.) As $\dot{\boldsymbol{\alpha}}$ is a free variable, there is actually no constraint on the value of $\mathbf{a}_1^r - \mathbf{a}_2^e$; so the constraint equation can be simplified to

$$\mathbf{a}_2^r - \mathbf{a}_2^e = \mathbf{b}_2^c. \quad (7)$$

Finally, transforming (4) to the special basis, and simplifying the result, produces

$$f_1^c = \mathbf{0}. \quad (8)$$

Now that the equations are all expressed in the special basis, all that remains is to solve them. Substituting (8) into (5) and (6) produces

$$\mathbf{a}_1^r = \Phi_{11}^r f_1^r + \mathbf{b}_1^r, \quad (9)$$

$$\mathbf{a}_2^r = \Phi_{22}^r (f_2^r - f_2^c) + \mathbf{b}_2^r \quad (10)$$

and

$$\mathbf{a}_2^e = \Phi_{22}^e f_2^c + \mathbf{b}_2^e. \quad (11)$$

(We are not interested in \mathbf{a}_1^e .) Substituting (10) and (11) into (7) produces

$$\Phi_{22}^r (f_2^r - f_2^c) + \mathbf{b}_2^r - \Phi_{22}^e f_2^c - \mathbf{b}_2^e = \mathbf{b}_2^c,$$

from which we obtain

$$f_2^c = (\Phi_{22}^r + \Phi_{22}^e)^{-1} (\Phi_{22}^r f_2^r + \mathbf{b}_2^r - \mathbf{b}_2^e - \mathbf{b}_2^c). \quad (12)$$

Equations (9), (10) and (12) between them describe the dynamic behaviour of B^r , taking into account the effect of the contact. Equation (9) describes the behaviour of B^r in $\langle S_1, T_1, \cdot \rangle$, while (10) (with f_2^c given by (12)) describes its behaviour in $\langle S_2, T_2, \cdot \rangle$.

The most striking feature of these equations is that (9) is independent of environmental dynamics. This means that the behaviour of B^r in $\langle S_1, T_1, \cdot \rangle$ is instantaneously independent of environmental dynamics, although the environment will, of course, have an integral effect that is evident over time. Another useful feature is that (9) and (10) are decoupled from each other, which means that the behaviour of B^r in $\langle S_1, T_1, \cdot \rangle$ is instantaneously independent of its behaviour in $\langle S_2, T_2, \cdot \rangle$; and a third interesting feature is that of all the quantities appearing in (6), only Φ_{22}^e and \mathbf{b}_2^e have any instantaneous effect on B^r .

To summarize, the equation of motion of B^r can be expressed as the sum of two instantaneously independent subsystems: one in $\langle S_1, T_1, \cdot \rangle$, which is aligned with the motion freedoms of the contact constraint, and one in $\langle S_2, T_2, \cdot \rangle$, which is aligned with the directions of constraint. The former is instantaneously independent of environmental dynamics, while the latter depends on only a subset of the dynamics of B^e . The only assumption needed to achieve these results is that Φ^r is an SPD matrix.

6 Control

This section applies the results of the previous section to the analysis of a control law for a hybrid motion/force controller. It is shown that the motion and force subsystems are instantaneously decoupled, and that the former is decoupled from the environmental dynamics. A modification to the force subsystem is suggested that decouples it also from the environmental dynamics.

Consider the following control law:

$$\mathbf{f}^r = (\Phi^r)^{-1} (\mathbf{S}_1 \mathbf{u}_m - \mathbf{b}^r) + \mathbf{T}_2 \mathbf{u}_f, \quad (13)$$

where \mathbf{u}_m and \mathbf{u}_f are vectors computed by the motion- and force-control subsystems, respectively. Observe that it does not require any knowledge of the environment's dynamics. Transforming this equation to the special basis produces

$$\begin{bmatrix} \mathbf{f}_1^r \\ \mathbf{f}_2^r \end{bmatrix} = \begin{bmatrix} (\Phi_{11}^r)^{-1} (\mathbf{u}_m - \mathbf{b}_1^r) \\ \mathbf{u}_f - (\Phi_{22}^r)^{-1} \mathbf{b}_2^r \end{bmatrix};$$

and substituting the expressions for \mathbf{f}_1^r and \mathbf{f}_2^r from this equation into (9) and (10) produces

$$\mathbf{a}_1^r = \mathbf{u}_m \quad (14)$$

and

$$\mathbf{f}_2^c + (\Phi_{22}^r)^{-1} \mathbf{a}_2^r = \mathbf{u}_f. \quad (15)$$

These are the equations of motion for B^r under closed-loop control via (13). Equation (14) describes the behaviour of B^r in $\langle S_1, T_1, \cdot \rangle$ as a function of the motion control signal \mathbf{u}_m ; and (15) describes the behaviour of B^r in $\langle S_2, T_2, \cdot \rangle$ as a function of the force control signal \mathbf{u}_f .

These equations describe a pair of decoupled subsystems, in the sense that \mathbf{u}_m has no instantaneous effect in $\langle S_2, T_2, \cdot \rangle$ and \mathbf{u}_f has no instantaneous effect in $\langle S_1, T_1, \cdot \rangle$. Therefore a hybrid motion/force controller that uses (13) to combine the outputs of the force and motion control laws exhibits no instantaneous crosstalk between the motion and force control channels. There will, of course, still be some degree of cross-coupling between the two subsystems; but these effects are carried via position- and velocity-dependent terms, so it may be feasible to treat them as slowly-varying disturbances for the control system to reject.

Observe that neither (14) nor (15) contains any instantaneous dependency on the environment. We have already established this property for (9), so it is not surprising if it is inherited by (14); but (15) demonstrates that the environmental dependency of \mathbf{f}_2^c exactly cancels that of $(\Phi_{22}^r)^{-1} \mathbf{a}_2^r$, so that the expression $\mathbf{f}_2^c + (\Phi_{22}^r)^{-1} \mathbf{a}_2^r$ is independent of environmental dynamics. This quantity can be thought of as the sum of the contact force and the force required to accelerate the robot so as to maintain contact with the environment.

If the force control subsystem is designed so that the objective is to control the value of $\mathbf{f}_2^c + (\Phi_{22}^r)^{-1} \mathbf{a}_2^r$ rather than \mathbf{f}_2^c , then the controlled behaviour in both the motion and force subsystems will be independent of environmental dynamics. The environment will, of course, still have an effect on the robot's behaviour, but it does so via position- and velocity-dependent terms which the control system could treat as slowly-varying disturbances. A control system that seeks to control $\mathbf{f}_2^c + (\Phi_{22}^r)^{-1} \mathbf{a}_2^r$ in the first instance, still has the option of controlling \mathbf{f}_2^c via an outer loop operating at a lower frequency.

Finally, note that (13) is not a new control law, so the results in this section apply also to any existing control scheme that happens to use the same control law, or an equivalent one.

7 Conclusion

This paper has presented an analysis of frictionless rigid-body contact between a general robot and a general dynamic environment, which assumes only that the participating robot body has six degrees of motion freedom. The analysis uses articulated-body equations to describe the dynamics of the participating bodies, and the change-of-basis technique to resolve the equations into independent subsystems.

It was shown that the equation of motion of the participating robot body can be resolved into two subsystems, one in the space of motion freedoms and one in the space of contact constraints, and that the former is instantaneously independent of environmental dynamics.

A hybrid motion/force controller based on these results is free of instantaneous cross-coupling between the motion and force control channels, and the controlled motion behaviour is instantaneously independent of environmental dynamics. If the force-control subsystem is designed to control the sum of the actual contact force and the force required to accelerate the robot in pursuit of maintaining contact, then the controlled force behaviour is also instantaneously independent of environmental dynamics.

References

- Cai L, Goldenberg AA (1989) An approach to force and position control of robot manipulators. In IEEE Int. Conf. Robotics and Automation, pp 86–91, Scottsdale AZ.
- De Luca A, Manes C (1994) Modelling of robots in contact with a dynamic environments. *IEEE Trans. Robotics and Automation*, 10(4):542–548.
- De Schutter J, Bruyninckx H (1992) Model-based specification and execution of compliant motion. In IEEE Int. Conf. Robotics and Automation, Nice France.
- De Schutter J, Bruyninckx H (1996) Force control of robot manipulators. In The Control Handbook, pp 1351–1358. CRC Press, Boca Raton Florida.
- De Schutter J, Torfs D, Bruyninckx H et al. (1997) Invariant hybrid force/- position control of a velocity controlled robot with compliant end effector using modal decoupling. *Int. J. Robotics Research*, 16(3):340–356.
- Doty KL, Melchiorri C, Bonivento C (1993) A theory of generalized inverses applied to robotics. *Int. J. Robotics Research*, 12(1):1–19.
- Duffy J (1990) The fallacy of modern hybrid control theory that is based on orthogonal complements' of twist and wrench spaces. *J. Robotic Systems*, pp 139–144.
- Faessler H (1990) Manipulators constrained by stiff contact – dynamics, control and experiments. *Int. J. Robotics Research*, 9(4):40–58.
- Featherstone R (1987) Robot dynamics algorithms. Kluwer Academic Publishers, Boston.
- Featherstone R, Fijany A (1999) A technique for analyzing constrained rigid-body systems and its application to the constraint force algorithm. *IEEE Trans. Robotics & Automation*, 15(6):1140–1144.

- Jankowski KP, ElMaraghy HA (1992) Dynamic decoupling for hybrid control of rigid-/flexible-joint robots interacting with the environment. In *IEEE Trans. Robotics & Automation*, pp 519–534.
- Khatib O (1987) A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. Robotics & Automation*, 3(1):43–53.
- Khatib O (1995) Inertial properties in robotic manipulation: An object-level framework. *Int. J. Robotics Research*, 14(1):19–36.
- Lipkin H, Duffy J (1988) Hybrid twist and wrench control for a robotic manipulator. *ASME J. Mechanisms, Transmissions & Automation in Design*, 110(2):138–144.
- McClamroch NH, Wang D (1988) Feedback stabilization and tracking of constrained robots. *IEEE Trans. Automatic Control*, 33(5):419–426.
- Mills JK, Goldenberg AA (1989) Force and position control of manipulators during constrained motion tasks. *IEEE Trans. Robotics & Automation*, 5(1):30–46.
- Selig JM, McAree PR (1996) A simple approach to invariant hybrid control. In *IEEE Int. Conf. Robotics and Automation*, pp 2238–2245, Minneapolis MN.
- Stramigioli S (1998) From differentiable manifolds to interactive robot control. PhD thesis, T. U. Delft, Netherlands.
- Yoshikawa T (1987) Dynamic hybrid position force control of robot manipulators – description of hand constraints and calculation of joint driving force. *IEEE J. Robotics & Automation*, 3(5):386–392.
- Yoshikawa T, Zheng XZ (1993) Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object. *Int. J. Robotics Research*, 12(3): 219–230.

Part 10

Robot Technology I

Session Summary

John Hollerbach¹ and Jan-Olof Eklundh²

Chairs

¹ Computer Science Department, University of Utah 50 S. Central Campus Drive Salt Lake City, UT 84112-9205, USA jmh@cs.utah.edu

² Department of Numerical Analysis and Computer Science, KTH 100 44 Stockholm, Sweden joe@nada.kth.se

Mobility is a key capability of autonomous robots. There is an extensive knowledge on robots using wheels, tracks, legs and also other modalities for their locomotion (Schempf (1999), Raibert (1999)), some of it dating back to the 19th century. The applications of these forms of locomotion to a large extent depend on what the environment looks like. While wheels are suitable on fairly smooth and hard surfaces, and legs work well on more rough terrain with varying support, tracks and similar systems form an intermediate mode. They give low contact pressure and allow greater variability in surface geometry than wheels, for instance, they cope more easily with stairs.

The first paper in this session, by Hagen Schempf, presents the design of a tracked robot, AURORA, that is small and, in particular, monotread. Existing tracked vehicles have hitherto typically been based on dual-tread systems and steering has been symmetric. Even more general approaches to control of robot locomotion, such as the one presented in Ostrowski, Burdick (1996), have been based on symmetry assumptions. In this paper a small and light weight system is demonstrated, that can deal with fairly rough terrain, such as sand, grass, and rubble, climb stairs using a deployable ramp/paddle, and even recover from turning over. An important design criterion was that the system should contain as many commercially-off-the-shelf components as possible.

The design of a monotread steering system is a key contribution in this work. It uses a laterally-compliant and yet longitudinally rigid flexbelt. Driving and steering of this continuous belt is achieved by simultaneous control of its rotational movement and flexing.

Many important applications can be envisaged for such a system in e.g. reconnaissance, inspection and rescue missions. The demonstrated system was provided with three miniature CMOS cameras and microphones. It is 24 inches long and only weighs 23 lbs. A number of rather impressive demonstration runs showed the potential of this new approach to mobility.

Colon cancer is a terrible disease and the second-leading cause of cancer deaths, but it can be substantially cured by early detection and treatment. Detection methods are advancing rapidly; for example, virtual colonoscopy based on medical imaging such as CT scans looks very promising Imaginis (2001). Invasive methods such as colonoscopy would still be necessary, however, to perform biopsies or excision of polyps. Because of tortuous bends in the colon, manual insertion of the colonoscope can be problematical.

Several robotics researchers have investigated robotic locomotion devices to assist in the navigation of the colon. A key issue is how such robots can achieve traction with the thin walls of the colon, because of the very low friction. A balloon clamp has the potential for significantly overextending the colon, while pneumatic suction causes the collapse of the wall around the robot probe. The second paper, presented by Paolo Dario, proposes mechanical clamping for an inchworm-style robot. Suction pulls a portion of the colon into a groove, which then closes under pneumatic actuation to clamp it. Marks left by the clamps heal in several days and are no more severe than can happen in normal colonoscopy.

Two prototypes of the inchworm robot are presented, both based on pneumatic bellows as the extension mechanism. In the first prototype, there are clamps at either end. Tests showed that sharp turns in the colon were difficult to navigate, due to an accordion effect with the bellows that snags with the colon walls. In the second prototype, concentric clamps are proposed at only one end of the device which can move relative to each other. Since the clamps are located distally, there is always positive motion of the mechanism without the problems of the accordion effect.

Sensing is a prerequisite for any robot activity. Robotics research consequently depends on sensor technology, that however is used also in many other applications and therefore follows a development of its own. Sensors should preferably be fast, accurate, reliable, and, not least, cheap. Off-the shelf components, that are often used in robotics do not always satisfy all these criteria. Standard cameras and range sensors are often not expensive, but may provide output that give unsatisfactory resolution in space and time, that is too noisy, or without allowing desirable control over tuning parameters.

Sonar sensors in principle satisfy all the requirements above, and are also used widely for navigation and mapping. Nevertheless, the systems that are most common often give low quality output. As the author points out in this paper, they are not advanced sonar sensors, in the meaning that they don't determine the angle to the target accurately, and are incapable of classifying the target. In fact, it may not even be possible to infer from their readings if a received echo originates from the same sonar system.

In the paper an advanced system is presented that combines interference rejection with classification in one sensor cycle, thereby allowing the discrimination between target types. The paper demonstrates that improvements on sensor technology may result in qualitatively different sensory data, hence removing the needs for various high-level processing and assumptions. Fusion of this information with output from other sensors forms an interesting new challenge.

Two opposite approaches towards mechanical displays for synthetic environments are haptic interfaces and special-purpose simulators. Haptic interfaces seek to synthesize any mechanical sensation such as the feel of a surface or external objects, but may lack fidelity due to device limitations. Special-purpose simulators such as airplane simulators contain physical mockups of surfaces and devices found in real systems, but while providing a high fidelity mechanical experience they lack the flexibility to present a different synthetic environment. A compromise between

the two has been termed robot graphics Gruenbaum, McNeely, Solwizral (1997) or encounter-type haptic devices Yokokohji, Hollis, Kanade (1999), by placement of objects on robot arms which can arbitrarily position them. When simulating the flipping of a light switch, for example, a real switch can be employed that is positioned by the robot to a location desired in the virtual world. The feel of the switch is highly accurate, but its position is not fixed as in special-purpose simulators which allows reconfigurability.

One problem is the multiplicity of potential objects to be manipulated, which can be partly solved by the use of multiple robot arms or by a turret of objects on the robot's end effector Gruenbaum et al. (1997). Another problem is the visual display of the synthetic environment. The most straightforward approach is to employ a head-mounted display where everything is synthesized, including an avatar representing the user Gruenbaum et al. (1997). Another approach is an augmented reality display, which allows the user's arm to be seen and the object to be manipulated. To block out the environment around the arm and object and synthesize a different environment, a chroma-key technique was proposed in Yokokohji et al. (1999), where a color filter is employed along with draping the environment with a cloth of that color. The video camera that captures the image of the user's arm and object can also track the location of the visual display relative to the robot arm, via fiducial points on the end effector.

The fourth paper in this session, presented by Yasuyoshi Yokokohji, summarizes two advances to the technique in Yokokohji et al. (1999): (1) a hybrid vision/inertial tracking system to remove the lags and improve the accuracy of visual tracking, and (2) a trajectory planning method for the robot arm that avoids the user's reach while repositioning to a new object location. Several systems employing inertial sensing have been proposed before, including the commercial Intersense trackers which are starting to replace magnetic trackers in virtual reality applications. The Intersense trackers employ a miniature inertial sensor package, comprised of three accelerometers and three MEMS-based angular rate sensors Foxlin, Harrington, Pfeifer (1998). To compensate for the drift when these signals are integrated to derive position and orientation, ultrasonic sensors are employed and their output fused with the inertial sensor output. Resolutions less than 2 mm and 0.05 degrees are reported.

Position sensors other than ultrasonic sensors can be employed to provide the drift correction, such as the visual sensing proposed in this paper. Instead of angular rate sensors, the proposed system employs three extra accelerometers on moment arms to detect the three directions of rotation. In its present configuration, the total 6-accelerometer package is not nearly as compact as the Intersense tracker, possibly because angular rate sensors in the Intersense system lead to a more compact design. The proposed system shows considerable improvement over the earlier camera-only tracking system, but it is difficult to compare against Intersense's system because the accuracy is stated in terms of pixel units.

The second part of the paper addresses the transition between object positions when the user wishes to grasp a different object. The robot has to move the object to the position where it is supposed to be grasped, while avoiding collisions with

the user's hand. This is a standard robot path planning problem, where a variety of approaches could be considered such as potential fields especially in consideration of the simple geometry of the problem. A point on a finger is tracked to represent the user's arm, and all virtual objects are represented as points as well. The authors propose an adaptation of the well-known I-COLLIDE system Cohen, Lin, Manocha, Ponamgi (1995): a polyhedron is formed from the object points and a Voronoi diagram is used to detect the closest region to the finger point. The robot end effector shadows the user's hand by moving along the closest point on the polyhedron. Should the user's hand penetrate the polyhedron, the robot end effector is retracted by the radius of a bounding sphere around the finger point.

References

- Cohen J, Lin M, Manocha D, Ponamgi K (1995) I-collide: an interactive and exact collision detection system for large-scaled environments. In Proceedings of ACM Interactive 3D Graphics Conference, pp 189–196.
- Foxlin E, Harrington M, Pfeifer G (1998) Constellation: A wide-range wireless motion-tracking system for augmented reality and virtual set applications. In Computer Graphics (Proc. SIGGRAPH), pp 371–378.
- Gruenbaum PE, McNeely WA, Solwizral HA (1997) Implementation of dynamic robotic graphics for a virtual control panel. *Presence*, 6:118–126.
- Imaginis (2001) Researchers investigate virtual colonoscopy (CT) for detecting colon cancer. URL <http://www.imaginis.com/ct-scan/news/news6.27.01.asp>.
- Ostrowski J, Burdick J (1996) Geometric perspectives on the mechanics, and control of robot locomotion. In The Seventh International Symposium on Robotics Research, pp 536–547, Berlin. Springer Verlag.
- Raibert MH (1999) Legged robots. In Robotics Science, pp 563–594. MIT Press, Cambridge, Mass.
- Schempf H (1999) Mobile robots and walking machines. In Nof SY (ed) Handbook of Industrial Robotics. Wiley & Sons, New York, NY.
- Yokokohji Y, Hollis RL, Kanade T (1999) Wysiwyf display: a visual/haptic interface to virtual environment. *Presence*, 8:412–434.

AURORA – Minimalist Design for Tracked Locomotion

Dr. Hagen Schempf

Automatika, Inc., 137 Delta Drive, Pittsburgh, PA 15238

Abstract. Much work by many researchers and developers (Schempf, 1999) (Menzel, D'Alusio, 2001) (Schraft, Schmierer, 1998) over the last century has developed a vast body of knowledge in the areas of locomotion utilizing wheels, legs, tracks, etc. for applications from research to real-world applications. This paper addresses a novel development of a steerable monotread, dubbed **AURORA** Advanced *Urban RObot for Reconnaissance and Assessment*), proving that a single continuous belt, designed with key flexure and guide elements, is capable of steerable locomotion. This is believed to be a significant departure from the theory that tracked vehicles need to have at least two treads to steer. The system was built with a flexible elastomeric monobelt with a central drive and guide spine, which, when flexed, forces the tread into a shape allowing it to steer. The system is also capable of inverted operations, stair climbing (with the help of a deployable ramp/paddle), and is easily portable due to its small size and low weight. The system is battery-driven and controlled/monitored over a wireless link, allowing it to be deployed safely into hazardous and remote areas in urban terrain. On-board cameras provide multiple side- and bird's-eye views, with on-board computing processing and interpreting imagery. A portable control-box is used for remote control. Preliminary tests have shown the capability of the system to handle rough terrain and steer in all of the environments tested so far. Future work will extend the autonomy capabilities of the system and ruggedize the tread and drive/steer elements even further.

1 Introduction

The use of the most adept locomotion mode(s) for natural/man-made environments represents an essential element of any mobile machine or robot system. Depending on the type of environments to be encountered, different locomotion modes are used to satisfy many requirements such as accessibility, terrain ability, speed and stability, etc. In the case of tracked vehicles, which are typically used in marginal (low contact pressure) environments, designs have typically focused on dual-tread systems, which are steerable in a differential-drive mode resulting in skid-steering locomotion – numerous examples exist in the literature (Crimson, 1992). Recently, commercial and military developments have led to the miniaturization of these systems into more 'suitcase-sized' devices, with the aim of becoming portable/luggable by an individual in such applications as anti-terrorism, EOD and military reconnaissance. Some examples from companies offering these systems for sale (Automatika, 2001; Inuktun, 2001; iRobot, 2001; Remotec, 2001; TeleRob, 2001), are shown in Fig. 1:

with other numerous examples also found in the research literature (Hirose, 1993) (Hirose, 1999) (Menzel, D'Alusio, 2001). The main attribute of all these systems

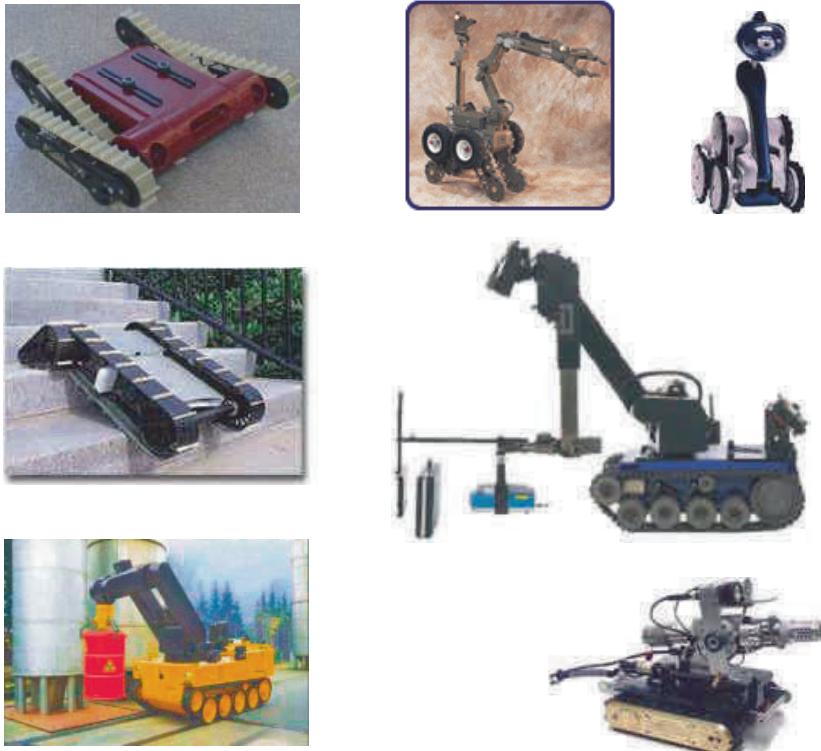


Fig. 1. Smaller-scale tracked/wheeled commercial ‘field-worthy’ locomotors

is their usage of parallel locomotor modules, whether they be wheels, tracks, legs, hybrids, etc. – the notion of symmetry seems paramount in terms of stability and traction/steering. Very few robot designs have ‘violated’ this seeming trend, with hopping and ‘slithering/crawling’ robots being obvious exceptions.

In order to provide for a smaller, lighter and more minimal and thus more portable design, it is the contention of the author that a single steerable tread is a possible answer for rough and marginal terrain.

2 Background

The notion of a single tread being used for locomotion over rugged terrain with low contact- and shear-resistance, is not necessarily a novel idea. Developments dating all the way back to the 1970s, and even before WW II, show that this idea had already been considered. The first evaluations of these types of locomotors were carried out mostly by the US Army, especially for use in arctic and marshy terrains for transporting light cargo and personnel. More recently (last 20 years), inventors have patented a wide variety of single-track steerable slat-based tracks with involved link-

age steering and segmented and interconnected tread sections Over the last 5 years, developers and inventors have taken miniaturization and extreme locomotion capability in single-tracked vehicles to another level, by developing in-line arrangements of these locomotors, and using them to crawl into areas through restricted passages, such as pipes, etc

Another area of industry that uses continuous moving surfaces is the materials-handling sector. They have been using continuous belted-/slatted-/roller-conveyors for decades, and have found particular usage for space-saving fixed-configuration curved conveyors in warehousing and E-commerce distribution and logistics areas. The only design that has been capable of providing for both straight and curved orientations in a single non-interrupted run is that of the slatted-conveyor system. In evaluating all this prior art, one can notice that outdoor steerable slatted- conveyors do not seem reliable enough, yet indoors they seem very feasible and usable. Continuously-belted single treads (dual systems exist) and conveyors have to date not been made steerable to our knowledge, yet they offer major advantages for outdoor rough-terrain locomotion, as do dynamically-steerable indoor slatted conveyor systems. It is proposed herein that AURORA is capable of filling these untapped niches.

3 Performance Requirement

The need for this specification was driven by the desire to rapidly, economically and manually be able to deploy capable remote-controlled robots into hazardous areas, such as disaster zones, earthquake-damaged buildings, terrorist/hostage situations, bomb-threat scenarios, military urban reconnaissance, etc. The original concept that was developed as part of the design-development phase of the program, centered on a device with as many commercially-off-the-shelf (COTS) components as possible, including a steerable conveyor-segment chain steered through linear actuators, and including on-board power, communications and drive actuation. An artist rendering is shown in Fig. 2

In order to refine and evaluate this and other proposed concepts, a set of performance requirements was developed, describing the key attributes that the system should meet (a more appropriate label might be ‘desirements’) (see Table 1).

The performance requirements that were developed for the proposed locomotion system focused on several key areas, namely (I) size and weight, (II) terrain ability, (III) environmental hardening, (IV) mission duration, (V) sensor payloads and (VI) usable teleoperable range. The ones the development chose to emphasize were (I) and (V), without compromising (II) and (III), while maximizing (IV) and (VI). This philosophy then led to the final design and prototype detailed in the next section.

4 System Description

The design developed for the steerable monotread system was driven by the above-described design mantra, and resulted in an extremely compact computer-

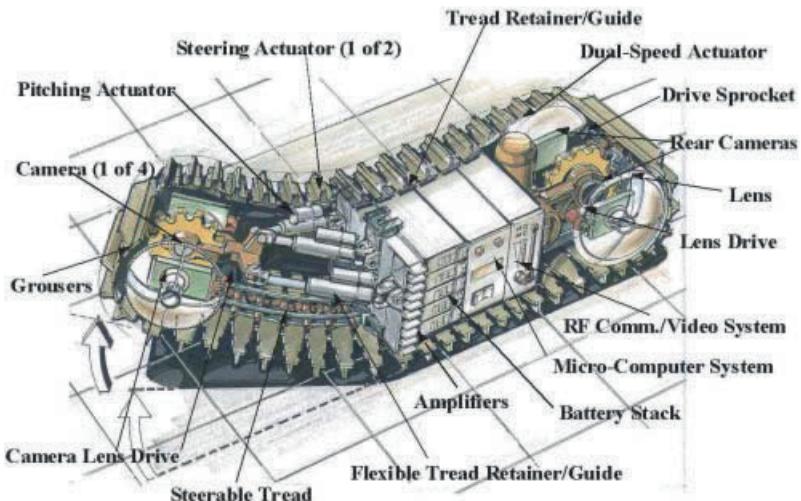


Fig. 2. Artist rendering of steerable monotread

Table 1. Performance requirements

Descriptor	Target	Value
Size	As small as possible	24" L x 6" W x 6" H
Weight	As light AP; man-packable	≤ 20 lbs.
Speed	Flat floor and climbing	< 2 m/sec. flat; 0.5m/sec. @ 60°
Terrain Types	Can handle: sand, grass, rubble	
Device Features	Self-righting, stair-climb	
Turing Radius	Minimize swept arcs	18" radius turn
Environments	Specs of temperature & humidity	-10°C 65°C; 3' UW
Shock	Handling & driving w/o damage	Drop heights: 8' on grass
Mission Duration	Driving at top performance level	1hr. at full speed; 3 miles+
Communications	Usable real-time link range	1-3 miles/300+ yds.out/-indoors
Interface	User control devices	Laptop, monitor & joystick
Sensing	Proprioceptive on-board devices	Color camera(s), microphone(s)

controllable untethered locomotion system. The component-design challenge in this concept lay in the ability to develop a controllable flex structure to shape, guide and retain a new type of laterally-compliant, yet longitudinally-rigid, flex belt. The integration challenge lay in providing for the driving and steering actuation in a compact package capable of meeting the performance requirements as specified in Table 1, while also providing for substantial on-board computing and communications power, and all of it capable of untethered operation through high-density chemical cells.

The main feature of the system is thus clearly its steerable track and guide system, as well as the miniaturization of the drive- and steer-components to fit into the small-scale package desired by the sponsor. On-board sensing and computing

was based on COTS components, as were the communication links built into the system. Battery power was provided through a set of rechargeable metal hydride and lithium ion packs custom built for this system. An overall image of the design developed as part of this program is shown in Fig. 3, with a more detailed treatise to follow.

OVERALL SYSTEM: The overall assembly design shown in Fig. 3 shows the tread in a straight configuration, including the front and rear drive and posture hubs, the central enclosure, cameras and the deployable paddle, which allows the system to self-right itself, and climb onto obstacles taller than half its own height. If the tread were to be removed, the central elements of the system can be exploded to show the design, as depicted in Fig. 3.

TREAD: The frame system in its most simple form consists of the continuous belt, the drive spine attached to it, and the guide spines, which when deflected will bow the belt and retain the bent shape, thereby enacting a turning motion (see Fig. 4 for the basic element design).

Different continuous-belt designs were explored in terms of their grouser- and webbing-arrangement, including typical slatted conveyor sections, plastic grousers with intermediate webbing, as well as continuously-cast urethane-webbed grouser designs. The latter approach proved to be the most rugged, reliable and straightforward to manufacture. The prototyping method utilized is based on developing an SLA positive of a tread section, developing a silicon mold and pouring a urethane section. Several sections if glued together would result in a continuous belt, with the required curvature capability as shown in Fig. 5.

The drive spine, which is glued to the inside curve on the continuous belt, is made from a custom-molded medium-durometer urethane, into which are embedded Kevlar backbone fibers, as well as a set of drive pins with low-friction ends to assure proper engagement with the drive sprockets and low-friction passage within the guide spine(s).

DRIVE & STEER: The continuous tread is driven at the end of the assembly by a multi-stage planetary gearbox, driven by a coaxially-mounted brushless motor. A set of clutch and brakes internal to the assembly allow the gearbox to change ratios, thereby giving the system a low- and high-speed capability – this was the only feasible approach to simultaneously satisfy the high-speed and high-torque climbing requirements. An image of the assembled and cross-section of the drive section are shown in Fig. 6.

The AURORA system uses two separate steering motors in order to steer the tread – one on each end offset longitudinally from the cylindrical end sections. The steering is achieved by a stepper motor, geared through a helical gear set, driving each cylindrical end section to a $+/-30^\circ$ angle, achieving thus a net 60° steering-curvature angle. An image of the steering actuator is shown in Fig. 7.

STRUCTURAL HOUSING: The design for AURORA utilizes the *mono-coque* method, by which the structural support and strength of the system is provided through the environmental enclosure. The enclosure is made of a custom-laid

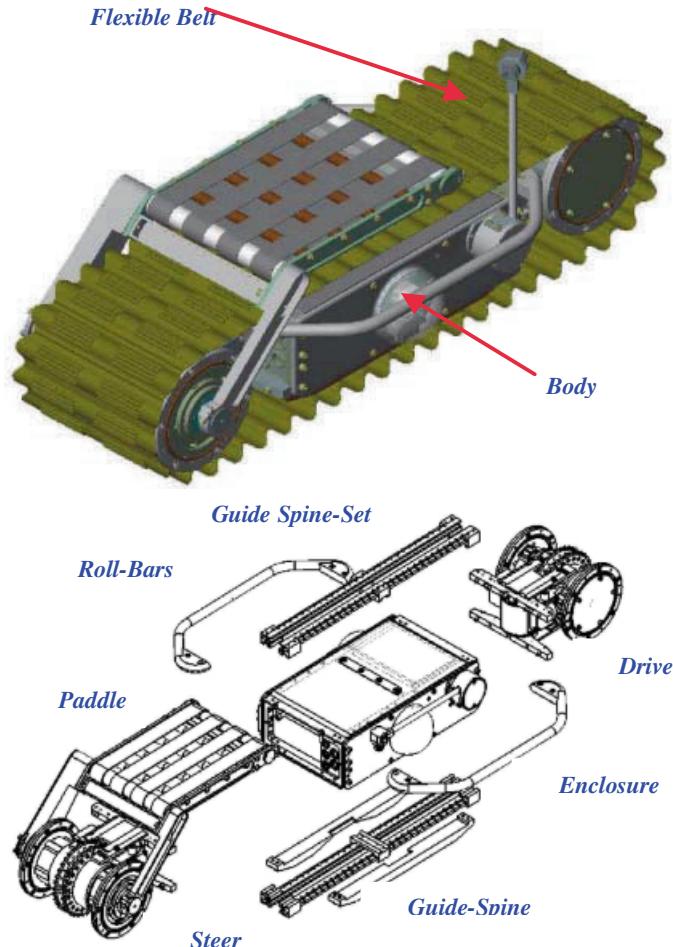


Fig. 3. Assembly design for AURORA

carbon-fiber epoxy rectangle, with embedded frame elements for the enclosure 'lids' and the battery compartment separator. The batteries are accessible through a separate cover in the lid, while the electronics (computing, navigation, communications, custom PCBs) are shock-mounted and heat-sunk to endplates bolted to the end sections of the enclosure. An exploded view of this assembly is shown in Fig. 8.

ELECTRONICS: The overall electronics architecture for the AURORA system is based on a simple PC 104 Pentium-based computer stack, which interfaces to all peripherals. Communications and control of all on-board devices is achieved over a dedicated parallel I/O card, which interfaces to a custom-built PCB to drive all the motors (PWM & Steps), control all the cameras (pan/tilt servos) and signal switching, while also processing the navigational attitude data stream (roll/pitch/yaw with

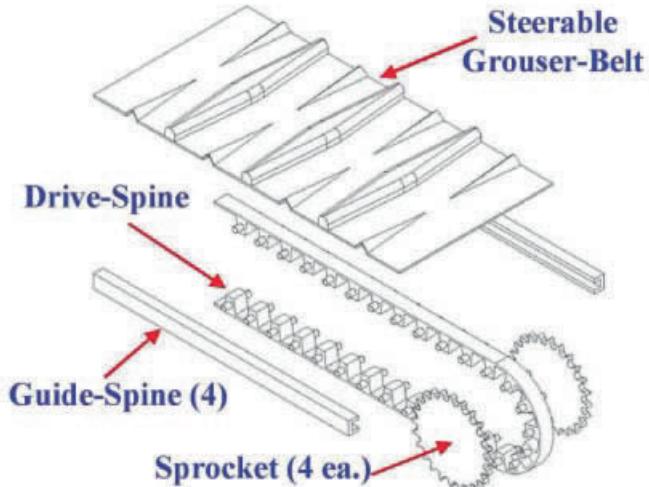


Fig. 4. Basic tread-elements



Fig. 5. Steerable monotread design & parts

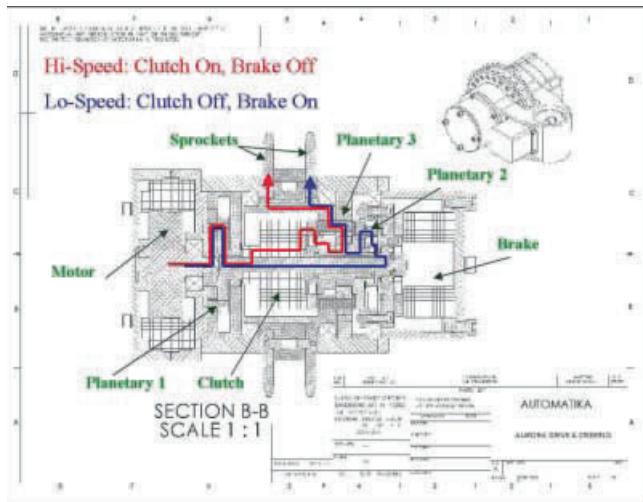


Fig. 6. Monobelt drive actuator system

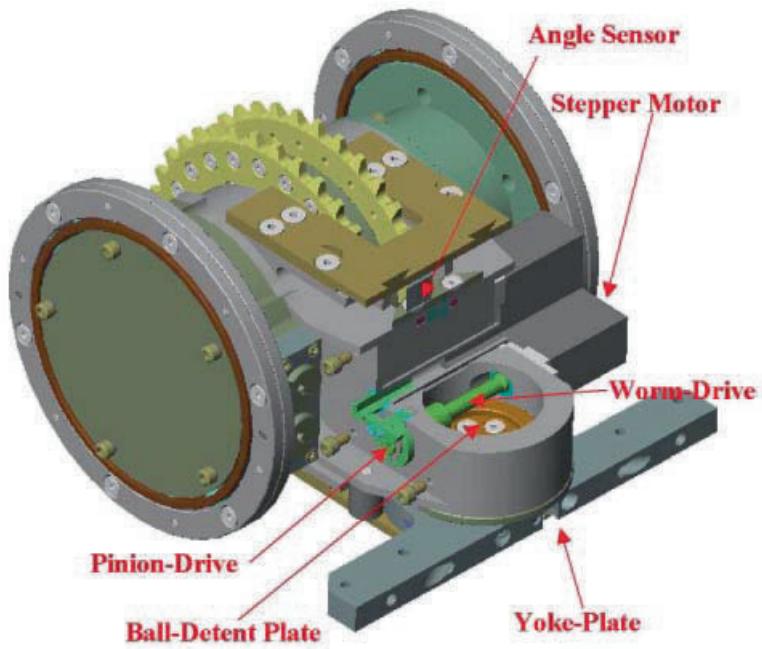


Fig. 7. Monotread steering actuator system

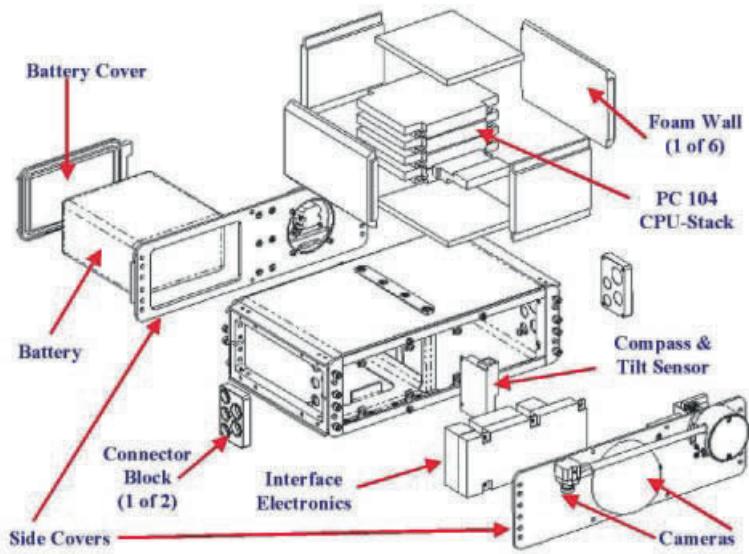


Fig. 8. Enclosure and internals view

magnetic compass heading). All video is frame-grabbed and digitized for transmission with all status and update data over the wireless Ethernet link using standard WaveLAN 2.4GHz PC card solutions. The power pack is controlled and monitored for voltage and (dis-)charge rates at all times. A simplified version of the implemented architecture is depicted in Fig. 9

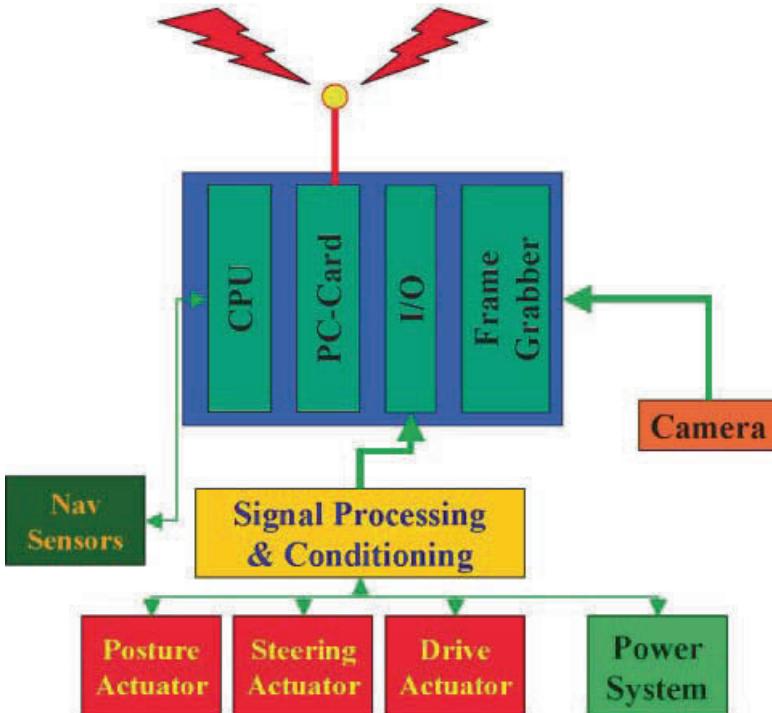


Fig. 9. Electronics architecture diagram

SOFTWARE: The main control mode for AURORA is through teleoperation over the wireless LAN link, utilizing a separate analog video- & audio-transmitter channel provided over a separate non-interfering frequency. However, there are several on-board autonomous or self-guarding algorithms that are intended to protect the system itself.

These algorithms are simple interrupt-driven semaphore-based loops that monitor key variables. Hardware protection is enabled by monitoring temperatures and currents, in conjunction with pitch, to allow the system to automatically switch gears (down or up) to reduce the load on the motor and achieve as high a speed as possible without overheating the motor, electronics and battery pack. A posture-guarding algorithm monitors the navigation sensor to ensure that the system is not operating on too steep a hill or slope. The system simply warns the user, without interrupting operation, as it can roll over and tumble and either self-right itself

or continue operating inverted (operator choice). Comm. reestablishment is based on the statistical data provided through the wireless Ethernet-based control link, allowing the system to determine link integrity. If link integrity is poor or packet loss excessive, the system will automatically stop and retrace a backwards-played script of commands for a period of up to a minute, while continually monitoring the link quality, before returning control back to the (tele-)operator.

EXTERNAL SENSING: The main sensory device(s) for the AURORA system are based on visual and audible feedback. Due to the systems' low-to-the-ground design, its need to operate in confined spaces or in stealthy modes, and the packaging requirement of minimizing protruding parts, the design developed into a three-camera arrangement. Two cameras are mounted inside of Lexan spheres, which are mounted to the side covers of the enclosure – each of them sits on a panning miniature RC servo controlled via PWM. The third camera is mounted atop a deployable flexible stalk (to avoid damage during flip over if deployed), which has a rotational joint within the enclosure to raise/lower the stalk, while a panning actuator turns the entire stalk, thereby creating a panning motion of the camera. All cameras are miniature CMOS-based single-board color cameras, encapsulated into silicone for environmental sealing. No external lighting has yet been provided for, but discussions about adding low-power high-intensity 'white' LED light rings are being considered for a future version.

SYSTEM ASSEMBLY: A fully assembled pre-prototype locomotion platform had been built to test the tread designs, and to allow teleoperation evaluation/training of operators via built-in RC servos and standard radio control interfaces. The pre-prototype that was built and tested is shown in Fig. 10.

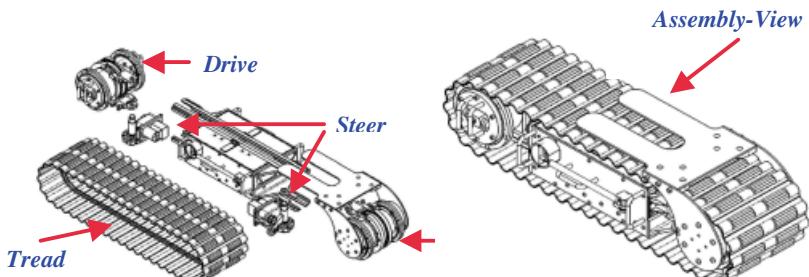


Fig. 10. RC tread-test platform – CAD and as built

5 Prototype Testing

The computer-controllable AURORA system is currently undergoing assembly and testing, with an expected completion date of late-summer/early-fall 2001. In the meantime though, a pre-prototype had been built to not only test the tread designs, but to also allow teleoperation training of operators via built-in RC servos and standard radio control interfaces. The pre-prototype that was built and tested is shown in Fig. 11.



Fig. 11. AURORA test prototype

Initial functional testing indicates that frictional losses due to the use of the captured steerable belt are fairly reasonable. Power consumption of 110% over nominal no-belt idle speeds was measured, while steering consumption jumped to 150% for the sharp 60° turning radius at average speeds. The guide spines were found to work surprisingly well, even becoming self-cleaning due to the drive pins in the drive spine keeping debris from the groove. The continuous belt systems' lack of pre-tension, afforded by the complete capture of the belt, clearly impacted to high-efficiency steering of the system. The guide spine material will have to be chosen carefully so as to maximize wear life without impacting power draw excessively. The computer-controlled prototype system was shown to be capable of climbing steep gravel slopes and driving through vegetative stands much taller than its own size (see Fig. 12). Stair climbing was also shown to work; two point runs were necessary on very narrow (fire-escape sized) stair landings.

6 Summary & Conclusion

The monotread design and prototype system presented in this paper is believed to be novel and a departure from the typical tracked (Crimson, 1992) locomotor system designs, as well as those considered hybrids (Hirose, 1999) and extreme (Hirose,



Fig. 12. Outdoor prototype testing system situations

1993). Its design represents a minimalist approach to rough-terrain tracked locomotion, and is believed to offer viable and beneficial alternatives to existing systems. The design for the tread was proven to be possible, with an optimal tread- and flex-structure design presented herein. Integration of mission-relevant sensors, computing, communications and power systems was shown to be feasible in the required size range. The choice of battery chemistry clearly drives the weight specification, with the final system exceeding its weight target by 15% (23 lbs.). Test results indicate that the system is capable of outdoor rough terrain driving and steering, while also climbing stairs and achieving sufficient traction and flotation in sandy, wet and soft soil conditions.

7 Future Work

The system presented herein represents the first generation of steerable monotreads for use as remote-controlled and autonomous inspection and reconnaissance mobile platforms. The next step in development will be the augmentation of system capabilities in the areas of sensor integration (microwave radar, acoustic, etc.) and power density (fuel cell or other battery chemistry) to enhance its mission-specific autonomy behaviors. Since the continuous belt is the main feature of the system, continued work will emphasize the selection of a proper combination of materials and durometers, and developing a single-step fabrication method to reduce labor and prototyping costs. Alternate applications for this system design are also being explored in other completely unrelated commercial arenas.

Acknowledgements.

The steerable monobelt system development was funded at Automatika, Inc., by DARPA under research grant #00-C-0655. We wish to further acknowledge the help from many other team members at Automatika, Inc. to develop the tread and structure systems. The notion of a flip-up camera was proposed by M. Hebert at CMU, and we have thus dubbed the deployable stalk camera the MartialCAM in his honor. The above-described AURORA system has been filed with the US and International Patent Offices under the PTC for patent protection and has a patent-pending status.

References

- Automatika (2001) URL <http://www.automatika.com/default.htm>.
- Crimson F (1992) Us military tracked vehicles. Motorbooks International Publishers & Wholesalers, Osceola.
- Hirose S (1993) Biologically inspired robots (snake-like locomotor and manipulator. Oxford University Press, Oxford.
- Hirose S (1999) Super-mechano-colony and SMC rover with detachable wheel units proc. In COE workshop, pp 67–72.
- Inuktun (2001) URL <http://www.inuktun.com/>.
- iRobot (2001) URL <http://www.irobot.com>.
- Menzel P, D'Alusio F (2001) Robo sapiens, a material world book. MIT Press, Cambridge.
- Remotec (2001) URL <http://www.army-technology.com/contractors/mines/remotec/>.
- Schempf H (1999) Mobile robots and walking machines. In Nof SY (ed) handbook of industrial robotics. John Wiley & Sons, New York.
- Schraft R, Schmierer G (1998) service robots. Springer Verlag, Berlin.
- TeleRob (2001) URL <http://www.telerob.com/>.

Analysis of Robotic Locomotion Devices for the Gastrointestinal Tract

L. Phee, A. Menciassi, D. Accoto, C. Stefanini, and P. Dario

Scuola Superiore Sant'Anna – MiTech Lab via Carducci, 40–56127 Pisa, Italy,
dario@mail-arts.sssup.it

Abstract. Various types of rigid and flexible endoscopes are used to inspect and to perform therapeutic procedures on different parts of the gastrointestinal (GI) tract. Due to the working characteristics of conventional endoscopes, most GI endoscopy procedures are unpleasant for the patient, and are technically demanding for the endoscopist. The authors are developing minirobots for semi-autonomous or autonomous locomotion in the GI tract. In this paper, the authors illustrate the systematic approach to the problem of “effective” locomotion in the GI tract and the critical analysis of “inchworm” locomotion devices, based on extensor and clamper mechanisms. The fundamentals of locomotion and the practical problems encountered during the development and the testing (*in vitro* and *in vivo*) of these devices are discussed. Finally, two mini-robots capable of propelling themselves in the colon and potentially suitable to perform rectum-sigmoidoscopy and colonoscopy are presented.

1 Introduction

Diseases of the gastrointestinal (GI) tract (stomach and colon cancer, ulcerative colitis, etc.), are common in most countries. Cancers of the colon are the fourth most commonly diagnosed cancers and rank second among cancer deaths in the United States. However, most colon cancers can be cured if detected at their early stages. Unfortunately, due to the pain and discomfort experienced by the patient, flexible GI endoscopy procedures like colonoscopy and gastroscopy are especially unpopular. Endoscopists also complain of the technical difficulties involved in introducing long, flexible shafts into the patient’s anus or mouth (Cotton, Williams, 1990). Any-way, if painless and less technically demanding screening and diagnosis methods for the GI tract could be developed, mass screening of the population could indeed be realised resulting in many lives being saved.

There exist alternative methods to detect ailments of the GI tract without the introduction of intrusive devices into the human body. However, there remains only one way to perform therapeutic procedures and biopsies; that is from within the GI tract. Phee, et al. (Phee, Ng, Chen, Seow-Choen, Davies, 1997b) presented a thorough survey of these works. More recently, researchers (Gisela, Miller, Tang, Raz, 2000) aim to develop autonomous capsules to be swallowed in order to perform diagnosis and even therapy of the gastrointestinal tract (Iddan, Meron, Glukhovsky, Swain, 2000). Locomotion of this capsule is entirely due to natural peristaltic movements of the GI. Without locomotive means, it is not possible to stop the capsule at any point of time for a “better look” or to navigate to an area of interest for ther-

peutic procedures. We are investigating devices that can propel semi-autonomously or autonomously in some parts of the GI tract (Dario, Carrozza, Pietrabissa, 1999a).

In this paper, we propose objective and quantitative criteria of efficiency for locomotion in the gastrointestinal tract. To pursue this aim, the devices for locomotion are analytically evaluated by considering not only the mechanical aspects but also – and firstly – the interaction model between the devices and the environment (the digestive tract in this case) which dramatically affects the behaviour and the efficiency of any navigation systems. A technological solution is then presented and the results of in vitro and in vivo tests are discussed and interpreted. The main novelty of this work, in respect to the authors' previous papers (Carrozza, Lencioni, Magnani, D'Attanasio, Dario, 1997; Dario, Carrozza, Lencioni, Magnani, D'Attanasio, 1997; Dario et al., 1999a) consists of the analytical approach to the problem of locomotion in the gastrointestinal tract. By exploiting just a phenomenological approach to the locomotion problem in the intestine and by simply adapting standard locomotion devices to navigate in the digestive system, it is possible to design and fabricate several locomotion solutions. However, it is not possible to understand the reasons of failures or to optimize the performances of the prototypes. The model we propose may be the basis for future developments in this very promising field.

2 Inchworm Type Locomotion

The human GI tract is a three dimensional maze of tubular soft tissue. The lack of mechanical constraints and the elastic and slippery nature of its walls contribute to the difficulty in developing a device which can propel itself inside the digestive tract. One might imagine that a miniature wheeled vehicle would probably 'drive' itself easily into the GI tract. However, the authors have already studied the frictional properties of the gut (Accoto, Stefanini, Phee, Arena, Pernorio, Menciassi, Carrozza, Dario, 2001) and concluded that the coefficient of friction of the GI tract is too low to enable conventional wheels or tracks to attain sufficient traction forces without damaging the tissue. Similarly, it would be difficult for a snake-like device to gain enough traction to propel itself inside the GI tract. As such, to solve the problem of locomotion in the GI tract, the authors have looked beyond traditional means of locomotion.

The larva of the Geometer moth, more commonly known as an inchworm, exhibits mobility that is flexible, robust and extremely stable. The inchworm is capable of manoeuvring in small places with arbitrary orientations to gravity and can withstand substantial external forces attempting to divert it from its intended course. This type of locomotion is particularly suited to unstructured or even hostile environments where wheels and tracks fail (Hirose, Cave, Goulden, 1993). An inchworm device would function especially well in a tubular, three-dimensional terrain. Realising its potential, various researchers have developed pipe inspection devices (Fukuda, Hosokai, Uemura, 1989) based on this inchworm type locomotion. With the successes of the pipe inspection devices, researchers (Grundfest, Burdick, Slatkin, 1997; Ikuta, Tsukamoto, Hirose, 1988; Ng, Phee, Seow-Choen, 2000; Phee,

Ng, Chen, Seow-Choen, 1997a; Slatkin, Burdick, Grundfest, 1997; Treat, Trimmer, 1997) thought that self-propelled robotic endoscopes using the same locomotion type could be developed for navigation in the GI tract and other tubular organs. In the authors' laboratory, inchworm devices (Dario et al., 1999a; Dario, Carrozza, Pietrabissa, Magnani, Lencioni, 1999b) catered for locomotion in the GI tract are being developed.

2.1 Description of the Inchworm Concept

An inchworm device is made up of basically two types of actuators: clamer and extensor. The clamer is used to adhere or clamp the device securely onto the "terrain" while the extensor brings about a positive displacement (stroke). The simplest inchworm device consists of two clammers at its ends and one extensor at its mid section. Figure 1 shows the gait sequence in which this device propels itself forward.

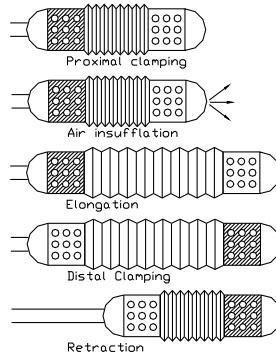


Fig. 1. Schematic diagram illustrating the sequence of the inchworm type locomotion concept. The shaded area on the distal and proximal clamping actuators indicates the active clamping states

Each cycle results in a net forward displacement known as the stroke of the inchworm. Primarily, the stroke is the difference in length of the extensor in its elongated and retracted states. Theoretically, n clammers and extensors can be integrated together to form more complicated inchworm devices. Grundfest et al. (Grundfest et al., 1997) described some of the more complex gait sequences in which these inchworm devices propel themselves.

2.2 Locomotive Efficiency of the Inchworm

Theoretically, the inchworm device should advance a distance equal to its stroke after each cycle of the locomotive sequence. However, this is not true in a real scenario. Losses could result due to factors like slippage, difficult bends and collapsible terrain. As such, we define the inchworm locomotion efficiency η as the

ratio between the real advancement and the theoretical one. To optimize the locomotive efficiency, it is important to study quantitatively which factors influence the value of η . We propose to break the inchworm's locomotion into three distinct features: elongation, retraction and clamping. The individual efficiency of each of these mechanisms contributes to the overall efficiency η of the locomotion system. The overall locomotion efficiency can be represented by:

$$\eta = \eta_e \eta_r \eta_c,$$

where η_e , η_r and η_c are the efficiencies of the elongation, retraction and clamping mechanisms respectively.

2.3 Modelling of the Elongation Mechanism

Separate studies (Gregersen, Kassab, 1996; Hoeg, Slatkin, Burdick, Grundfest, 2000) have shown that the viscoelastic properties of pig's intestine are quite similar to human intestine in terms of dimensions and biomechanical properties. The authors have exploited these studies and have performed additional tests to develop a model of locomotion in the GI tract. The GI tract (colon and small intestines) can be geometrically described as cylindrical shells having the following typical dimensions:

Wall thickness:	$t = 2$ to 3 mm
Average diameter: (when not folded or stretched)	$\Phi = 20$ to 50 mm

It would be quite difficult to develop a mathematical model taking into account all the factors that influence the locomotion efficiency of an endoscopic device. On the contrary, a simplified model, which takes into account all practical aspects, will provide sufficient guidance to design the locomotive device. The viscoelastic properties of the GI tract have been modelled using a simple quadratic expression which derives from the expansion in Taylor series, up to the second order term, of well-known pseudo-elastic constitutive equations (Fung, 1990):

$$\sigma = \gamma \varepsilon^2, \quad (1)$$

where σ is the stress and ε the strain. For longitudinal strips of tissue, it has been found that $\gamma = 5.0 \text{ MPa} \pm 30\%$ using standard measurement methods (Fung, 1990; Gregersen, Kassab, 1996) applied to pig's colon.

Prior to elongation (with reference to the air insufflation stage of Fig. 1), the extensor is in the retracted state and the proximal clamer is activated and assumed to grip onto the GI tract without slippage. In order for the extensor to elongate, the inchworm device has to overcome an initial impeding force F_c ¹ whose intensity comes from the effects of superficial friction and forces required to push open the

¹ The values of F_c are different in the elongation and retraction processes.

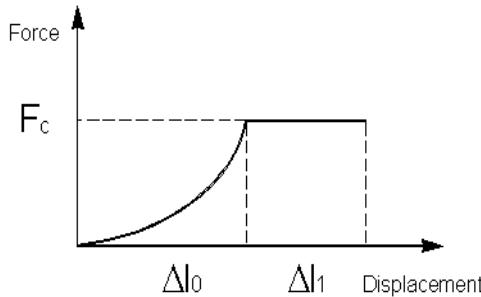


Fig. 2. Relationship of force vs. elongation

collapsed lumen in the frontal direction. Figure 2 describes the *force* vs. *elongation* relation. Due to the existence of friction between the device and the GI tract, part of the compliant intestinal wall would shift together with the advancing distal head. The impeding force increases till a value F_c as the intestinal wall is being stretched.

Up till this point, the displacement Δl_0 does not contribute to the effective stroke of the inchworm device. However, beyond this point, slippage would occur as the propulsion force exceeds the impeding forces resulting in F_c becoming constant.

The distal head of the device begins to advance forward with respect to the intestinal wall for a distance Δl_1 , which is the effective stroke of the device. The total elongation displacement of the inchworm device is defined as Δl_t . Evidently:

$$\Delta l_t = \Delta l_0 + \Delta l_1.$$

Since only Δl_1 contributes to locomotion, we can define the inchworm elongation efficiency η_e as

$$\eta_e = \frac{\Delta l_1}{\Delta l_t} = 1 - \frac{\Delta l_0}{\Delta l_t}. \quad (2)$$

Since the GI tract has very thin walls as compared to its diameter, it can be assumed that its thickness does not change during the application of forces. A further assumption is that the deformation of tissue is essentially due to a uniform tension applied over the cross section of the GI tract. Therefore, the stress in the tissue is

$$\sigma = \frac{F_c}{\pi \Phi t}. \quad (3)$$

The related strain, from (1), is

$$\varepsilon = \sqrt{\frac{F_c}{\pi \gamma \Phi t}}. \quad (4)$$

The strain may be regarded as the ratio

$$\varepsilon = \frac{\Delta l_0}{l_0}, \quad (5)$$

where l_0 is the distance between the proximal clamp and the distal tip before the elongation occurs. Consequently, the following relation can be derived

$$\Delta l_0 = l_0 \sqrt{\frac{F_c}{\pi \gamma \Phi t}}. \quad (6)$$

From (2) and (6) it descends

$$\eta_e = 1 - \frac{l_0}{\Delta l_t} \sqrt{\frac{F_c}{\pi \gamma \Phi t}}. \quad (7)$$

If biomechanical properties of the GI tract are assumed not to vary, from (6) it can be seen that the efficiency of elongation is still primarily dependent on the stroke Δl_t of the device. A longer stroke would yield higher elongation efficiency. However, it is important to note that longer strokes are undesirable for navigating bends. A longer stroke would also give rise to longer cycle times and increases the stiffness of the device. It is also noted that η_e is dependent on F_c . The lower F_c is, the higher η_e would be. Fortunately, the frictional component of F_c is dependent on the material and mechanical design of the distal head. It should be made with material with low coefficient of friction and geometrically designed so that it does not oppose the oncoming GI tissue. Air insufflation to pre-open the lumen also drastically reduces F_c .

2.4 Modelling of the Retraction Mechanism

The force vs. elongation behavior depicted in Fig. 2 still applies during the retraction phase. But in this case, the distal clamp is activated as the bellow retracts. Δl_1 , the effective retraction stroke, now represents the distance traversed by the proximal end after slipping has occurred. Thus, the inchworm retraction efficiency η_r can be written as:

$$\eta_r = \frac{\Delta l_1}{\Delta l_t} = 1 - \frac{\Delta l_0}{\Delta l_t}. \quad (8)$$

Equations (3), (4) and (5) still apply but l_0 in (4) now refers to the distance between the proximal end and the nearest proximal mesentery muscle that holds the GI tract in place (Fig. 3). If the device is not a wireless system, much of the generated propulsion force would be used to pull the power supply cable or 'tail' as the device propels itself forward. Together with the weight of the device, this pulling force contributes to the magnitude of F_c . In comparison, F_c in the retraction phase would be much greater than that of the elongation phase.

Thus, from (7), the retraction efficiency is derived as

$$\eta_r = 1 - \frac{l_0}{\Delta l_t} \sqrt{\frac{F_c}{\pi \gamma \Phi t}}. \quad (9)$$

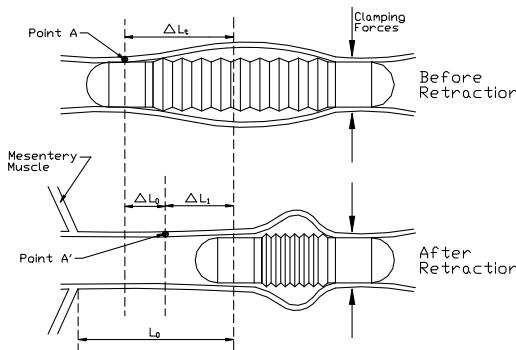


Fig. 3. The retraction mechanism

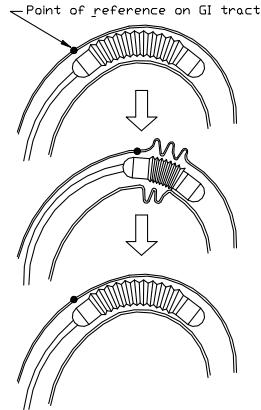


Fig. 4. The accordion effect

The authors' past *in vivo* experimental results (Carrozza et al., 1997; Dario et al., 1997, 1999a) have shown that high inchworm retraction efficiency η_r is generally more difficult to achieve as compared to elongation and clamping efficiencies. In addition to the higher value of F_c in the retraction phase, the GI tract is also very loosely supported in the body cavity. It is attached to other surrounding organs by means of elastic mesenteries that allow the gut to slide freely in between the other organs. The inchworm configuration of Fig. 1 would perform reasonably well in a straight path. But in an acute and unconstrained bend, the GI tract could easily 'crumple' up and be thrown into folds during the retraction phase. With reference to Fig. 4, this means that points A and A' would remain more or less in the same position with respect to the proximal clamer. This causes Δl_0 to be near or even equal the value of Δl_t resulting in very low retraction efficiency η_r . The folds that result from the retraction phase would be straightened during the elongation phase. Repeating the gait sequence would only result in the device elongating and retracting at the same position in the GI tract. The authors have termed this the accordion effect (Fig. 4). The accordion effect has not been taken into consideration in (7) and (9) because it not only depends on the mechanical properties of the GI tissue and on the locomotion parameters of the device, but also on the anatomical conformation of the GI tract and on previous strains imposed by the same device. Experimental evidences have shown that to minimise this accordion effect, a long stroke Δl_t and low F_c would be desirable. Air insufflation, reducing the size of the 'tail' (if one exists) and using light-weight materials of low coefficient of friction are ways to reduce F_c .

2.5 Modelling of the Clamping Mechanism

The modelling of the clamping mechanism is more straightforward as compared to that of the elongation and retraction mechanisms. The clamping force F_g is defined

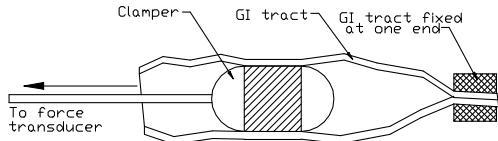


Fig. 5. Experimental setup to determine F_g .

as the holding or adhesion force that the device applies on the GI tissue. The higher F_g is, the better the device attaches itself onto the GI tract. The rule of thumb is that if

$$F_g / g \geq F_c,$$

then the single clasper is rendered effective. Thus, the clamping efficiency on a large number of locomotion cycles can be defined as

$$\eta_c = \frac{N_e}{N_t},$$

where N_e is the number of effective clampings (when $F_g \geq F_c$) and N_t is the total number of attempted clampings. An efficient clasper would have a high value of F_g and is capable of grasping consistently onto the GI tissue. Regardless of the type of clamping actuation used, a simple experimental setup (as depicted in Fig. 5) can be used to determine F_g . A piece of GI tract (e.g. pig's colon) is fixed at one end. The clasper is introduced into the other end and activated. The clasper is pulled slowly to the left by means of a motor. A force transducer measures this pulling force. The highest force registered before slippage (between the clasper and the GI tract) occurs is taken to be F_g .

3 Types of Actuators and Mechanisms

In the inchworm locomotion concept, the main task of the extensor is to bring about a forward net displacement or stroke. The clasper, on the other hand, is used to gain a firm hold on the GI tract. This section discusses the different types of extensors and clampers that can be integrated into an inchworm device.

3.1 Extensors

In previous attempts at developing an inchworm device for the GI tract (Carrozza et al., 1997; Dario et al., 1997), the authors used a flexible rubber bellow as the extensor. The introduction of pressurised air would cause the bellow to extend while a vacuum would cause it to retract. One of the greatest advantages of using a rubber bellow is its ability to conform to the bends that exist in the GI tract. Its stroke length can also be regulated by varying the supplied air pressure. However, being

pneumatically actuated, the device would most likely have a 'tail' that connects the device to an external pneumatic distributor. Another disadvantage of using a rubber bellow is the presence of undesirable pressure losses when the air passages are miniaturised. The effects of friction from the walls of the air passages become more prominent when the size of the bellow and its connecting air tubings are decreased. These losses decrease the extensor's efficiency.

The authors are also exploring different extensor solutions more suitable for a wireless autonomous locomotion device. By avoiding the use of pneumatic actuators and by exploiting on-board batteries, it will be possible to design wireless devices which could be swallowed by the patient, as illustrated in the introduction. Commercial minimotors, linear or rotative, could be easily integrated in an extensor tube or bellow to produce the required elongation and retraction. However, energy considerations are important when choosing an appropriate extensor. For example, if the estimated force required to propel a swallowable device is in the order of 0.2 N and the length of the GI tract is between 4 and 5 meters (adult men) (Davenport, 1982), the required energy can be estimated to be in the order of 1 Joule. The energy density for a new generation battery is about 1.5×10^3 J/cm³ (Vincent, 1998) and the maximum space available in a swallowable device for energy storing is estimated to be in the order of 0.1×10^{-1} cm³. This means that the efficiency of the entire locomotion system can be calculated to be more than 6.7 percent. Consequently the minimotor, as a component of the system, has to perform at 15–20 percent efficiency or better. Mechanisms based on the sudden expansion of gas or vapors could be designed in order to obtain an "on-board" extensor. An extensible chamber can be elongated by pressurised vapor by heating an appropriate fluid to its transition temperature. From observation of thermodynamic tables of biphasic fluids, generation of high-pressured vapor is possible even at relatively low temperatures. Biomimetic gels that typically behave as tunable rigidity springs are also possible alternative extensors. These actuators allow large displacement, long actuation durations, a high power/weight density and are intrinsically compliant. Gels for biomimetic actuation include: polyelectrolyte gels (Okuzaki, Osada, 1994), ion-polymer metal composites (IPMC) (Shahinpoor, 1996) and conducting polymers (Della Santa, Mazzoldi, De Rossi, 1997). All these solutions show some advantages and several drawbacks for their implementation in a next wireless miniaturised inchworm device. Discussion and evaluation of these solutions are not in the scope of this paper.

3.2 Clampers

The greatest difficulty in clamping onto the GI tract is the grasping of the soft, slippery and often wet intestinal walls. Using a balloon (Grundfest et al., 1997) could over-distend the intestinal walls (thus causing ischemia) and still not achieve sufficient traction forces due to the extremely low coefficient of friction of the GI tract. Sharp objects like hooks may be viable but are likely to cause damage to the delicate tissue. In the authors' previous works, a vacuum was employed to 'suck' onto the GI wall. We observed that with the above-mentioned suction clasper is that with the introduction of a vacuum, tubular sections of the GI tract (colon, small

intestine, etc.) would collapse around any rigid object that is in the vicinity of the suction supply. In doing so, the GI tract would take the shape of the hard object, filling any gaps, holes or troughs in the process. To take advantage of this situation, instead of using the vacuum to attain traction forces, it could be used to cause the tissue to fall into the 'jaws' of a mechanical clamp (e.g. grippers, pincers, forceps). Having a prominent hold on the tissue, the 'jaws' of the clamp can easily close in for a positive mechanical grasp. The movement of the 'jaws' can be brought about by linear actuators (e.g. rubber bellows, minimotors) This concept is illustrated in Fig. 6.

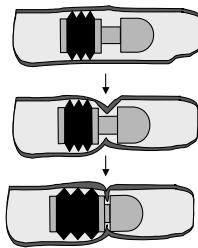


Fig. 6. Concept of the mechanical and suction clamer

The grip of the mechanical clamp can be improved by using materials with high coefficient of friction. Changing the profile of the clamping surfaces like introducing grooves and steps would also enhance the grip and improve the clamping force F_g . However such measures may inflict damages to the delicate GI tissue. Histological tests are in progress to evaluate the damage, if any, caused by different mechanical clampers fabricated with different materials and exerting different forces. Preliminary experiments (using the experimental setup of Fig. 5) showed that this mechanical and suction clamer yield very high clamping efficiency (c. In fact, depending on how tightly the 'jaws' are closed, the F_g attained could be well over 30 N. Plastic deformation or rupture of the GI tract could result at such high values of F_g .

4 First Prototype for Navigation in the Colon: Experimental Results

In the authors' quest to develop a locomotive device for the GI tract, they have thus far built and experimented with various inchworm prototypes. Valuable lessons have been learnt with every successive prototype and many modifications have been made to improve the performance of the device. In this paper, the latest locomotion prototype was able to navigate in the colon or, at least, in the rectum, sigmoid and descending colon is described and experimental results are discussed.

Figure 7 shows a picture of the prototype that was designed to represent an alternative to the traditional colonoscope. It measures 24 mm in diameter and has lengths



Fig. 7. The inchworm prototype

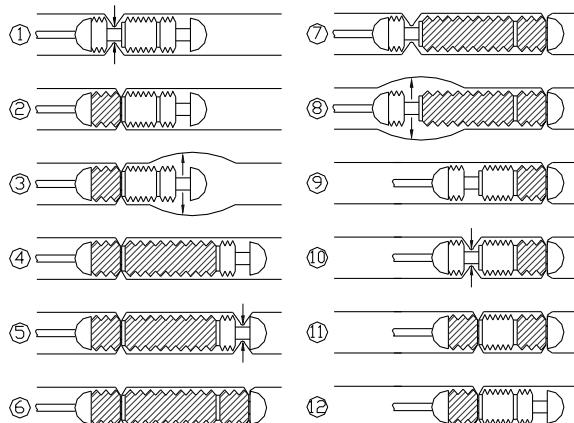


Fig. 8. Gait sequence of the inchworm prototype

of 115 mm and 195 mm when retracted and elongated respectively. A flexible rubber bellow acts as the extensor that gives the device a stroke of 80 mm when extended. The mechanical and suction clamer (as described in earlier) is employed at both its distal and proximal ends. Rubber bellows are used to open and close the clamp jaws while a hole of 2 mm diameter situated in between the jaws is responsible for suction and insufflation of air. Grooves were carved on the clamping walls to enhance the grip of the clamer. Five flexible air tubings, each of diameter 2 mm, exit from the proximal end of the device to form the 'tail' of the inchworm device. These are connected to an external pneumatic distributor. A computer controls the activation of solenoid valves which are responsible for driving the extensor and clammers according to a gait sequence as shown in Fig. 8. Working parameters like air pressures and time intervals can be easily monitored and modified from a human machine interface (HMI). These parameters and gait sequence can be changed on-line to suit different situations. For example, when approaching an acute bend, air of higher pressure could be insufflated to create more space in front of the device. Simultaneously, less pressure could be introduced into the extensor to increase flexibility and conformance. Furthermore, the elongation and retraction phases could be repeated to reduce the accordion effect.

4.1 In vitro Experiments

In vitro experiments were carried out by exploiting a “home-made” simulator made of polystyrene. This artificial path was patterned according to the indications of medical doctors in order to reproduce a realistic 3-dimensional structure of the human colon (Fig. 9).

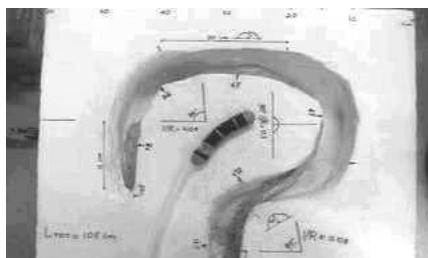


Fig. 9. Picture of the 3D colon simulator

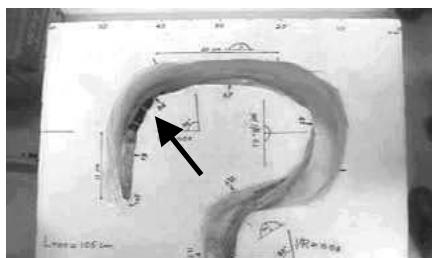


Fig. 10. Inchworm locomotion in the simulator

Table 1. Summary of in-vitro tests

Paths	T (s)	D (cm)	S (cm/s)	E (%)
Straight, 2D path	95	20	0.210	84
3D Curve of 80°, radius of curvature 12 cm, height of hump 5.5 cm	100	16	0.160	64
3D Curve of 180°, radius of curvature 10 cm, height of hump 5 cm	182	31	0.170	68
2D Curve of 84°, radius of curvature 12 cm	102	18	0.175	70
Entire simulated path of colon	583	105	0.18	72

The overall size of the simulator was 60 cm × 90 cm × 20 cm and it included 60 cm of straight path and 105 cm of realistic “colon-like” curved paths. A pig’s colon of length 150 cm was placed in the test bench and constrained at both its ends. Before inserting the prototype in the colon, all parameters were measured and optimized; the characteristics of the device’s locomotion were as follows:

Stroke length:	8 cm
Time of 1 cycle of gait sequence:	32 s
Theoretical speed:	0.25 cm/s

The device was inserted in the colon and its performance was recorded both for the entire path and for each colonic tract (bend or straight path). Table 1 summarized

the experimental results. The last phase of the prototype locomotion is illustrated in Fig. 10.

4.2 In vivo Experiments

An in vivo experiment was carried out on a 35 kg male pig under general anaesthesia. The experiment was performed in an authorized laboratory, with the assistance and collaboration of a specially trained medical team in accordance to all the ethical considerations and the regulatory issues related to animal experiments. Prior to the experiment, the pig's bowels were properly prepared for colonoscopy. A colonoscopy procedure using a conventional colonoscope was first performed by a skilled endoscopist to inspect the terrain of the colon. Two bends in the colonic tract were revealed during the inspection. The first, a gentle bend, was situated about 30 cm from the anus while the second, an acute kink, was situated about 50 cm from the anus. After the withdrawal of the colonoscope, the prototype was introduced manually about 10 cm into the pig's anus. Upon activation of the gait sequence, the inchworm device propelled itself a distance of 40 cm into the colon with an estimated speed of 0.19 cm/s. After which, its speed decreases and the device was observed to remain stationary 55 cm from the anus. The gait sequence was stopped and the device was retrieved by manually pulling its 'tail'. A second inspection with the conventional colonoscope revealed reddish clamping marks on the colonic walls. These were caused by the clamps as the device inched its way into the colon. According to the endoscopist, these marks were similar to those that appear occasionally during colonoscopy when the scope stretches the colonic walls excessively. These marks normally heal within a few days. From the positions of the clamp marks, it is evident that the device surpassed the first gentle bend without much difficulty. The last clamp mark was situated a few centimeters after the second, more acute bend. This showed that the distal head of the device conformed and surpassed the second bend, thus propelling itself for the entire part of the GI tract necessary for a rectum sigmoidoscopy diagnosis. In terms of efficiency, since the theoretical speed can be calculated to be 0.25 cm/s, the locomotion efficiency of the device was 76 percent along the straight portion of the colon; then it decreased to zero during the navigation of the second curve. The prototype has demonstrated high elongation and clamping efficiencies, and showed its ability to traverse pass the rectum, sigmoid and descending colon. The major drawback is its low retraction efficiency, which ultimately affects the overall locomotion efficiency. As such, the main reason for its failure to surpass entirely the second (less constrained) bend is due to the accordion effect as described in earlier.

5 Modified Clampers Disposition: A Second Prototype

In order to overcome the "bends problem", the configuration of the inchworm device has been modified. Figure 11 shows a photograph of the modified inchworm device. It has a diameter of 24 mm and a rigid length of 80 mm. Basically, the same types

of extensors and clampers are employed in this prototype but the disposition of clampers is different. The device is configured with two clampers such that one of which would always grasp onto the GI tissue that is distal with respect to the other clamping mechanism. In doing so, even if the tissue 'crumples' and is thrown into folds during the retraction phase, the device would always clamp onto tissue distal with respect to these folds, resulting in high locomotion efficiency. The first clamper is situated in between the second clamper. In the retraction phase, the linear actuator causes the first clamper to shift to the left, outside the enclosure of the second clamper. At this phase, the clamping surface of the first clamper is proximal with respect to the second one. The opposite happens in the elongated phase. The extensor extends to cause the first clamper to shift into the enclosure of the second clamper, resulting in the clamping surface of the first clamper being distal with respect to the second one. A repetitive sequence of these 2 phases would ensure that the clampers always grasp the tissue distal to its predecessor, thus eliminating the "bends problem".

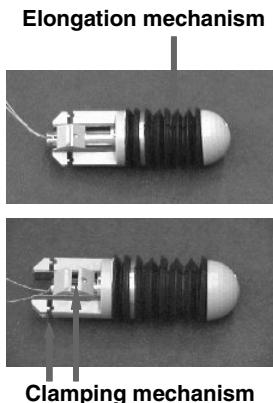


Fig. 11. The second inchworm prototype

5.1 Preliminary Experiments

In vitro experiments on pig's colon were performed with this last prototype. We tested both its ability to navigate in a straight intestine as well as in acute bends. The speed of the device in a straight intestine is 0.083 cm/min and the experimental efficiency is about 60 percent (calculated by considering the ratio between the experimental speed and the theoretical speed). Its speed during the navigation of bends is 0.050 cm/min with an experimental efficiency of about 40 percent. In general the speed of this prototype is less than the speed of the previous versions. This is due to the shorter stroke of this lastest inchworm device; anyway the speed could

be maximised by optimising the gait sequence. On the other hand, the locomotion efficiency is very good and this feature assures the ability to navigate in every conditions: Fig. 12 shows the device “climbing up” the intestine.



Fig. 12. The second inchworm prototype climbing the colon

6 Conclusions

In this paper, the authors illustrated their approach to the problem of “effective” locomotion in the GI tract and the critical analysis of “inchworm” locomotion devices, based on extensor and clamper mechanisms. This study leads to the design, fabrication and testing of a flexible locomotion device able to propel itself in the entire colon and to perform (when endowed with a video system) at least rectum-sigmoidoscopy (according to the *in vivo* test results). The inchworm type locomotion is analytically modelled by taking into account the interaction of locomotion devices with the locomotion environment. This model leads to the definition of a locomotive efficiency which can be broken in three factors: the efficiency of elongation, retraction and clamping. With these definitions, one can predict and compare the performances of new device configurations analytically. The following analysis of actuators and mechanisms for extensors and clampers aims to identify the best technological solution that implements the previously derived model. A clamping system based on suction and mechanical grasping was designed and implemented in a locomotion device. This clamping solution assures a high clamping efficiency and has better reliability than previous solutions developed by authors (Dario et al., 1997) and other researchers (Ng et al., 2000) based essentially on pneumatic suction or over-extension of the GI tissue. The inchworm device exploiting the above clamping mechanism was fabricated and tested *in vitro* and *in vivo*, accordingly to the current ethical rules. It exhibited high locomotion efficiency in tissues laid in straight paths (over 80 percent), while it is less effective in overcoming acute bends

because of the called “accordion effect”, which is related to the low retraction efficiency. However, the average efficiency is always more than 70 percent, even if considering curved paths. The minimum radius of curvature surpassed by the device is 10cm with a curvature angle of 180°. By considering a colon with a length of 150 cm, the recorded medium speed of 0.18 cm/s would allow a navigation time of less than 15 minutes. This result is in line with time taken to perform a conventional colonoscopy, which is approximately 15-20 minutes.

Besides further testing in vitro and in vivo the current prototypes, future activities will be mainly devoted to the investigation on the possible miniaturization of the devices and to the modeling of the behavior of the robots in the colon and, in general, in the gastrointestinal tract. The miniaturization process has the ultimate aim of developing minirobots for an effective application in the entire gastrointestinal tract; they will incorporate “on board” actuators, sensors and energy supply in order to become “wireless” highly integrated mechatronic machines. Further theoretical studies and models on colon tissue (or gastrointestinal tissue) and adhesion mechanisms between tissue and minirobots would also be performed in order to interpret the behavior of current minirobots (especially in presence of anomalies of the organs) and to predict the behavior of novel locomotion robots.

Acknowledgement

The authors wish to acknowledge that this paper is a result of the research accomplished with the financial support of the Intelligent Microsystem Center, Seoul, Korea, which is carrying out one of the 21st century’s New Frontier R&D Projects sponsored by the Korea Ministry of Science & Technology (www.microsystem.re.kr). The authors wish to thank M.G. Trivella, MD, Prof. P. Spinelli, MD, and Prof. A. Pietrabissa, MD, for their medical support and consultancy and Y.K. Jeong, A. Arena and G. Pernorio for their technical support.

References

- Accoto D, Stefanini C, Phee L, Arena A, Pernorio G, Menciassi A, Carrozza M, Dario P (2001) Measurements of the friction properties of the gastrointestinal tract. In 2nd AIMETA International Tribology Conference, Vienna.
- Carrozza MC, Lencioni L, Magnani B, D’Attanasio S, Dario P (1997) The development of a microrobot system for colonoscopy. In Proc. First Joint Conf. of Computer Vision, Virtual Reality and Robot, vol 1205 of *Medicine and Medical Robot. and Computer Assisted Surgery*, pp 779–788. Springer-Verlag, Berlin-Heidelberg.
- Cotton PB, Williams CB (1990) Practical gastrointestinal endoscopy. Blackwell Scientific Publications, Oxford.
- Dario P, Carrozza MC, Lencioni L, Magnani B, D’Attanasio S (1997) A micro robotic system for colonoscopy. In Proc. 1997 IEEE Int. Conf. Robot. Automat., pp 1567–1572, Albuquerque, NM.
- Dario P, Carrozza MC, Pietrabissa A (1999a) Development and in vitro tests of a miniature robotic system for computer assisted colonoscopy. Journal of Computer Aided Surgery, 4: 4–14.

- Dario P, Carrozza MC, Pietrabissa A, Magnani B, Lencioni L (1999b) Endoscopic robot. United States Patent, Patent Number 5,906,591.
- Davenport W (1982) Physiology of the digestive tract. Year Book Medical Pub., Chicago.
- Della Santa A, Mazzoldi A, De Rossi D (1997) Characterisation and modelling of a conducting polymer muscle-like actuators. *Smart Mater. Struct.*, 6(1):23–34.
- Fukuda T, Hosokai H, Uemura M (1989) Rubber gas actuator driven by hydrogen storage alloy for in-pipe inspection mobile robot with flexible structure. In Proc. 1989 IEEE Int. Conf. Robot. Automat., pp 1847–1852, Scottsdale, AZ.
- Fung YC (1990) Biomechanics: Motion, flow, stress and growth. Springer-Verlag, New York.
- Gisela L, Miller MM, Tang WC, Raz S (2000) Improved sensor pills for physiological monitoring. Nasa Tech Brief from JPL New Tech report NPO-20652, 25(2).
- Gregersen H, Kassab G (1996) Biomechanics of the gastrointestinal tract. *Neurogastroenterol. Mot.*, 8:277–297.
- Grundfest WS, Burdick JW, Slatkin AB (1997) Robotic endoscopy. United States Patent, Patent Number 5,662,587.
- Hirose S, Cave P, Goulden C (1993) Biologically inspired robots: Snake-like locomotors and manipulators. Oxford University Press, New York.
- Hoeg HD, Slatkin AB, Burdick JW, Grundfest WS (2000) Biomechanical modelling of the small intestine as required for the design and operation of a robotic endoscope. In Proc. 2000 IEEE Int. Conf. Robot. Automat., pp 1599–1606, San Francisco, CA.
- Iddan G, Meron G, Glukhovsky A, Swain P (2000) Wireless capsule endoscopy. *Nature*, 405 (6785):417.
- Ikuta K, Tsukamoto M, Hirose S (1988) Shape memory alloy servo actuator system with electric resistance feedback and application for active endoscope. In Proc. 1988 IEEE Int. Conf. Robot. Automat., pp 427–430, Philadelphia, Pennsylvania.
- Ng WS, Phee SJ, Seow-Choen F (2000) Robotic endoscope and an autonomous pipe robot for performing endoscopic procedures. United States Patent, Patent Number 6,162,171.
- Okuzaki H, Osada Y (1994) Effects of hydrophobic interaction on the cooperative binding of a surfactant to a polymer network. *Macromolecules*, 27(2):502–506.
- Phee SJ, Ng WS, Chen IM, Seow-Choen F (1997a) Development of new locomotive concepts to be used in automation of colonoscopy. In 9th Int. Conf. for BioMedical Engineering, pp 87–92, Singapore.
- Phee SJ, Ng WS, Chen IM, Seow-Choen F, Davies BL (1997b) Locomotion and steering aspects in automation of colonoscopy. *IEEE Engineering in Medicine and Biology Magazine*, 16(6):85–96.
- Shahinpoor M (1996) Ionic polymeric gels as artificial muscles for robotic and medical applications. *Int. Journal Sci. Technol.*, 20:89–136. Transaction B.
- Slatkin AB, Burdick J, Grundfest W (1997) The development of a robotic endoscope. In Experimental Robotics IV, The 4th Int. Symp., pp 161–169, Berlin. Springer-Verlag.
- Treat MR, Trimmer WS (1997) Self-propelled endoscope using pressure driven linear actuators. United States Patent, Patent Number 5,595,565.
- Vincent CA (1998) Modern batteries. Edward Arnold Pub., London, U.K., 2nd edition.

Advanced Sonar Sensing

Lindsay Kleeman

Intelligent Robotics Research Centre, Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, Australia,
lindsay.kleeman@eng.monash.edu.au

Abstract. Robotics research is heavily dependent on fast, accurate, reliable and cheap sensors. Sonar sensing can fulfil these requirements in air and underwater environments. Moreover sonar physics provides robotics researchers with a natural selection capability for landmark detection in navigation problems. This paper presents new sonar results that allow high-speed accurate measurement and classification suitable for moving platforms that has been combined with interference rejection to allow multiple sonar systems to co-exist.

1 Introduction

Sonar sensing is common-place in robot applications, however advanced sonar sensing is not widely applied. What is meant by advanced sonar? To answer this we need to look firstly a sonar that is not advanced, such as the Polaroid Ranging Module (PRM) (Polaroid Corp., 1982). In its commonly applied form, PMR offers range to 10 metres to the first echo above a threshold¹ within a 20 degree beamwidth that changes with range and depends on target reflectivity. The user knows little about the target, the angle to the target, the strength of the echo or whether the echo comes from the same sonar system at all. Advanced sonar can accurately determine angle, target classification, target strength and whether that sonar system owns the echo it received – that is to allow interference rejection. Accurate range and bearing measurements of multiple targets have been achieved in (Jorg, Berg, 1998; Kleeman, 1999; Kleeman, Kuc, 1995; Peremans, Audenaert, Campenhout, 1993), interference rejection reported in (Borenstein, Koren, 1995; Jorg, Berg, 1998; Kleeman, 1999), and targets classification in (Chong, Kleeman, 1999; Kleeman, Kuc, 1995; Peremans et al., 1993). Knowing the shape, or class, of a target assists in robot localisation and mapping applications. It enables prediction of how the target will appear from different sensor positions, and it simplifies associating sensed environmental features to a map and building maps themselves. Mistakes in association can lead to persistent gross errors. Properties of acoustic wave propagation and the structure of indoor environments has lead sonar researchers to the adoption of three classes of target: plane, 90° concave corner and convex edge. The work in (Kleeman, Kuc, 1995) provides a proof that the minimum requirements to classify targets into these categories are two transmitter positions and two receivers positions. The transmitter

¹ Discrete gain increases with time in the PRM pre-amplifier combined with integration before thresholding can complicate modelling.

positions need not be distinct from the receiver positions. Sonar relies on insonifying the environment in a cyclic fashion, so the energy emitted in one cycle has dissipated before a new cycle is commenced². In pulse-echo sonar this means a new pulse is not transmitted before there is a possibility that a previous pulse can be detected by a receiver. Similarly in CTFM (continuous time frequency modulated) sonar a new frequency sweep is not commenced until the start of the previous acoustic sweep has died out. Existing sonar target classification methods require multiple cycles to obtain coordinates of virtual images of a transmitter in two different positions. This requires moving a single transmitter, or incorporating two transmitters into the sensor. Either way, in the interval between the two cycles there can be significant air movement, which contributes to errors in the measurement, and reduces the reliability of the eventual classification. If the interval can be reduced, reliability should be improved. Additionally, the latency before the target can be classified will be reduced. This paper discusses new results that reduce this latency, and allow classification of a target as a corner, plane, edge, or unknown. The classification is compressed into a single processing cycle by firing two transmitters nearly simultaneously. Indeed, the precise separation identifies the sensor, thus providing interference rejection using double pulse coding – a technique presented for interference rejection alone in (Kleeman, 1999). This paper shows that there is a natural synergy between classification and interference rejection. Moreover the signal processing can be performed at cycle rates (eg 30 Hz) by a DSP implementation, allowing on-the-fly sonar classification and interference rejection. Advanced sonar is now more mature and should become a commercial reality, resulting in wider applications in research papers on robotics. However, single sensor modalities are rarely the answer to robotics problems and this paper highlights some unique characteristics that sonar offers compared to other range sensors such as laser techniques, stereo vision and tactile sensing. This paper is organised as follows: Section 2 discusses how a DSP sonar sensor can be implemented to achieve an advanced sonar design. Techniques for extracting accurate range and bearing measurements are described in section 3, while section 4 illustrates how sonar can naturally extract sparse landmarks – a useful property of sonar when combined with other sensors, such as vision and laser systems. Section 5 discusses the important problem of sonar discrimination between targets that are closely spaced. Section 6 addresses the problem of interference rejection and presents results of experimental verification of difference approaches. On-the-fly classification approaches are outlined in section 7. Extensions of the author's air sonar to underwater are given in section 8. Conclusions are then presented.

2 Sensor Design

In this section, the design of a DSP based sonar sensor is discussed. The sonar sensor is built from commercially available 7000 series Polaroid transducers (Polaroid Corp., 1987) with their front grille removed – this significantly reduces pulse lengths

² Some researchers are examining tagging transmissions to allow one cycle to begin before the previous has completed.

since the grille causes reverberation within the transducer. The sensor includes custom designed digital and analogue electronics using readily available integrated circuits, and a digital signal processor (DSP). The signal processing is implemented within the sensor which communicates the results to a host computer via high speed serial communication. The sensor measures 150 by 100 by 70 mm and is powered by a single 5 Volt supply. Results on range and bearing accuracy, once the speed of sound is calibrated, are reported in (Kleeman, Kuc, 1995) and vary with air conditions. Typical office conditions give range error standard deviations of 0.2 mm and bearing error standard deviations of 0.1 degrees at 3 metres range. The sensor is capable of detecting walls to 8 metres, but echo sampling is stopped at 5 metres to increase speed.

2.1 Transducer Arrangement



Fig. 1. The DSP sonar sensor showing transmitters T1 and T2 and receivers R1 and R2.

The transducers are arranged in a square shown in Figure 1, where adjacent transducers are just 40 mm apart. An important feature of sonar is that time of flight errors are well correlated in time and space (Kleeman, 1999; Kleeman, Kuc, 1995). By placing two receivers close together, very accurate measurement of bearing can be achieved despite the short baseline (Kleeman, Kuc, 1995), because the bearing calculation depends critically on the difference between the two times-of-flight from a single transmitter to the two receivers. The error in this difference is typically much smaller than the straight time of flight errors since these are highly correlated. Additionally placing the two transmitters close together produces highly correlated errors for all the four time-of-flight measurements of a target – that is T1 to R1, R2 and T2 to R1, R2. This is important because the dominant factor in determining the class of a target is the difference between the two measured bearings (Heale, Kleeman, 2001). Classification of targets with closely spaced transmitters has been

achieved out to 5 metres (Heale, Kleeman, 2001). In summary the receiver and transmitter close spacing exploits the spatial correlation of errors in time of flights to achieve accurate bearing estimation and reliable classification.

2.2 Electronics and Processing Hardware

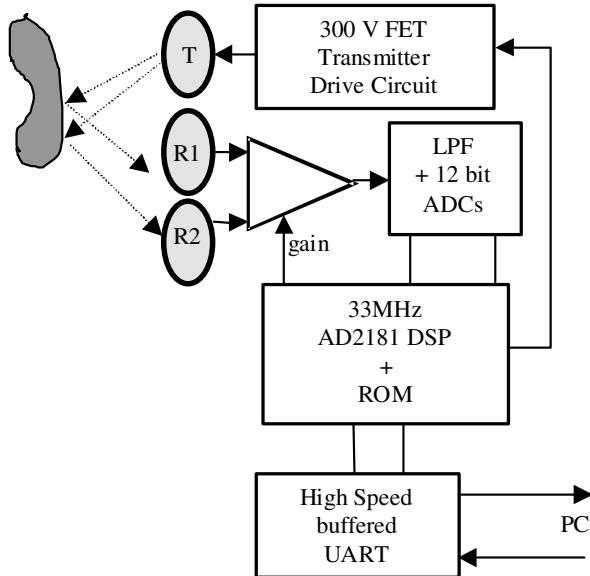


Fig. 2. DSP sonar hardware block diagram.

Referring to Figure 2, the received signals are amplified, low pass filtered and digitised at 1 MHz and 12 bit precision, then processed on an Analog Devices 33 MHz ADSP2181. The DSP also generates the transmit waveforms and communicates with the host via an external UART. A pulse is fired from the right transmitter first, and rapidly followed from the left (200 μ s delay is typical). Echoes are digitised and processed on a DSP, yielding up to four arrival times for reflections from each target.

2.3 Interface

The sensor communicates with a host computer via a set of commands. These commands enable access to all levels of signal data from the sonar sensor, from the received pulse data samples up to the target range, bearing and class. Low level access is implemented by the DSP, sending to the host, sonar data structure addresses within the DSP memory space, so the host can interrogate intermediate results. In

normal operation, packets of target information are send periodically with the sensor firing continuously.

2.4 Time stamping and synchronisation

In real time sensing applications sensor measurements themselves are of little value without knowing the time they were collected. This is for two reasons – the sensor or platform may be moving so that sensor data needs to be fused with other navigation sensors, and the targets being sensed can be dynamic. Facilities on the sonar sensor are provided for synchronising the sonar local time with a host time and providing a time stamp with every measurement set.

3 Range and Bearing Measurements

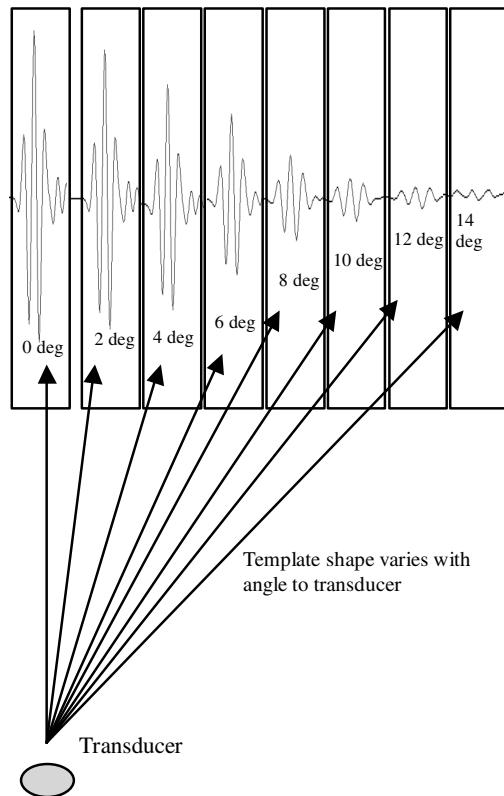


Fig. 3. Template pulse shapes as a function of angle.

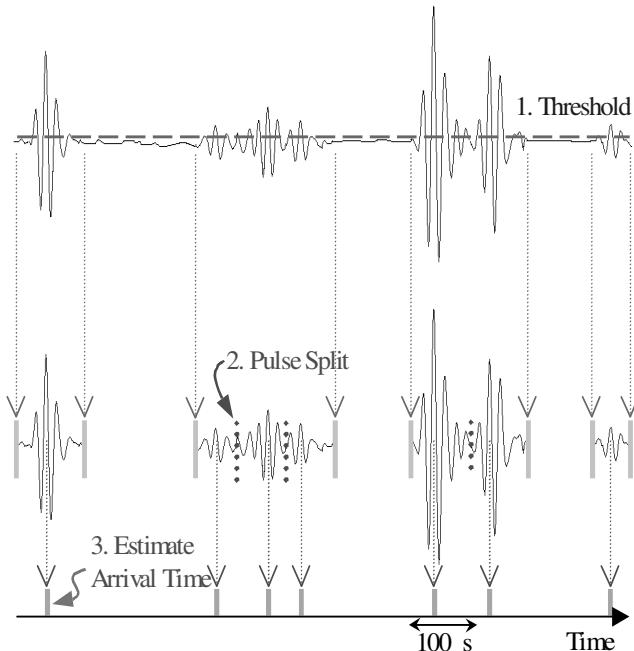


Fig. 4. DSP processing steps to extract arrival times.

The aim of the sonar data processing is to extract the arrival times of the echo pulses. From the arrival times, range and bearing to objects can be estimated by using the speed of sound and receiver geometry. Arrival times are estimated using a technique known as matched filtering first used in RADAR systems. Matched filtering obtains the arrival time by cross correlating the received echo with an echo template stored in the sensor. A template is a noise free pulse shape computed offline. The template shape depends on the angle of arrival to the transducer as shown in Figure 3. This dependency on angle can be accurately modelled by using convolution with the impulse response of a circular transducer (Kleeman, Kuc, 1995). The template is shifted across the echo to find the maximum correlation. By fitting a parabola to the maximum three correlations and their shift times, a very accurate arrival time estimate is obtained (Kleeman, Kuc, 1995). The software to process the sonar echoes has been optimised for speed. Careful design of the sonar front end electronics has led to the high signal to noise ratio and consequently the data processing is significantly simplified. In particular, matched filtering is performed only on sections of the echo signal deemed to be “pulses” by virtue of exceeding a threshold. The threshold is set to 7 noise standard deviations to avoid spurious triggering. Other sonar systems reported in the literature (Jorg, Berg, 1998; Peremans et al., 1993) perform expensive matched filtering on the entire received waveform since pulses cannot be separated reliably from noise as can be done in this system.

A pulse commences 30 samples before exceeding the noise threshold and ends 30 samples after falling below the threshold. A further process of “pulse splitting” is applied to separate nearly overlapping pulses by searching for a local maximum within a 60 sample sliding window. The DSP software is organised into two stages. During the first stage, highly optimised assembly code performs on-the-fly processing of the samples from the two receivers to extract discrete pulses that exceed the noise floor. The second stage processes the extracted pulses with C code to partition closely spaced yet separable pulses and to extract arrival times using matched filtering. The processing steps are shown on an example echo signal in Figure 4. Stage one has a main program and a timer interrupt routine that runs every microsecond. The timer interrupt routine fetches the next 12 bit ADC samples from the two receiver channels and places them into a circular buffer. The interrupt routine is also responsible for generating the transmit pulse and receiver gain changes. The main program runs in a loop where each iteration processes the block of data acquired since the previous iteration. The two channels are processed independently through four stages: DC bias removal, thresholding, aggregation and storing into a pulse data structure.

3.1 Association

In order to derive the bearing angle to physical targets, pulse arrival times must be associated between the left and right receiver channels. Ambiguities are possible in this process when there are many closely spaced pulses. To guard against this, a conservative approach is adopted and possible ambiguities actively sought out and discarded. The effect of an incorrect association between two pulses is that the bearing angle is grossly in error. Every pulse extracted from the left channel is compared with every pulse from the right. An association is declared reliable if the following conditions are met:

- Arrival times are consistent with the receiver physical spacing, $\text{SEP} (=40 \text{ mm})$ – that is arrival times differ by less than $\sin(\text{max_angle}) * \text{SEP}/\text{speed_sound}$.
- Pulse amplitude ratio is between 0.5 and 2.0.
- Both correlation coefficients over 95 percent.

If only the last condition is not met, the association is flagged as unreliable. Range and bearing of all associations are calculated (Kleeman, Kuc, 1995) and called targets. If any pulse is associated with more than one pulse on the other receiver, all associations involving these pulses are demoted to unreliable. Pulses that are not associated with any pulse are retained and are assumed to correspond to targets in the pointing direction of the receiver and are marked extremely unreliable³. These targets combined with amplitude information across several measurements taken at different directions can give some indication of the roughness and position of a surface (Bozma, Kuc, 1991).

³ Essentially the same information that the Polaroid Ranging Module provides except amplitude is known.

4 Natural Selection of Landmarks

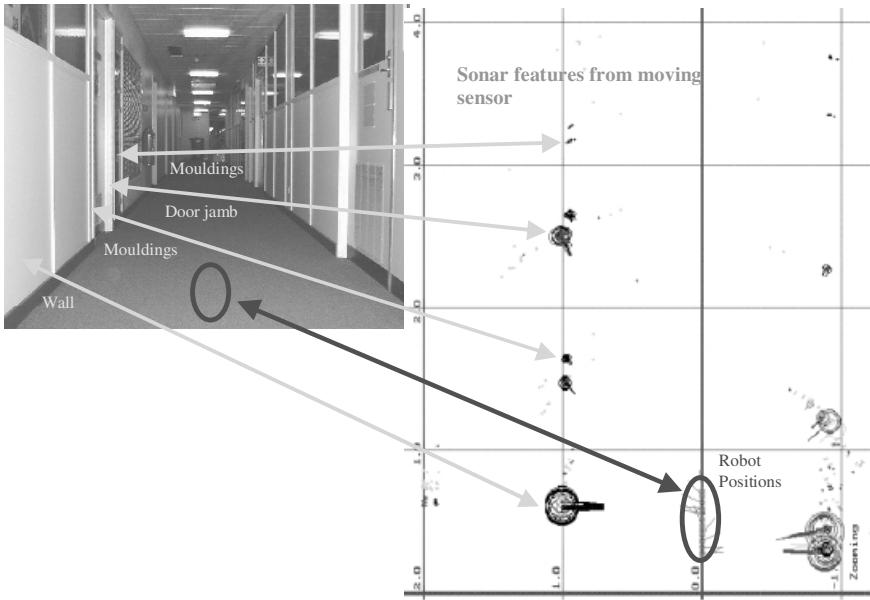


Fig. 5. Examples of natural landmark extraction with advanced sonar.

In Figure 5, the sonar sensor is mounted on a mobile robot with panning mechanism and sent down a corridor. Whilst moving, the sonar performs “wind-screen wiper” scans of the environment. Using odometry position information, sonar range/bearings and pan angles, targets are plotted in Figure 5. The radius of the circle representing a target is proportional the echo amplitude measured by the sonar and indicates a measure of reliability of the target. There are several important characteristics of the resulting map:

- Sparse distribution of features – low density ranging reduces map complexity for navigation.
- Only useful natural landmarks appear, such as the orientation and position of walls, positions of corners, edges or doorways.
- Reliable targets can be identified by their sonar amplitude and pulse shape integrity.

5 Discrimination

The ability of a sonar system to separate targets closely spaced in range is important. Good discrimination allows targets, for example, with quite difference bearings but

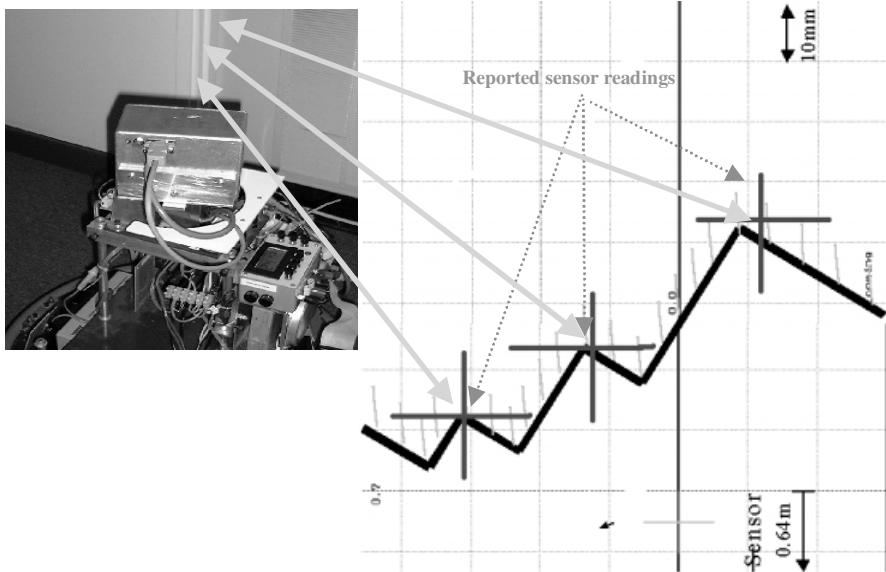


Fig. 6. Example showing the limit of discrimination with 10 mm separated targets.

similar ranges to be distinctly sensed. Fine discrimination can be achieved using short pulses or pulse coding techniques that allow extraction of overlapping pulses. The former approach is adopted with the DSP sonar described here since it allows faster processing times. Also, to the author's knowledge, the current system achieves results unsurpassed in the literature. Short pulses are achieved by exciting the transducer with a high energy pulse and a shorter out of phase pulse to dampen ringing effects. An example of the smallest possible discrimination achievable by the sonar system is shown in Figure 6.

6 Interference Rejection

In order to reject interference, one needs to be able to identify your own echoes. Some form of encoding of the transmitted pulse can be deployed to this end. In this work the simplest compatible approach is taken and this is to transmit two identical pulses and use the time separation as the encoding information. Received pulses are examined and checked for this same time separation, subject to Doppler variations due to robot or target motion (Kleeman, 1999). Other more sophisticated coding techniques exist but result in longer pulses or additional processing overhead. Two approaches have been used to validate that pulse pairs are properly separated and the same shape:

1. (Kleeman, 1999) validates echoes by checking that the maximum difference between pulse waveforms, taken sample by sample, must fall below a threshold

to be validated. Doppler-like effects slightly change the separation of returns from a moving target, and since the robot itself moves, many returns exhibit this effect. To allow for this variation, the two returns may be shifted slightly relative to each other and tested again.

2. (Heale, Kleeman, 2000, 2001) uses a similar algorithm. The chief difference is the use of correlation rather than difference, and the result is rather like using the first return to define a matched filter that will identify the second.

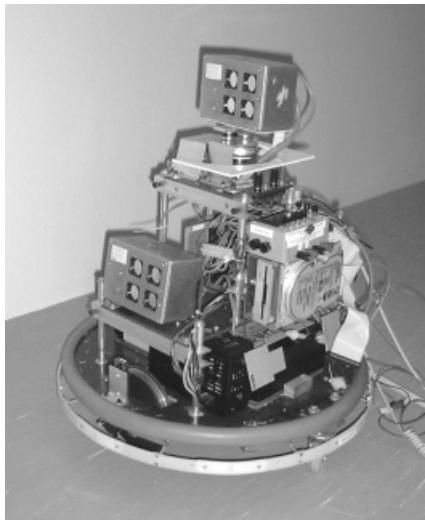


Fig. 7. Interference experiments on a robot with two sensor firing at the same wall simultaneously.

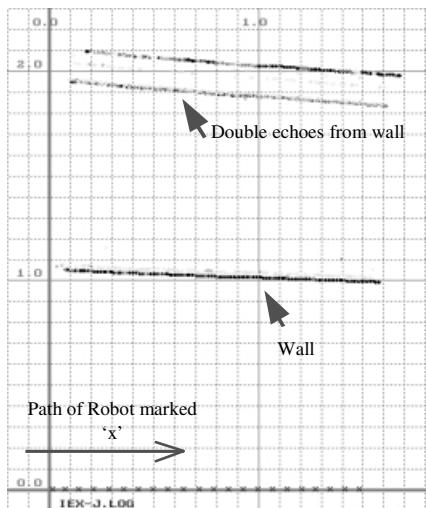


Fig. 8. Interference from a double pulse source is rejected from another sonar using a different double pulse spacing.

A spin-off from the second approach is that the Doppler shift contains relative velocity information about the target and robot that may be useful. The double pulse coding approach has been tested using experiments that introduce deliberate interference. Figure 7 shows a robot with two sonar systems that fire simultaneously – the lower one is the interfering source and the upper one is used to construct a map of the wall. Figure 8 shows results from double pulse coded interference with a difference spacing to the double pulse coding used in the map building sensor. Note that the wall floor corner is seen as a phantom target beyond the expected wall position and double reflections also add other phantom targets. Note the lack of spurious readings in the map. Compare Figure 8 with Figure 9 where the same spacing is used by the interference and the mapping sonar. The interference is clearly present in the results. Figure 10 shows the results when the double pulse spacing is varied from cycle to cycle randomly for both sensors. Most interference is rejected, but not all. The remaining interference is due to coinciding random spacings and also environ-

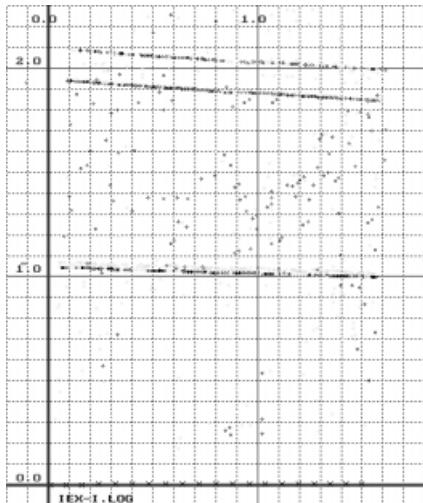


Fig. 9. Interference is not rejected since both sensors use the same double pulse spacings.

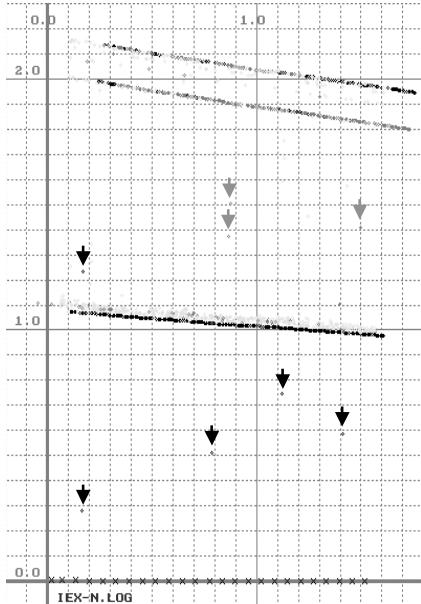


Fig. 10. Random double pulse spacing interfering with random double pulse spacing – most but not all interference is rejected.

mental faking effects where a single transmitted interference pulse generates two echoes with a spacing coinciding with the random spacing. Since the interference is not synchronised to the mapping sensor, these occurrences do not repeat at the same position cycle to cycle.

The advantage of randomly spaced double pulse coding is that no a priori code negotiating needs to be established between sensors. This allows multiple sensors to operate in the same environment without any communication required between them.

7 Classification

Target classification into planes, corners and edges, as shown in Figure 11 has been performed in (Kleeman, Kuc, 1995) in two sensor cycles by alternatively firing one transmitter and then the other. The basis for the classification can be explained simply by an analogy with virtual images in mirrors. If one looks into a plane mirror and then your left eye appears on the left. Similarly in a sonar system the left transmitter (T2 in Figure 1) will be observed to the left (in the bearing measurement) of the right transmitter T1 observed angle. However the situation is reversed when looking into a right-angled mirror – that is the left eye is observed on the right. If one looks at a polished metal chair leg (high curvature specular surface), your left

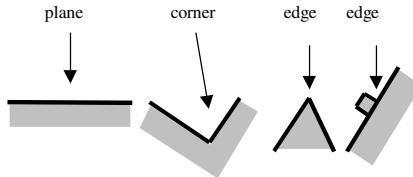


Fig. 11. Target types that sonar can classify.

and right eye images are compressed to appear to be in the same position approximately – that is the angle between left and right eyes is zero. Sonar classification can exploit the difference in bearing angles to the two transmitters to classify: positive difference = plane, Recently, double pulse coding has been combined with classification to produce a classifying measurement in one sensing cycle (Heale, Kleeman, 2001). Transmitter T2 is fired a known short time (eg 200 μ sec) after T1 and echo arrival times are assembled from the two receivers. Given vertical planes, corners and edges, as commonly found in indoor environments, the distance of flight T1 to R2 is the same as T2 to R1. This property is exploited to determine the double pulse coding and hence transmitter identities (ie T1 or T2) of the received pulses. Therefore classification can be performed in one measurement cycle. Measurement and classification of targets has been implemented at 27 Hz using the DSP sonar system in Figure 1 (Heale, Kleeman, 2001).

8 Some Underwater Results

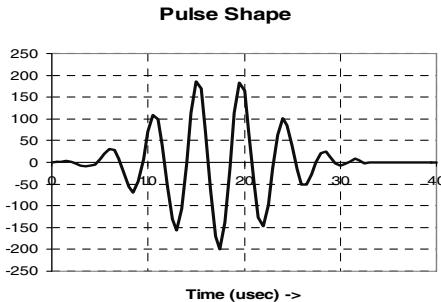


Fig. 12. Underwater echo pulse shape for maximum amplitude and minimum pulse length.

Current work has been to implement the DSP sonar in an underwater environment for autonomous underwater navigation. A 200 kHz centre frequency system is under development with custom wide bandwidth transducers supplied by SensorTech in Canada. Three receiver boards based on a DSP processor and 2 MHz

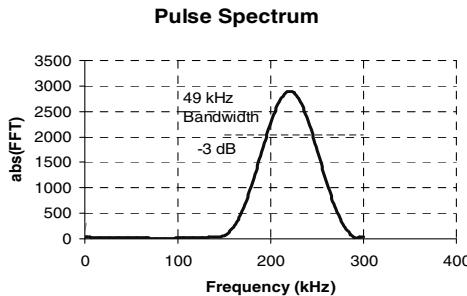


Fig. 13. Spectrum of pulse from Figure 12.

sampling ADCs have been completed. The echo shape is similar to the air based system and is shown in Figure 12 as captured by the DSP system and the derived spectrum shown in Figure 13. The initial aim is to achieve highly accurate range and bearing measurements to specular targets where simple geometric models exist. These targets will initially be man-made reflectors, such as existing channel markers, that act as landmarks.

9 Conclusions

This paper has presented results illustrating the capability of advanced sonar. Significant improvements in sonar research have been made in terms of range and bearing accuracy, speed, discrimination, interference rejection and classification. Recent work has been highlighted that combines interference rejection with classification in one sensor cycle, allowing robust on-the-fly applications involving classification. Future work needs to concentrate on the fusion of advanced sonar with other sensor modalities in way that exploits characteristics of each. Acknowledgments Funding is acknowledged from the Australian Research Council. The design, construction and testing skills of Steven Armstrong are gratefully acknowledged. Andrew Heale's highly optimised contributions to the DSP sonar system are cheerfully acknowledged.

References

- Borenstein J, Koren Y (1995) Error eliminating rapid ultrasonic firing for mobile robot obstacle avoidance. *IEEE Trans. Robotics & Automation*, 11(1):132–138.
- Bozma O, Kuc R (1991) Characterizing pulses reflected from rough surfaces using ultrasound. *Journal Acoust Soc of America*, 89(6):2519–2531.
- Chong KS, Kleeman L (1999) Feature-based mapping in real, large scale environments using an ultrasonic array. *Intern Journal Robotics Research*, 18(1):3–19.
- Heale A, Kleeman L (2000) A sonar sensor with random double pulse coding. In: Australian Conf on Robotics & Automation, pp 81–86, Melbourne.

- Heale A, Kleeman L (2001) Fast target classification using sonar. In IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems, pp 1446–1451, Hawaii, USA.
- Jorg K, Berg M (1998) Mobile robot sonar sensing with pseudo-random codes. In Proceedings IEEE Conf. on Robotics & Automation, pp 2807–2812, Belgium.
- Kleeman L (1999) Fast and accurate sonar trackers using double pulse coding. In IEEE/RSJ Intern. Conf. on Intelligent Robots & Systems, pp 1185–1190, Kyongju, Korea.
- Kleeman L, Kuc R (1995) Mobile robot sonar for target localization and classification. International Journal of Robotics Research, 14(4):295–318.
- Peremans H, Audenaert K, Campenhout JMV (1993) A high-resolution sensor based on tri-aural perception. IEEE Trans on Robotics and Automation, 9:36–48.
- Polaroid Corp. (1982) Ultrasonic range finders.
- Polaroid Corp. (1987) Data sheet for series 7000 transducer. Ultrasonic Components Group, 119 Windsor St., Cambridge, MA.

Mechano-Media that Transmit Kinesthetic Knowledge from a Human to Other Humans

Yasuyoshi Yokokohji¹, Yoshihiko Sugawara^{1,2}, Junji Kinoshita^{1,3}, and Tsuneo Yoshikawa¹

¹ Department of Mechanical Engineering, Kyoto University, Kyoto 606-8501, Japan

² Currently with DENSO CORPORATION

³ Currently with Hitachi, LTD.

Abstract. We have been developing a visual/haptic interface to virtual environments, called a WYSIWYF Display, which ensures spatially/temporally consistent visual/haptic sensation. Three key components are (i) vision-based tracking for correct visual/haptic registration (ii) blending live video and computer graphics (CG) images by a chroma-keying technique, and (iii) encountered-type haptic rendering that provides real free and real touch sensations.

In this paper, the recent progress on the above key components are presented. First, accurate image overlay technique by using vision and accelerometers is shown. Then, a path planning for encountered-type haptic devices that render multiple virtual objects is presented. A virtual control panel of automobiles was developed to demonstrate the proposed path-planning algorithm.

The developed visual/haptic interface system can be regarded as a kind of mechano-media, a new concept extending the framework of teleoperation, which transmit kinesthetic knowledge from a human to other humans.

1 Introduction

Haptic interfaces have been recognized as important input/output channels to/from the virtual environment (Burdea, 1996). Usually a haptic interface is implemented with a visual display interface such as a head-mounted display (HMD) or a stereoscopic screen, and the total system is configured as a visual/haptic interface. Correct spatial registration of the visual interface with the haptic interface is not easy to achieve, however, and has not been seriously considered. For example, some systems have a graphics display simply located next to the haptic interface – resulting in a “feeling here but looking there” situation, as shown in Fig. 1(a). Visual/haptic interfaces are expected to be useful for training of visuo-motor skills such as medical operations. However, poor visual/haptic registration may cause an inter-sensory conflict that leads to a wrong adaptation and a skewed sensory rearrangement (Groen, Werkhoven, 1998).

Ideally, a visual/haptic interface should be configured in such a way that “what you see is what you feel,” as is the case in real life situations. Hereafter we refer to such an ideal situation as “WYSIWYF,” in analogy with the term “WYSIWYG,” or “what you see is what you get,” used in the context of document creation. Several years ago, we proposed a new method to realize WYSIWYF for visual/haptic interfaces, with a potential application area being the training of visuo-

motor skills (Yokokohji, Hollis, Kanade, 1996b) (Yokokohji, Hollis, Kanade, 1999). Three key components of the proposed method are (i) vision-based tracking, (ii) blending live video and computer graphics (CG) images by a chroma-keying technique, and (iii) introducing an encountered-type haptic interface controlled by a position-command-based haptic rendering algorithm. The user's hand can "encounter" the haptic device exactly when/where his or her hand reaches the virtual object in the visual display. We referred to a visual/haptic interfaces realizing WYSIWYF as a "WYSIWYF Display." Figure 1(b) illustrates the concept of WYSIWYF display. Based on the proposed method, we also developed a first prototype of WYSIWYF display as shown in Fig. 1(c).

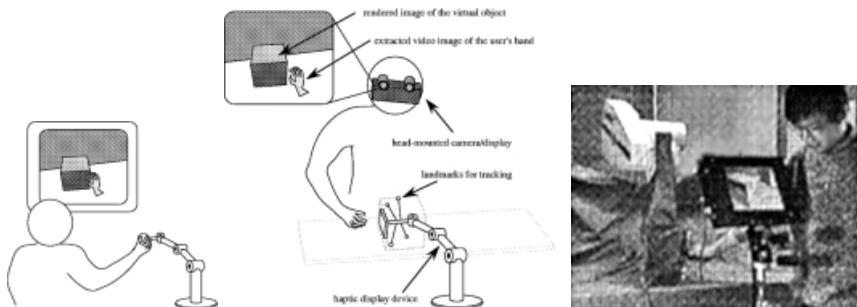


Fig. 1. WYSIWYF display. (a) "Feeling here, but looking there" or non-WYSIWYF situation. (b) Conceptual sketch. (c) Prototype WYSIWYF display.

Although the first prototype could show the validity of the proposed method, we found many problems to be solved. First of all, the vision-based tracking was not perfect. As a result, real image overlaid on the synthetic image was misaligned, resulting in an imperfect blended image. Secondly, the encountered type haptic device could render only one virtual object. In this paper, the recent progress on the key components of WYSIWYF display are presented. First, accurate image overlay technique by using vision and accelerometers is shown. Then, a path planning algorithm for encountered-type haptic devices that render multiple virtual objects is presented. A virtual control panel of automobiles was developed to demonstrate the proposed path-planning algorithm.

The developed visual/haptic interface systems can be regarded as a new kind of media that transmit kinesthetic knowledge, such as motor skills, from a human to other humans. In the last part of this paper, we propose a new concept called mechano-media, extending the framework of teleoperation, that could be a new research direction of robotics.

2 WYSIWYF display

2.1 Prototype system

Figure 1(c) shows an overview of the prototype system developed in 1996. Although a head-mounted camera/display would be ideal for WYSIWYF as shown in Fig. 1(b), we decided to use an existing LCD panel (10 inch TFT color) for the first prototype. A color CCD camera was attached to the back plane of the LCD panel. The LCD/camera system is mounted on a movable stand so that the user can move it around to change his or her viewpoint. A Unimation PUMA 560, a 6-DOF industrial robot, is used for the haptic device. Four fiducial points (small incandescent lamps covered by translucent lenses) are attached to the endpoint of the robot for vision-based tracking.

The CCD camera behind the LCD panel captures the image including the haptic device and the user's hand. From the location of the fiducial points in the image plane, position and orientation of the camera/display system with respect to the haptic device is recovered and CG images are overlaid on the haptic device. The working environment including the PUMA was covered by blue cloth so that the user's hand image can be easily extracted by chroma-key.

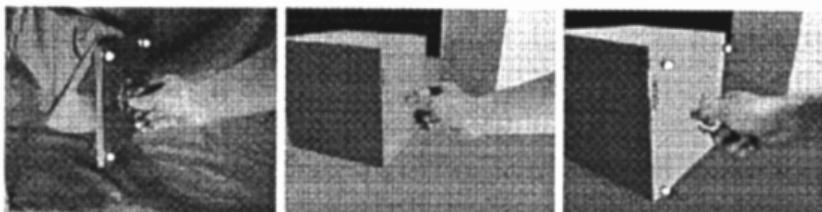


Fig. 2. Results of static/dynamic registration. (a) Original video scene. (b) Final blended image in static scene. (c) Dynamic registration error.

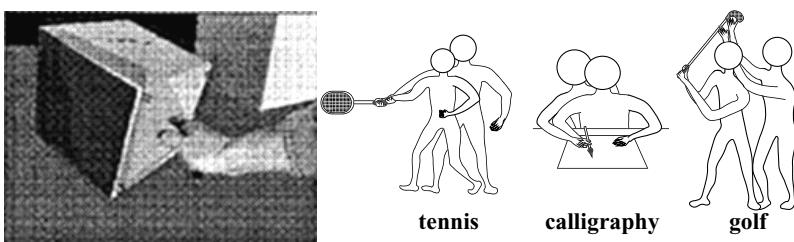


Fig. 3. Motor skill training. (a) A simple example with the WYSIWYF display. (b) Bodily instruction of motor skill.

A simple frictionless virtual environment was built, where a cube with edges of 20 cm is placed on top of a flat table. Figures 2(a)(b) show the video blending

Table 1. Estimated performance of the visual display component

Mode	Frame rate	Latency	Alignment error
Camera fixed	50 Hz	negligible (≤ 10 msec)	4 pixels*
Tracking by SIRIUS	3 Hz	0.9 sec	2 pixels
Tracking by TRV	20 Hz	150 msec	50 pixels

* as long as the motion is within 20 cm from the initial position

process. As shown in Fig. 2(b), a CG image is accurately overlaid on the haptic device when the camera/display system is stationary. Figure 3(a) shows a simple example of training. The user is trying to follow a pre-recorded motion demonstrated by an expert beforehand, which is shown by a translucent cube. A position servo can guide the user to the reference motion, as though a human instructor takes a trainee's hand and teach him step by step as shown in Fig. 3(b). Unlike just watching a video, the trainee can feel the reaction forces from the virtual environment while trying to follow the reference motion. The servo gain can be adjusted according to the trainee's progress; for example, starting from a high gain and adjusting it to lower gains as the trainee's performance improves. For more discussion about training, see (Henmi, Yoshikawa, 1998; Yokokohji, 1998; Yokokohji *et al.*, 1996a).

2.2 Remaining problems

With the first prototype, the user can manipulate a virtual object quite realistically with the real hand image that is well aligned to the virtual object and with crisp force feedback by an industrial robot. Therefore, the first prototype system could show the validity of the proposed approach. However, we found several remaining problems to be solved.

Image overlay The most serious problem of the first prototype was system latency. Table 1 summarizes the estimated performance in terms of frame rate, end-to-end latency (total latency from measuring data until displaying the final image), and alignment error. In camera fixed mode, the vision-based tracking is used only for the initial registration. After this one-shot registration at the initial position, the user is not allowed to move the camera/display system around. As a matter of fact, we reluctantly introduced this camera fixed mode in order to avoid annoying low frame rate and large latency in tracking mode by SIRIUS, a video board for SGI PowerOnyx, which was used for rendering CG images in our prototype system. Figures 2(b) and 3(a) were taken in this camera fixed mode. Vision-based tracking using SIRIUS ended up with low frame rate and large latency, far beyond the acceptable levels.

To allow the user to move the camera/display system anytime, we introduced “Tracking Vision”, a motion tracking system made by FUJITSU CO.,LTD. Figure 2(c) shows a demonstration in tracking mode by TRV. This figure shows an instant when the user moves the cube downward. Introducing the Tracking Vision, the frame rate was recovered to 20 Hz. Since the user’s hand image is displayed through the chroma-key circuitry in SIRIUS with almost no delay, the delay of the displayed images is noticeable for the user. The estimated delay was about 150 msec, which results in 50 pixels misalignment when the user moves the device at approximately 30 [cm/s].

The vision-based tracking can provide good enough accuracy for the static registration. However, the end-to-end system delay is still large even after we introduced the fast video tracker and the tracking mode was not practical. Dynamic registration error due to the end-to-end delay is a fundamental problem for head tracking in VR and AR applications (Azuma, 1997). In the next section, we will introduce our improvement on the dynamic registration by using hybrid vision/inertial tracking approach.

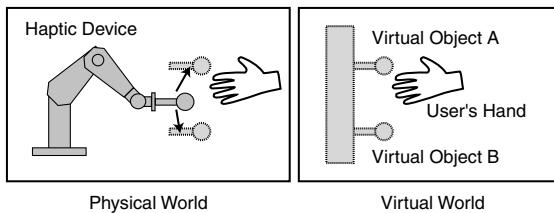


Fig. 4. Rendering multiple objects by an encountered-type haptic device.

Rendering multiple virtual objects by a single encountered-type haptic device
 Encountered-type haptic device was first proposed by McNeely (McNeely, 1993). Tachi *et al.* (Tachi et al., 1994) proposed a similar idea independently. As shown in Fig. 1(b), the encountered-type device is not held by the user all the time. Instead, the device stays at the location of the virtual object and waits for the user to *encounter* it. Therefore, the encountered-type haptic rendering can provide real free and real touch sensations to the users.

In the first prototype, we assumed that there was only one virtual object. In this case, the device could be simply placed at the location of this object. When multiple objects need to be rendered by a single haptic device as shown in Fig. 4, however, the device should be placed in advance at the location of a target object that the user is going to reach. In section 4, we will show our improvement on the encountered-type haptic device.

3 Accurate Image Overlay on HMDs by Hybrid Vision/Inertial Tracking

3.1 Image overlay

In augmented reality (AR) applications, where synthetic images are overlaid on real scenes as shown in Fig. 5, head tracking should be more accurate than for virtual reality (VR) applications, because users may easily detect even a small misalignment between the synthetic image and the real image (Azuma, 1997).

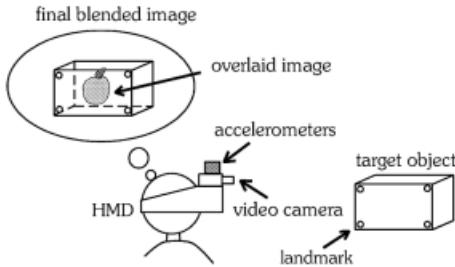


Fig. 5. Augmented reality by image overlay.

In AR applications, see-through HMDs are used for image overlay. Since video see-through HMDs display real scenes captured by video cameras, they are compatible with the vision-based tracking approach. If the target scene includes some points or lines that could be landmarks, it is possible to recover the camera pose (position and orientation in 3D space) with respect to the target object by tracking these points or lines (Uenohara, Kanade, 1995).

The vision-based approach can achieve more accurate static registration than using emitter-based sensors such as magnetic trackers (Azuma, 1997). However, video images are usually taken at 30 [Hz] (video rate) and the end-to-end system delay is not negligible. Hence, if the user moves his/her head quickly, the overlaid image is misaligned from the real image and even swings around (dynamic registration error) (Azuma, 1997). A quick head motion may also cause the vision-based tracker to lose the landmarks.

WYSIWYF display shown in section 2 also used vision-based tracking and image overlay. In this section, we will show our improvement of accurate image overlay using hybrid vision/inertial tracking.

3.2 Hybrid vision/inertial tracking

To achieve accurate image overlay on HMDs, we proposed a method using vision and accelerometers as shown in Fig. 5 (Yokokohji, Sugawara, Yoshikawa, 2000). With vision-based tracking, landmark information is obtained at 30 [Hz], while the acceleration is measured at much higher sampling rate 1000 [Hz]. The low rate

vision-based tracking compensates the bias caused by integrating the accelerations, while the high rate acceleration compensate the delayed vision information with slow update frequency. This kind of complementary use of vision and inertial sensors such as gyro and accelerometer is called hybrid vision/inertial tracking.

The proposed method uses the extended Kalman filter (EKF) framework. Dickmanns et al. (Dickmanns, Mysliwetz, Christians, 1990) used this framework for vision-based autonomous vehicle guidance. Setting appropriate state variables, this recursive state estimation allows us to predict future states. In our system, acceleration information is also used for predicting the head pose more precisely than using vision information alone in the EKF. The accurate prediction also makes the vision-based tracking robust. In addition, acceleration information is used to predict the head pose further until the rendering timing in order to compensate the end-to-end system delay.

Several methods using hybrid vision/inertial tracking have been proposed so far (Azuma, Bishop, 1994; You, Neumann, Azuma, 1999). Our approach is unique in a sense that both linear and angular accelerations are measured. More importantly, we could show a solid result considering some implementation tips about synchronization and timing issues (Yokokohi et al., 2000).

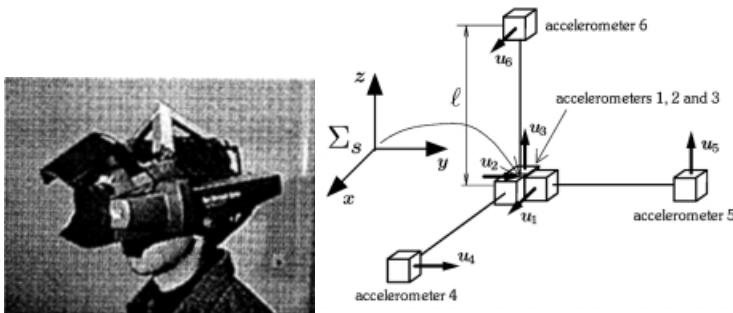


Fig. 6. Prototype video see-through HMD. (a) Overview. (b) Accelerometer configuration.

3.3 Prototype system

We used a HMD by SHIMADZU Co. (See-Through Vision STV-E). We added a small CCD camera (IK-UM40 by TOSHIBA Co.) and modified the original HMD to a video see-through type. Figure 6(a) shows an overview of the prototype HMD with accelerometers. Six accelerometers were configured as shown in Fig. 6(b) so that both linear and angular accelerations can be measured.

The target object is a paper box ($40 \text{ [cm]} \times 30 \text{ [cm]} \times 20 \text{ [cm]}$) with four landmarks, each one at each corner of the frontal face. The box is located about 2.5 [m] apart from the user. The overlaid image is a wire frame of the target box.

A user actually wore the HMD and moved his head. Figure 7(a) shows the experimental result. In this sequence, the maximum linear/angular accelerations were

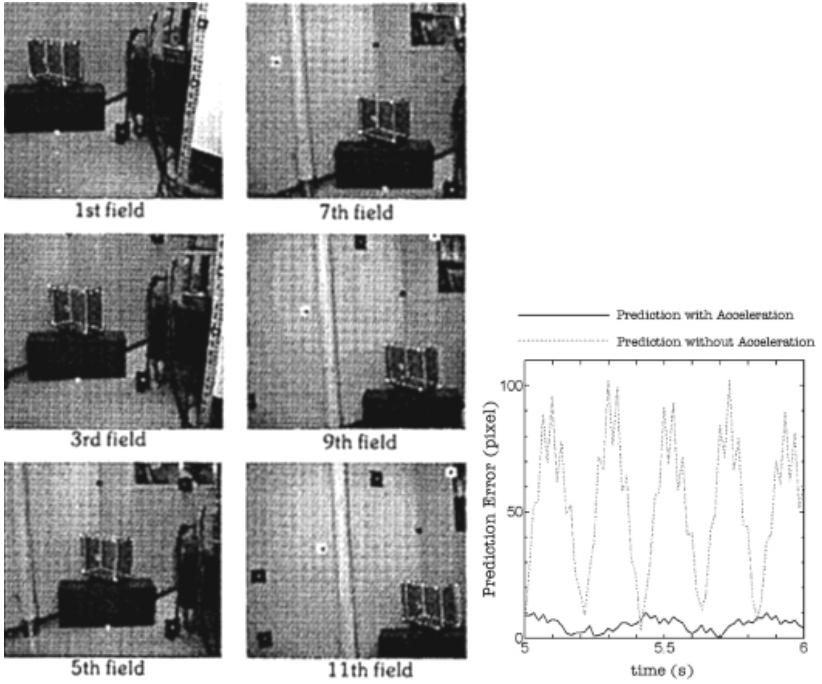


Fig. 7. Experimental result. (a) Sequence of the displayed images. (b) Prediction error of the landmark location.

10.5 [m/s²] and 49.3 [rad/s²], respectively, and the maximum linear/angular velocities were 0.65 [m/s] and 3.3 [rad/s], respectively. As can be seen from Fig. 7(a), the wire frame was overlaid accurately and the tracker did not lose the landmarks, even with such a quick head motion. The alignment errors were within 6 pixels on average and 11 pixels at maximum. Figure 7(b) shows errors of the predicted landmark locations on the image plane. The proposed method could reduce the alignment errors within 5.3 pixels on average and 10.2 pixels at maximum. The 9th field in Fig 7(a) corresponds to the instant of the maximum alignment error. Compared to the result without acceleration (53.3 pixels on average and 102.3 pixels at maximum), the prediction errors were reduced by 1/10.

4 Encountered-type Haptic Device That Renderes Multiple Virtual Objects

4.1 Path planning for encountered-type haptic devices

When only one virtual object is rendered like the case in section 2, the device can be simply placed at the location of this object (Yokokohji *et al.*, 1999). When multiple objects are rendered by a single haptic device, however, we should plan a path for

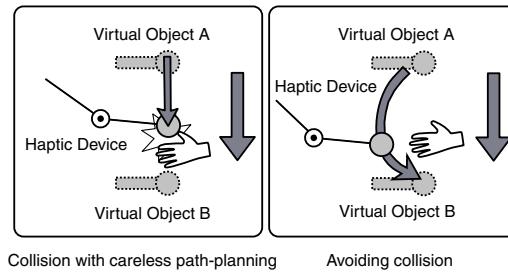


Fig. 8. Planning a safe path.

the device so that it can be placed in advance at the location of a target object that the user is going to reach (Gruenbaum *et al.*, 1997). When the device is moving from an object to another one, the device should keep away from the user's hand, avoiding an unexpected collision. To realize such a consistent behavior, the user's hand motion must be tracked in some way, such as using a passive linkage (Tachi *et al.*, 1994) and vision-based tracking (Gruenbaum *et al.*, 1997).

In the previous work by Gruenbaum *et al.* (Gruenbaum *et al.*, 1997), the motion of the haptic device was restricted in 2D, which makes the path-planning problem relatively easy. However, path-planning in 3D space is not so easy. As shown in Fig. 8, suppose a case where the device is located at object A and should move to object B, which is the next target object that the user is going to reach. Simply moving the device straight may induce unwanted collision. In a sense, the user's hand is a kind of moving obstacle that should be avoided. However, one cannot apply a simple obstacle-avoiding algorithm because the user's hand must encounter the device eventually at the location of the virtual objects. Even if the unwanted collisions are avoided anyway, the path planning is still not a simple problem. For example, always moving the device to the closest virtual object from the current user's hand seems to be a simple solution. However, this method may result in inefficient movement of the device when the user's hand is moving around a point that is equally distant from multiple virtual objects.

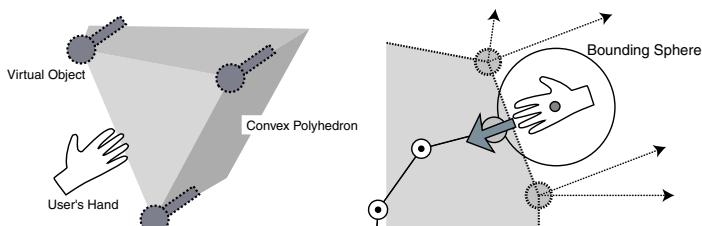


Fig. 9. Path planning strategy. (a) Polyhedron constructed from virtual objects. (b) Avoiding collision when penetrating the polyhedron.

4.2 Path-planning algorithm

We proposed a path-planning algorithm for encountered-type haptic devices that render multiple virtual objects in 3D space (Yokokohji, Kinoshita, Yoshikawa, 2001). Considering the efficiency of the device movement as well as the safety to the user, we took the following strategy:

1. Suppose a convex polyhedron which vertices are the reference point of the virtual objects as shown in Fig. 9(a). Based on the current user's hand location, the device moves to a point on the surface of this polyhedron from which the distance to the user's hand becomes minimum.
2. When the user's hand penetrates this polyhedron, the device avoids the user's hand and goes inside the polyhedron if necessary.

To find the minimum distant point on the polyhedron, the space around this polyhedron is divided by Voronoi regions (Preparata, Shamos, 1985). As long as the user's hand is outside the polyhedron, the device stays on the surface of the polyhedron. Therefore, the motion of the device is efficient.

The second strategy should be taken into account because the user's hand may go inside the polyhedron. Introducing a bounding sphere around the user's hand as shown in Fig. 9(b), the device can avoid the user's hand as though it is pushed by this sphere. More detailed algorithm is described in (Yokokohji *et al.*, 2001).

4.3 Virtual control panel

We chose a virtual control panel of an automobile as the target application, aiming at the practical use of the encountered-type haptic device in the near future. When designing an instrument/control panel of an automobile, the location of switches must be determined from the viewpoint of ergonomics as well as the visual impression. Reachability to each switch could be examined to some extent with an immersive virtual environment without haptic device. However, the encountered-type haptic device does not constrain the user's hand and can provide convincing touch sensations when the user reaches the switches. Therefore, we believe that introducing the encountered-type haptic device enables us to evaluate the arrangement of switches more precisely in the preliminary design stage.

Figure 10(a) shows the graphic scene shown to the user in our prototype system. The haptic device is a 6DOF industrial manipulator, PUMA 260, shown in Fig. 10(b). Two color CCD cameras (TOSHIBA IK-642) capture a landmark attached on the user's hand from orthogonal directions. The graphic scene is displayed on a screen of a graphics workstation located beside the haptic device. Therefore, our experimental system is currently non-WYSIWYF.

Figure 11 shows snapshots of the physical world and the virtual world in the experiments. Figure 11(a) is the case when the user's hand is actually touching one of the switches. Figure 11(b) shows the case when the device is penetrating the polyhedron to avoid the user's hand. In the virtual world image in these figures, a polyhedron used for path planning and the current device location are also rendered just for demonstration purpose.

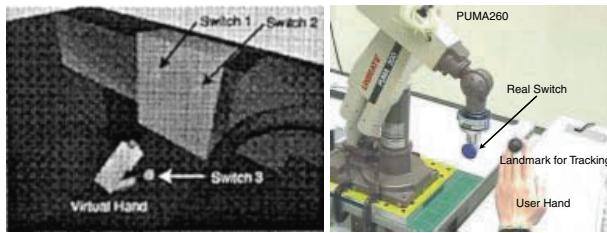


Fig. 10. Experimental system. (a) Virtual control panel. (b) Haptic device.

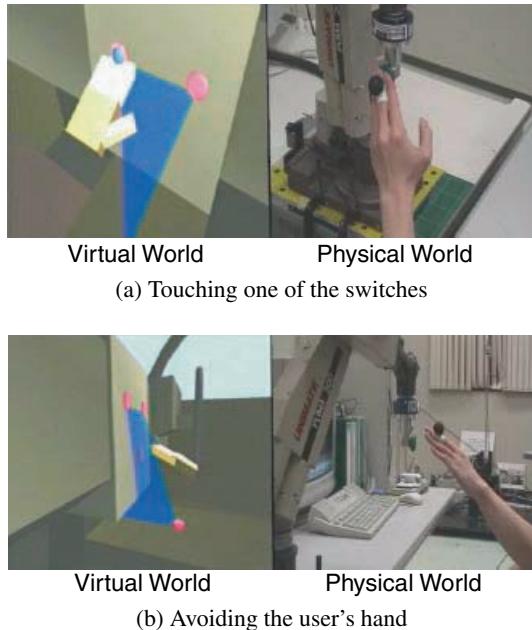


Fig. 11. Snapshots of experiment.

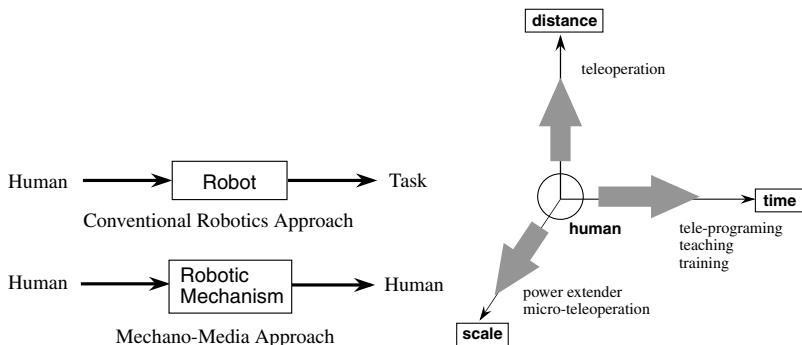


Fig. 12. Mechano-media. (a) Comparison of the approaches. (b) Augmentation.

5 Mechano-Media

5.1 Robotic mechanisms as media

Ever since our ancestors invented letters and medium to write them down, such as papers, our knowledge has been accumulated and shared. Scientific technologies have been improved based on this accumulated knowledge. However, conventional medium cannot handle all information we have. One of them is kinesthetic knowledge such as motor skill.

Motor skill contains not only visible data, such as motion, but also invisible data, such as force. Motor skill is also linguistically difficult to describe. Even if a skilled motion is given visually (e.g., by video tape), it would be difficult for beginners to follow that motion. Therefore, physical guidance as shown in Fig. 3(b) is often effective for motor skill training. The WYSIWYF display shown in section 2 could be used for transmitting motor skills. Virtual control panel shown in section 4 could also be regarded as a kind of media, transmitting industrial design concept from one to others. Conventional 2D drawings or even 3D CAD data displayed on a computer screen cannot transmit the designer's concept perfectly because one cannot actually touch it.

Here, we would like to propose a new direction of robotics: *mechano-media*, i.e., using robotic mechanisms as media of motion/kinesthetic knowledge from a human to other humans (Yokokohji, 2001). In Fig. 12(a), the top figure represents conventional robotic application where human intelligence is mapped to a robot that executes a target task. The way of mapping could be programming, teaching or teleoperation. The bottom figure, on the other hand, shows the mechano-media approach, where a motion intelligence of someone is transferred to other ones via a robotic mechanism.

According to this mechano-media concept, even conventional industrial robots can belong to this category. Suppose a mister man of coating painting has transferred his skill to an industrial robot by teaching. After that time, this mister man gets old and eventually he would pass away; however, his painting skill will exist on this industrial robot forever. Later on, another man may learn this painting skill from this robot. This is an example where a robot, which is originally designed based on the top approach in Fig. 12(a), works as a medium shown in the bottom in Fig. 12(a). In this sense, humanoid robots could be the most appropriate mechano-media for transmitting human motion/kinesthetic knowledge. In this framework, we need not find a new "personality" in humanoid robots. Instead, we can simply recall the person who originally demonstrated the motion.

5.2 Augmentation

The authors have been focusing on teleoperation research (Yokokohji, 2001). Teleoperation technique can augment our ability in a sense that we can reach and touch an object in a remote site where we cannot access directly. Micro teleoperation can

augment our ability in other way by which we can touch and manipulate a small object that we cannot manipulate by bare hands.

Figure 12(b) illustrates this augmentation concept. Mechano-media can be regarded as another type of motion mapping beyond the constraint of time. Therefore, mechano-media is a new concept extending the framework of teleoperation.

6 Conclusion

In this paper, we introduced a concept of WYSIWYF or perfect visual-haptic agreement. Our first prototype of WYSIWYF display developed in 1996 was shown. We then presented our recent progress on the key components of WYSIWYF display. If the image of the virtual control panel is displayed on an HMD and this CG image is consistently registered to the physical world using hybrid vision/inertial tracking technique that we have developed, a “WYSIWYF” situation (Yokokohji et al., 1999) will be realized. We are planning to realize such a WYSIWYF situation in the near future.

The developed visual/haptic interface systems can be regarded as mechano-media, a new concept extending the framework of teleoperation, which transmit kinesthetic knowledge from a human to other humans. We believe that mechano-media could be a new research direction of robotics. In order to transmit kinesthetic knowledge, such as motor skills, effectively, we need to understand human itself. Therefore, we need further research, taking an interdisciplinary approach not only with control engineering, robotics and virtual reality but also with psychology, biomechanics, neuroscience, etc.

References

- Azuma R (1997) A survey of augmented reality. *PRESSENCE*, 6(4):355–385.
- Azuma R, Bishop G (1994) Improving static and dynamic registration in an optical see-through hmd. In Proceedings of SIGGRAPH'94, pp 197–204.
- Burdea GC (1996) Force and touch feedback for virtual reality. Jhon Wiley & Sons.
- Dickmanns ED, Mysliwetz B, Christians T (1990) An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles. *IEEE Trans. Sys., Man, and Cyb.*, 20(6):1273–1284.
- Groen J, Werkhoven PJ (1998) Visuomotor adaptation to virtual hand positioning in interactive virtual environment. *PRESSENCE*, 7(5):429–446.
- Gruenbaum PE et al. (1997) Implementation of dynamic robotic graphics for a virtual control panel. *PRESSENCE*, 6(1):118–126.
- Henmi K, Yoshikawa T (1998) Virtual lesson and its application to virtual calligraphy system. In Proc. IEEE ICRA'98, pp 1275–1280.
- McNeely WA (1993) Robotic graphics: a new approach to force feedback for virtual reality. In Proc. 1993 IEEE Virtual Reality Annual International Symposium (VRAIS'93), pp 336–341.
- Preparata FP, Shamos MI (1985) Computational geometry. Springer-Verlag, New York.
- Tachi S et al. (1994) A construction method of virtual haptic space. In Proc. of Int. Conf. on Artificial Reality and Telepresence (ICAT'94), pp 131–138.

- Uenohara M, Kanade T (1995) Vision-based object registration for real-time image overlay. *Comput., Biol., Med.*, 25(2):249–260.
- Yokokohji Y (2001) Toward mechano-media from teleoperation. In Proc. Workshop on Advances in Interactive Multimodal Telepresence Systems, pp 3–20, Munich, Germany.
- Yokokohji Y, Kinoshita J, Yoshikawa T (2001) Path planning for encountered-type haptic devices that render multiple objects in 3D space. In Proc. IEEE VR 2000, pp 271–278.
- Yokokohji Y, Sugawara Y, Yoshikawa T (2000) Accurate image overlay on video see-through HMDs using vision and accelerometers. In Proc. IEEE VR 2000, pp 247–254.
- Yokokohji Y (1998) A visual/haptic interface to virtual environment (WYSIWYF display) and its application. In Proc. 1998 IEEE and ATR Workshop on Computer Vision for Virtual Reality Based Human Communications (CVVRHC'98), pp 99–104.
- Yokokohji Y et al. (1996a) Toward machine mediated training of motor skill. In Proc. IEEE RO-MAN'96, pp 32–37.
- Yokokohji Y, Hollis RL, Kanade T (1996b) What you can see is what you can feel—development of a visual/haptic interface to virtual environment-. In Proc. 1996 IEEE Virtual Reality Annual International Symposium (VRAIS'96), pp 46–53.
- Yokokohji Y, Hollis RL, Kanade T (1999) WYSIWYF display: a visual/haptic interface to virtual environment. *PRESENCE*, 8(4):412–434.
- You S, Neumann U, Azuma R (1999) Hybrid internal and vision tracking for augmented reality registration. In Proc. of IEEE VR99, pp 260–267.

Part 11

Robot Technology II

Session Summary

Raymond Austin Jarvis¹ and Alex Zelinsky²

Chairs

¹ Intelligent Robotics Research Centre, Monash University,

² System Engineering, RSISE, Australian National University

Whilst this session bears a relatively standard title, the papers in it represent a set of miscellaneous and disparate topics which have practically no relationship to one another. Thus, trying to introduce them as a set is hardly useful. Instead some general comments on each will serve as an entrée to entice further scrutiny by the reader.

The first paper is a down to earth description of the approaches, methodologies and smart tracker which were used to secure the Legged league RoboCup win for the second year in succession (2000, 2001) by the University of New South Wales (in Sydney, Australia). The real-time requirements to localise the team members, assess the immediate requirements of capturing the ball, targeting the goal and subsequently manipulating a four legged robot to implement the contact and deployment strategy are all described without ambiguity or unnecessary theory rhetoric. The approach is elegant in its directness and clearly very successful. Tuning by observing real competition behaviour is lauded as a critical strategy as is a simple but effective method of avoiding teammate clutter by cooperation using sound cues. The winning strategy is handcrafted rather than learned in an autonomous mode.

The second paper concerns the evolution of a hybrid motion simulator in which real physical objects and numerically simulated models interact in real-time. Complex contact and collision states can be accommodate by this approach. A real prototype model object can be tested in a simulated environment in which other components are virtual (simulated numerically). Interaction force/torques are measured and resultant motion activated using a fast parallel robot (Delta style). Flexibility and reaction are thus provided simultaneously.

The third paper provides very detailed analysis of error propagation in vehicle odometry. Sources of error are carefully identified and examined at an intense level of theoretical rigor. A comparison is made between a linearised model formulations and exact non-linear numerical solutions. Three distinct classes of error (linearisation, discretisation and sampling) are clearly distinguished between. The results presented in the paper can be used to design systems to meet acceptable levels of sensor error as well as to plan trajectories which minimise exposure to specific sources of error and to formulate optimal approaches to compensate for error.

The fourth paper presents a new approach to simultaneously providing efficiency and robustness for vision systems subject to wide lighting and dynamic variations. A Bayesian probabilistic theory is adopted for system representation and for reasoning under severe uncertainty. A Bayesian inference formulation for the dynamic case is approximated using a particle filter approach and a Bayes net is adopted for providing linkages between weighted multiple sources of hypothesis supporting

evidence. The methodology is illustrated by combining colour and (stereo) depth agents for tracking a human in varying lighting and other object interference situations. It is clear that the methodology can be scaled up to more complex scenarios and be generalised to cope with a greater variety of sensor modes and other sources of ambiguity resolving evidence.

The Evolution of a Robot Soccer Team

Claude Sammut and Bernhard Hengst

School of Computer Science and Engineering,
University of New South Wales, Sydney, Australia

Abstract. This paper traces four years of evolution of the UNSW team in the RoboCup Sony legged robot league. The lessons learned in the creation of a competitive team are instructive for a wide range of applications of robotics. We describe the development of vision and localisation procedures for robot soccer, as well as innovations in locomotion and the design of game play strategies. Since all teams in the competition are required to use identical hardware, the key factor to success in this league is the creativity of the software designers in programming the robots to perform skills that the robots were not originally intended to do and to perform them in a highly dynamic and non-deterministic environment.

1 Introduction

In 1998, the RoboCup competition introduced the Sony legged robot league as a demonstration and a full competition started in 1999. Some initial reactions were that these robots could not be made to play soccer with any real competence. In 2000, the team from the University of New South Wales (UNSW) proved otherwise.

In the first years of the competition it appeared that the vision system of the robots could not reliably detect objects on the field. Their locomotion was too clumsy to control the ball well and they could not act as a team to play an interesting game. However, it was clear that the designers of the robots believed that they could be programmed to play credibly. Unlike other RoboCup leagues, in which teams can design and build their own hardware, the competition in the legged league is based entirely on the creativity of the programmers to make robots do things that they were never originally intended for. This is an exercise that is interesting for robotics research since it tests the boundaries between what should be done in hardware and what can be done in software.

In 2000, UNSW team introduced a vision system that was fast and reliable even in the confusion of soccer match; localisation that was robust to frequent occlusions and buffeting from other robots; locomotion that was both agile and stable and behaviours that made the robots play with apparent purpose and with the ability to recover from failures. These developments were the result of long hours of analysis of the robots' capabilities and the study of many specific game play situations. Each year, following our initial entry into the competition in 1999, the teams built upon the successful ideas of the previous year and added their own innovations so that the 2000 and 2001 teams became champions of the league with decisive wins. In this paper, we first give a brief history of the technical developments involved in the RoboCup software. We then give details of the vision, localisation and locomotion systems. We also discuss communication between robots, first using sound signals

and then using the Ethernet introduced in the 2002 competition. Finally, we describe the behaviours required to make a legged robot play a good game of soccer.

2 The Sony Legged Robot League

The aim of RoboCup is to utilise the competitive spirit of the participants to stimulate research in robotics. The Sony legged robot league is one of four leagues that currently form the RoboCup soccer competition. In some leagues, the teams are required to design and build the robot hardware themselves. In the case of the Sony legged robot league, all teams use the same robotic platform, manufactured by Sony. Since all teams use the same hardware, the difference lies in the methods they devise to program the robots.

From 1999 to 2001, each team in the Sony legged robot league consisted of three robots with the matches of two 10-minute halves. In 2002 the team size was increased to four, with a larger field. The robots used in the legged league are a slightly modified version of the Sony AIBO entertainment robot. Although designed for the entertainment market, the Sony robots are extremely sophisticated machines, with an on board MIPS R4000 processor, colour camera, accelerometers, contact sensors, speaker and stereo microphones. Each of the four legs has three degrees of freedom, as does the head. Programs are written off-board in C++ and loaded onto a memory stick that is inserted into the robot. All of Sony's ERS robots run a proprietary operating system called Aperios, which provides an interface for the user-level programs to the sensors and actuators.

The field used in the Sony legged league up to 2000 measured 2 metres wide by 3 metres long. For the 2002 competition the dimensions were increased by 150%. The field has an angled white border designed to keep the ball within the field. The game ball is coloured orange and the two goals are coloured yellow and blue. The field also contains six coloured landmark poles to aid the robots in localising themselves in the field. In front of each goal there is the penalty region that only the goalie is allowed to defend.

3 History

In 1998, three teams from Carnegie-Mellon University (CMU), the Laboratoire de Robotique de Paris (LRP) and Osaka University played demonstration matches in the RoboCup competition. The robots used then were only prototypes and their capabilities were still a matter of exploration for the teams. The biggest problem they faced was programming the vision system to see the ball reliably. The CMU team won that competition largely because their robots were able to find the ball.

UNSW's first entry into the competition was in 1999. Based on the advice of Will Uther, from CMUs 1998 team, we concentrated on developing the vision system first. Our approach to strategy was to take the simplest possible behaviour and make it work with reasonable reliability. Being our first entry and being pressed for time, we made no attempt to implement any sophisticated theory. We simply wanted

to find the ball, get behind it and then walk towards the goal. We had no goalie until a simple one was written at the competition itself. However, our “keep-it-simple” approach was sufficient to get us to the final, where we were defeated by the LRP who had developed a superior locomotion method. We had used the walking routines provided by Sony whereas LRP developed a walk that was faster and more stable.

The 1999 team consisted of two undergraduate students: John Dalgliesh and Mike Lawther. Their efforts laid the foundations for the 2000 team: Bernhard Hengst, Son Bao Pham and Darren Ibbotson. They built on the 1999 vision system and world model, significantly improving the localisation method. They also developed an entirely new locomotion method that was very agile and stable. Keeping to the principle of “keep-it-simple”, the vision, localisation and locomotion were much faster than those of the other teams in the 2000 competition.

One of the biggest lessons learnt in 1999 was the importance of testing new behaviours under competition conditions, that is, with several robots getting in the way. Because of lack of time in the first year, all behaviours were tested using a single robot. Seeing all the robots interacting on the field at the competition was a revelation. In 2000 we resolved to always test with several robots on the field and to run practice games as often as possible. Our method for developing strategies was to observe and analyse as many game situations as possible and build situation-specific heuristics where needed.

In the early days of Artificial Intelligence research, attempts were made to build general purpose problems solvers. These worked well for small-scale problems, but were rarely successful in practical applications. Later, “knowledge-based” systems took the place of the general purpose problem solver when it was realised that domain specific heuristics are more powerful than general methods. We quickly made the same transition in designing strategies for our robots. The key to good performance, has been to have simple behaviours that are tailored to suit particular situations. Using this approach, the 2000 team became the legged league champions, never scoring less than 10 goals in every match and only having one goal scored against them throughout the tournament.

After each competition, the teams publish their methods and most also release their source code, as we did. So we expected the 2001 competition to be somewhat tougher. Indeed, many of the teams adopted or were inspired by UNSW’s 2000 locomotion method and several teams also used similar localisation methods. In particular, CMU developed skills that were very similar in performance to our own. UNSW’s 2001 team: Spencer Chen, Martin Siu, Tom Volgelgesang and Tak Fai Yik, developed several new manoeuvres for the robots. But the outcome of the final against CMU was certainly not a foregone conclusion. It was somewhat surprising that the score at the end of the game was 9-2 in UNSW’s favour. The main difference appeared to be that we had a larger range of situation-specific behaviours and through exhaustive testing, we had reduced the number of mistakes made by the robots during the match.

At the time of writing, the 2002 competition is only a few weeks away and much of the basic development for this year's team has been completed. Many changes in our software are due to changes in the game itself. The size of the field has been increased, an additional robot has been added to the team and wireless Ethernet communication between the robots is now possible.

In the following sections we describe the development of each of the major skills required by the robot and conclude with a discussion of the development of game play behaviours. Throughout the evolution of the UNSW software, the same basic architecture has been used. A layer of infrastructure modules provides the basic skills: vision localisation and locomotion. These are combined to produce the behaviours for playing the game. The behaviours are then combined to create different roles for the goalie and forwards.

4 Vision

4.1 Colour Classification

A colour camera is the robots primary sensor. The current robot, the ERS-2100, has a 196x144 resolution camera mounted in the “snout” of the robot. This delivers 25 frames per second. The information in each pixel is in YUV format, where Y is the brightness and U and V are the colour components. Since all the objects on the field are colour coded, the aim of the first stage of the vision system is to classify each pixel into the eight colours on the field: orange for the ball, blue and yellow for the goals and beacons, pink and green for the beacons, light green for the field carpet, dark red and blue for the robot uniforms.

The Sony robots have an onboard hardware colour look up table. However, for reasons that will be explained later, we chose to perform the colour detection entirely in software. Because colour detection can be seriously affected by lighting conditions and other variations in the environment, we need a vision system that can be easily adapted to new surroundings. One way of doing this is by using a simple form of learning. This was originally developed by the 1999 team and extended by all the subsequent teams.

The first step is to take about 25 snapshots of different objects at different locations on the field. Then for each image, every pixel is classified by “colouring in” the image by hand. All these pixels form the set of training data for a learning algorithm. Each pixel can be visualised as a point in a 3 dimensional space.

In 1999 teams software, all these pixels were projected onto one plane by simply ignoring the Y value. Pixels were then group according to their manually classified colours. For each colour, a polygon that best fits the training data for this colour was automatically constructed. An unseen pixel could then be classified by looking at its UV values to determine in which polygons it lies. As the polygons can overlap, one pixel could be classified as more than one colour.

Figure 1 shows a screen grab at the end of a run of the polygon growing algorithm. It illustrates why we chose to use polygonal regions rather than the rectangles



Fig. 1. Update of position based on single beacon

used by the colour lookup in the hardware. We believe that polygonal regions give greater colour classification accuracy.

For the 2000 competition, we kept the learning algorithm but changed the way it was used. By slicing the colour space into different Y planes, the colours are more separated in UV space. Initially, Y values were divided into 8 equally sized intervals. All pixels with Y values in the same interval belong to the same plane. For each plane, we ran the algorithm described above to find polygons for all colours.

Discretisation of the Y values into eight equal intervals was a considerable improvement, however, the robots were still unable recognise the red and blue colours of robots. Those colours appear very dark in the camera images. The Y values of all pixels were examined to try to group them in such a way that dark red and blue pixels can be distinguished from black pixels. We did this manually and settled on 14 planes of unequal sizes. In this configuration, we have more planes for lower Y values, reflecting the fact that dark colours are harder to separate. With these 14 planes, the robots could recognize the colour of the robot uniforms with reasonable accuracy.

This method of colour classification worked well in 2000 and during the development of the 2001 team. However, on arrival at the 2001 competition in Seattle, we found that the lights were so bright that some colours started washing into others. A particular source of difficulty was that the orange of the ball started to look yellow in some spots. This threw out the robots localisation because it thought it was seeing the yellow goal in places it should not. Fortunately, we had experimented with an alternative approach to learning colour classification, using Quinlan's C4.5 decision tree learning program (Quinlan, 1993). C4.5's input is a file of examples where each example consists of a set of attribute values followed by a class value.

In this case, the attributes are the Y, U and V values of a pixel and the class value is the colour manually assigned to that pixel. C4.5 turns the training data into a set of classification rules that should be able to determine the colour of a new, unclassified pixel.

Indeed, this method did prove reliable in Seattle and resulted in a considerable improvement in performance of our robots. The success of C4.5 encouraged us to investigate other, general purpose learning algorithms. For the 2002 competition we will be using a nearest-neighbour algorithm which seems to improve robot recognition further.

4.2 Object Recognition

Once colour classification is completed, the object recognition module takes over to identify the objects in the image. The possible objects are: the goals, the beacons, the ball and the blue and red robots. A blob formation algorithm links neighbouring pixels of the same colour in the image to form blobs. Although straight-forward, the blob formation algorithm is the most time consuming operation in the robot's cycle of sensing, decision making and action. In successive years, we have made the implementations faster. Based on the blobs, we identify the objects, along with their distance, heading and elevation relative to the camera and the neck of the robot.

Objects are identified in the order: beacons first, then goals, the ball and finally the robots. The reason the ball is not detected first is because we use some "sanity checks" for the ball based on the location of the beacons and goals. Since the colour uniquely determines the identity of an object, once we have found the bounding box around each colour blob, we have enough information to identify the object and compute its parameters.

Because we know the actual size of the object and the bounding box determines the apparent size, we can calculate the distance from the snout of the robot (where the camera is mounted) to the object. We also calculate heading and elevation relative to the nose of the robot and the bounding box's centroid. However to create a world model, needed for strategy and planning, measurements must be relative to a fixed point. The neck of the robot is chosen for this purpose. Distance, elevations and headings relative to the camera are converted into neck relative information by a 3D transformation using the tilt, pan, and roll of the head.

A crucial part of the vision system is a collection of "sanity checks". These are situation-specific heuristics that are used in object recognition. They were developed in the course of running practice matching and observing when object recognition failed. Some examples follow. Every beacon is a combination of a pink blob directly above or below a green, blue or yellow blob. One side of the field has the pink on the top of the colour in the beacons, while the other has it below. The beacons are detected by examining each pink blob and combining it with the closest blob of blue, yellow or green. This simple strategy was used with reasonable success in 1999, but was found to fail occasionally. For example, when the robot can just see the lower pink part of a beacon and the blue goal, it may combine these two blobs and call it a beacon. A simple check to overcome this problem is to ensure that the bounding

boxes of the two blobs are of similar size and the two centroids are not too far apart. The relative sizes of the bounding boxes and their distance determine the confidence in identifying a particular beacon.

During development, it was sometimes noticed that the robot would try to kick the ball into a corner. The cause was the robot seeing only the lower blue part of the beacon in the corner and identifying that as the goal. When the robot is near the corner and only sees part of the blue half of the beacon, the blue blob may appear large, having its width roughly twice as long as its height, which matches a feature of the goal. To avoid this misidentification, we require the goal to be directly on top of the green of the field.

It is clear that object recognition requires further development. Either a more systematic approach using model-based vision is required or a method for learning situation-specific heuristics is needed. Robot recognition presents a particular problem because: they are irregular shapes; they have sub-parts that move; they can adopt different orientations and they are very often occluded by other robots, possibly of the same team and therefore there is a confusion of blobs.

This has been an ongoing difficulty in all the competitions. Improved colour classification and blob formation in the 2002 team has improved the separation of the blobs that form robots and a robot's orientation can roughly be detected as heading left, right, away or towards the observing robot. Distance measurements can also be made by using the number of pixels from the bottom of the image to the blob of the robot. Knowing the height and angle of the head, we can compute an approximate distance. This, however, is still not very reliable.

5 Localisation

The Object Recognition module passes to the Localisation module the set of objects in the current camera image, along with their distances, headings and elevations relative to the robot's neck. Localisation tries to determine where the robot is on the field and where the other objects are. It does so by combining its current world model with the new information received from the Object Recognition module. Since all beacons and goals are static, we only need to store the positions of the robots and the ball.

The world model maintains three variables: the x and y coordinates of the robot and its heading. The robots first attempt to localise using only the objects found in the current image. Beacons and goals serve as the landmarks to calculate a robot's position but because of the camera's narrow field of view, it is almost impossible to see three landmarks at once, so any algorithm that requires more than two landmarks is not relevant. If two landmarks are visible, the robot's position is estimated using a triangulation algorithm.

More information can be gathered by combining information from several images. Thus, the localisation algorithm can be improved by noting that if the robot can see two different landmarks in two consecutive images while the robot is sta-

tional, then triangulation can be applied. Typically, this situation occurs when the robot stops to look around to find the ball.

The world model also receives feedback from the locomotion/action module, P_{walk} , to adjust the robot's position. The feedback is in the form (dx, dy, dh) where dx and dy are the distances, in centimetres, that the robot is estimated to have moved in the x and y directions and dh is the number of degrees through which the robot is estimated to have turned. Odometry information is not very accurate and small errors in each step accumulate to eventually give very large inaccuracies. Another problem occurs when the robot gets stuck, for example, by being blocked by another robot. P_{walk} is not aware of this and keeps feeding wrong information to the world model.

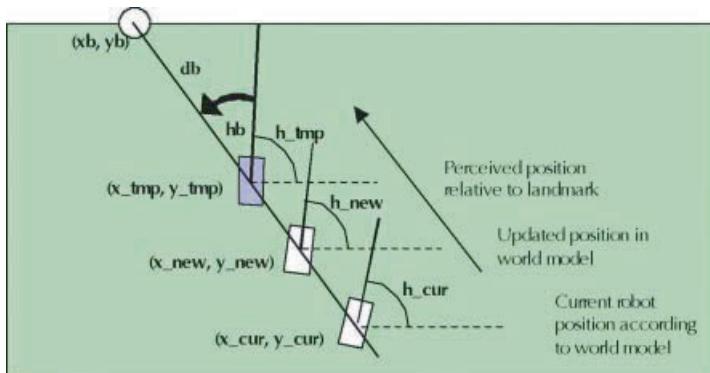


Fig. 2. Update of position based on single beacon

The methods described above cannot provide reliable localisation because the robot usually sees only one landmark in an image and the robot is moving constantly. In 2000, we introduced a method for updating the robots position using only one landmark. Suppose the landmark has coordinates (xb, yb) and its distance and heading relative to the robot are db and hb , as shown in Fig. 2. We draw a line between the (xb, yb) and the estimated current position of the robot in the world model, (x_{cur}, y_{cur}) . The robots perceived position, relative to the landmark, is the point on that line d cm away from (xb, yb) and on the same side as the current robot position, in the world model. The localisation algorithm works by “nudging” the estimated position in the world model towards the perceived position relative to the landmark.

With a camera frame rate of 25 frames/second, this algorithm converges quite quickly and accurately. Single-landmark update overcomes many of the problems caused by odometry error. Even when the robot's movement is blocked and the odometry information is incorrect, if the robot can see one landmark it will readjust its position based on that landmark. One problem remains due to the perception of landmarks. Since we mostly use the size of the object to estimate distance, noise

in the image, occlusions, motion and other factors make distance measurements unreliable.

Beginning in 1999, a confidence factor was associated with each dynamic object in the world model. This was used when incorporating new data into the existing model. To avoid the problem of error accumulation, the confidence factors of the objects were decayed after every camera frame. It was later realised that our method was actually a crude form of Kalman filtering (Kalman, 1960). Therefore, in 2002, we have cleaned up the localisation method and replaced the ad hoc confidence factors by the variance obtained from the Kalman filter.

The 2000 team adopted the philosophy of passive localization, where the robot does not actively look around to localise itself during match play. This allows the robot to concentrate on gaining control of the ball instead of wasting time locating itself all the time. This strategy resulted in our players being able to maintain their focus on the ball and move it down the field while opposition robots would take their eyes off the ball to localise themselves. Thus, for the majority of the game, we had control of the ball.

The robots obtain estimates of their positions from occasional glimpses of the beacons and the goals while they are competing for the ball. The frequency of these glimpses is sufficient for the world model to maintain a fairly accurate representation of the game state. An advantage of the low stance of our robots is that they can often see beacons without having to look up. In tuning the locomotion parameters, special care was taken to maintain this benefit. Another factor that increases the frequency of seeing landmarks is the fact that the robot inevitably loses track of the ball, for example when challenged by an opponent. During execution of the find ball routine, the tilt of the scanning head was also tuned so that the robot can see the beacons as it scans for the ball. So the Find Ball routine also acts simultaneously as a method of active localisation.

In 2001 a more active localisation method was introduced that forced the robots to look about and localise when the confidence levels in the world model fell below a specified threshold. In 2002 this has been modified so that when the variance from the Kalman filter is large, active localisation is triggered. A major change in 2002 is the introduction of wireless Ethernet communication between the robots. This enables the robots to share their world models and offers the potential to improve the accuracy of the positioning of objects in the model.

6 Locomotion

In 1999, we used the walking routines provided by Sony. These were not designed for playing soccer. A library contained routines for a variety of simple walks and turns. Our biggest problem was that there was no smooth transition from a straight walk to a turn, so the robots were very clumsy in manoeuvring and frequently fell over in transitions from one type of walk to another. The 2000 team wanted to be able to drive the robots as if controlled by a three degree of freedom joystick. That is, the robot should be able to travel in any direction, regardless of which way it

was facing. To achieve this, they devised a parameterised walk. The paw of each leg describes a rectangle as it moves (Fig. 3). A particular kind of walk or turn is specified by giving the dimensions and orientations of the rectangles. The joint angles are automatically computed by inverse kinematics.

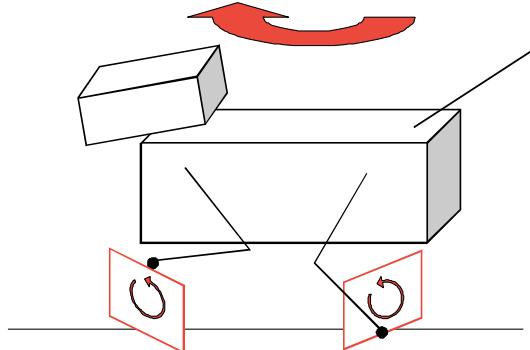


Fig. 3. The robots paw describes a rectangular locus. The robots walk can easily be changed by adjusting the dimensions and orientation of the rectangle

By continuously varying the parameters, the robots can make smooth transitions from a straight walk to a turn to a sideways walk, etc. Thus, the robots are now highly manoeuvrable. A further refinement was to improve the speed and stability of the robots. After considerable experimentation with different parameter settings, it was found that a long, low stride gave the greatest speed while also being very difficult to push over during a game.

There are three control parameters, one speed parameter and 8 stance parameters that influence the leg movements for a particular walk style. A machine learning solution to finding the best speed and stance parameter setting suggested itself. We could perform a gradient ascent on the performance measure (for example forward speed) incrementally adjusting the parameters over many test runs. The problem with automating this approach was that it would take considerable time and resources to set up the learning environment. Of concern was the wear and tear on the robot motors and leg joints given the long periods of training required. Hornby, et al (Hornby, Fujita, Takamura, Yamamoto, Hanagata, 2000), report continuous training times of 25 hours per evolutionary run using their genetic algorithm (GA). The approach we adopted was to manually adjust the parameters after a number of runs of observing and measuring the performance. Unlike gradient ascent or a GA we were able to adjust the parameters using our judgement and knowledge about the robots dynamics. These considerations include:

1. The home position of the legs needs to be adjusted so that the robot will dynamically balance on the two legs that touch the ground.
2. A longer stride required the robot body to be closer to the ground to allow the legs to reach further.

3. The front and back legs do not interfere with each other during the cycle of the gait.
4. The legs needed some sideways spread to allow the robot to move its legs laterally so it could walk sideways.

Besides the ground speed, we would also judge the reliability of the walk and the deviation from the intended path as a part of the performance measure. In this way we found that there was a trade off between the frequency of the gait and the length of the stride. Maximum ground speed was achieved with a longer stride and a slower gait frequency. This manual approach used about 10 minutes of robot running time and only about 12 iterations of parameter adjustments to find a good performance setting resulting in the characteristic low forward leaning stance of the UNSW robots.

A further refinement that increased ground speed considerably was the introduction of a canter action. The canter action sinusoidally raises and lowers the robots body by 10mm synchronised with the trot cycle. The parameters could then be manually tuned so that the robot was able to reach speeds of 1200cm/min. This compares to 900cm/min achieved using a genetic algorithm approach (Hornby et al., 2000). The camera is not as steady in this type of walk because of the additional canter movement.

Head and walk parameters are varied dynamically allowing interesting skills to be developed in the behaviour module such as ball tracking, ball circling and walking towards an object. To track the ball, for example, we would take the horizontal and vertical displacement of the ball from the centre of the camera image and use parameters proportional to these measures to drive the head movement relatively. In this way, the head can be made to move to keep the ball in the centre to the image. As another example, to make the robot walk towards the ball in a straight line, we first track the ball as above and then use the degree of twist in the neck from straight ahead to feed the turn parameter of the walk. This will ensure that the robot will turn towards the ball if it should veer to either side as it walks towards it. These types of applications of the action module form the skills that are more fully described in the section on behaviours.

The locomotion method described above gave us a great advantage in the 2000 competition. In 2001 several teams copied us, either by adopting our source code or by reproducing our gait in a different implementation. So in 2002, we looked for a ways to speed up the walk further. We found that by changing the locus from a rectangle to a trapezoid and reworking the trot cycle time, we were able to obtain a 25% increase in speed.

7 Communication

Team coordination amongst the robots is quite difficult. Until 2002, the only communication possible was by sound and this was quite unreliable. The Sony robots have onboard speakers and stereo microphones. However, during a competition

match, there are large crowds cheering, creating loud background noise and the robot motors also generate noise through the body. So the only reliable signal that could be sent was a single, high-pitched tone. As we shall see in the next section, this was sufficient to provide some useful information to team mates.

In 2002 a wireless Ethernet link became available. The robots have an onboard 802.11b PCMCIA card that enables them to become nodes on a Local Area Network. It was agreed by all the teams that the wireless communication should only be used between robots and should not be used for centralised control from another computer. Since the WLAN has not yet seen full use in a competition, there is still much exploration needed to properly exploit its capabilities.

At present, the primary use for the WLAN during a game is for transmission of the world models from one robot to all the team mates. In theory, this should approximate global vision since four robots should be able to see most of the field. However, all the robots have significant localisation errors, so combining world models is rather tricky. We still prefer to act upon sensor data rather than the world model so rather than combining world models from different robots, we use the onboard sensor data, if it is available and only use the data from other robots if nothing else is available.

Perhaps the greatest benefit of the WLAN, so far, has not been during game play, but during debugging. We are able to transmit real-time images from the robots so that we can see what they are seeing when something goes wrong. We can also transmit real-time displays of the world model so that we can detect mistakes in the localisation routines.

8 Game Play Behaviours

To play a game of soccer, we require a library of behaviours that can be combined to create a player. These skills include: find the ball; get behind the ball; dribble on the chest; push with the chest; kick with the paw; a power kick and a sideways kick while turning with the ball. The players are the forwards and the goalie.

The basic strategy for the forward is simple: the robot tries to align itself behind the ball, and then selects an appropriate attacking skill to move the ball towards the target goal. However, the details of the strategy are modified by the robot's location on the field and its relation to other robots.

An important innovation in 2000 was keeping team mates separated so that they did not interfere with each other. For example, robot 1 is dribbling the ball. Its team mate, robot 2, sees the ball and starts running at the ball. When robot 2 gets close to robot 1 it backs up, giving way to robot 1. This behaviour also reduces the occurrence of rugby-type "scrumbs" when a pack of robots all fight for the ball, mostly just getting in each other's way. The backup behaviour tends to keep one robot on the "wing" of its team mate, which effectively makes one robot wait outside a scrum for the ball to pop out. Once that happens, the "wing" can attack the ball in clear space.

In 2001, sound communication was added when a robot backed away from its team mate. A high-pitched tone signalled to the front player that a team mate was behind. Using the robots' stereo microphones, it was sometimes possible to know which side the supporting robot was on. Thus, if an opponent challenged the front player, the ball could be kicked to the side of the supporting player.

With the introduction of a fourth player and the WLAN in 2002, the backing away strategy has been extended to include the notion of a "potential field". That is, each robot on the same team tends to repel the other robots so that they stay well spread. Of course, special conditions apply as the robots approach the ball since we do not want a deadlock to occur if two team mates are attacking. In general, the robot that has the best position (eg, facing the goal) has the right-of-way.

The behaviour of the forward is also modified depending on its position in the field. For example, when the robot has the ball near the target goal, it is worth taking time to line up on the goal. However, if the robot is far from the target, it is more effective to simply knock the ball into the opponents half. This wastes little time and does not allow opponents the chance to take the ball away. If the robot is approaching the goal and the ball is in the clear, it is worth lining up carefully. However, if an opponent robot is nearby, it is better to simply attack and take control away from the opponent.

The goalkeeper strategy employed by our teams since 2000 utilises three behaviours to defend its goal: find the ball; track the ball and acquire a defensive position; clear the ball. Clearing the ball is activated when the ball gets close to the goal or the ball enters the penalty area. The mode finishes when the ball is kicked, lost, or moves more than a set distance from the defended goal. Upon deciding to clear the ball the robot will evaluate its position relative to the ball and determine whether it can directly attack the ball or should reposition behind it.

Ideally, we would like to have a game in which team mates pass the ball to each other. This is still illusive since it takes considerable accuracy in vision and localisation and fast tracking to catch the ball.

9 Conclusion

The Sony legged league of RoboCup has been a source of many interesting research problems for robotics and artificial intelligence. Initially, it was not clear if it made sense to use these robots for playing soccer. However, the fact that we now have credible play demonstrates that providing a powerful software development environment for a "general purpose" platform allows programmers to be creative in constructing competent behaviours. The performance of a robot will always be limited by the quality of its sensors and actuators but clever programming allows us to get the most out of the available resources and sometimes even surprise the designers of the robots.

The other biggest lesson learned through our involvement in RoboCup has been the importance of situation-specific behaviours. The reason for much of the success of the UNSW robots is that they are highly reactive and therefore fast. All the plan-

ning for the robots is done during countless hours of practice sessions so that when they are on the field, we have anticipated many of the different situations that the robots might find themselves in. Clearly this is very labour intensive. Ideally we would like the robots themselves to learn through practice.

Acknowledgements.

The results described here are due to the efforts of successive UNSW RoboCup teams: John Dalgliesh and Mike Lawther (1999); Bernhard Hengst, Son Bao Pham and Darren Ibbotson (2000); Spencer Chen, Martin Siu,, Tom Vogelgesang and Tak Fai Yik (2001); James Brooks, Albert Chang, Min Sub Kim, Benjamin Leung, Canh Hao Nguyen, Andres Olave, Tim Tam, Nick von Huben, David Wang and James Wong (2002).

References

- Hornby GS, Fujita M, Takamura S, Yamamoto T, Hanagata O (2000) Evolving robust gaits with Aibo. In IEEE International Conference on Robotics and Automation, pp 3040–3045.
- Kalman RE (1960) A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering, 82(D):35–45.
- Quinlan JR (1993) C4.5: Programs for machine learning. Morgan Kaufmann, San Mateo, CA.

A New Class of Hybrid Motion Simulation Using a Very Fast Parallel Robot

Masaru Uchiyama¹, Susumu Tarao², and Hiroshi Kawabe¹

¹ Department of Aeronautics and Space Engineering, Tohoku University, Aoba-yama 01, Sendai 980-8579, JAPAN

² Department of Intelligent System Engineering, Ichinoseki National College of Technology, Takanashi, Hagisho, Ichinoseki 021-8511, JAPAN

Abstract. The paper presents application of a parallel robot to a new class of hybrid motion simulation in which most of simulation is done using a complex numerical model and only a small but indispensable part of simulation is done using a physical model. The advantage of this type of simulation is that it exploits rapidly growing computational means for testing a real object in the simulation. A key element in such a class of simulation is a motion table that realizes motion with high frequency and combines the results of the numerical simulation precisely with the motion of the physical model or vice versa. The paper presents application of a fast parallel robot to such a high-speed motion table. It also presents modeling of the impact dynamics of motion in the simulator and its verification through experiments.

1 Introduction

With the increase of computing power in recent years simulation using numerical models is making rapid progress. Design of mechanical systems is more and more being based on the simulation instead of hand-made prototyping. All equipments on a spacecraft, for example, are supposed to be simulated thoroughly on the ground before being operated in space.

Motion simulation is a technique for such design requirement to generate adequate motions of the mechanical system corresponding to given different situations. Multibody dynamics is a theoretical backbone upon which the motion simulation is performed (García de Jalón, Bayo, 1994), (Wittenburg, 1977). Indeed, it is a promising field and will be used progressively in the design of mechanical systems. However, the technique cannot be used for testing real equipments that are prototyped, of course. On the other hand, equipments need to be tested when they are prototyped before being put into real operation.

A solution to this is a hybrid motion simulation using virtual environments. The hybrid motion simulation is an effective and economical means for such purpose. A hybrid model for the simulation includes a numerical (virtual) model and a real (prototyped) model. By using the real model together with the numerical model, one can simulate, for example, a motion including complicated collisions and contacts. This type of hybrid motion simulation is common in aerospace engineering; examples are a so-called “iron bird” which is a ground rig test facility for aircraft development used in preproduction phase (Sakurai, Hirata, Horie, et al., 1997) and

a docking simulator which deals with collision and contact of a spacecraft in zero gravity (Grimbert, Noirault, Marchal, et al., 1989).

The paper presents application of a parallel robot to the hybrid motion simulation. The new concept that this paper presents for the hybrid simulation is that most of the simulation has been done using a complex numerical model; consequently, only a small but indispensable part of simulation is done using a real model. The advantage of this type of simulation is that it exploits the rapidly growing computational means for testing a real object in the simulation.

A key element in such a class of simulation is a motion table that realizes motion with broad bandwidth and combines the results of the numerical simulation precisely with the motion of the physical model or vice versa. The hybrid simulator being presented in this paper exploits a parallel structure (Pierrot, Uchiyama, Dauchez, et al., 1990), (Pierrot, Dauchez, Fournier, 1991), (Uchiyama, Iimura, Pierrot et al., 1992) to implement the high-speed motion table.

Parallel structures are typically those structures in which the output link (i.e. the end-effector in the case of robot manipulators) is driven in parallel by the link mechanisms (Merlet, 2000), (Uchiyama, 1994). These structures have characteristics that usual robot structures, namely, serial structures with serially connected links, don't possess. Parallel structures have remarkable characteristics such as high precision, high load capacity, high rigidity and high speed. As the position and orientation of the output link is decided by the position and orientation of each of the links that is connected with the output link, these structures don't have the problem of accumulation of actuator errors as is the case for robots of serial structure. Therefore, they can achieve very precise positioning. Also, the output link is supported by more than one link mechanism, and thus they exhibit high load capacity and high rigidity. For parallel structures, the robot with no actuators on the movable parts can be designed. So mass of the movable parts can be reduced drastically, thereby giving rise to high-speed characteristics.

The following paper explains first, the basic idea of the new class of hybrid simulation, before presenting the design of a fast parallel robot to be used in the simulation. Then the paper presents the dynamic performance of the parallel robot, implementation of a simulator using the robot, and target capturing experiments performed by the simulator, and finally, followed by concluding remarks.

2 A New Class of Hybrid Simulation

There are some research reports found related to the hybrid simulation mainly in the field of space engineering. A hybrid motion simulator has been used to test docking and berthing operations for the ETS-VII mission by the National Space Development Agency of Japan (NASDA) (Shimoji, Inoue, Inaba, et al., 1995). For the MUSES-C mission by the Institute of Space and Astronautical Science (ISAS), in which a spacecraft touches down on an asteroid, a hybrid robotic simulator has been used for the touchdown tests (Sawai, Kawabe, Asahina, et al., 2000). At the laboratory level, some types of hybrid motion simulator have been developed for

the space robotics research (Dubowsky, Durfee, Corrigan, et al., 1994), (Yoshida, Mavroidis, Dubowsky, 1995).

These conventional motion simulators have adopted ball screws, gimbals and/or Stewart platforms (Stewart, 1965) as driving mechanisms for the motion table. Therefore, the frequency response bandwidth achieved is at most around 10 [Hz]. This bandwidth is not sufficient to simulate precisely the impact dynamics of high frequency to appear, especially, in the collision of smaller objects.

The new class of hybrid simulation that this paper proposes is different from those conventional hybrid simulators in the following respects:

1. The simulator exploits the computer power as much as possible and maximize the numerical part of simulation. Thus the simulation becomes as flexible as a numerical simulation.
2. A minimum set of real model is tightly connected to the numerical model using a broad bandwidth motion table.

The hybrid simulation based upon the above principle becomes almost like a computer simulation, in which a part of function is executed by a real model instead of a computer program. The part to be simulated by the real model can be a prototype equipment such as a sensor or a mechanical part to be tested in the preproduction phase.

The hybrid simulation is effective for the case where the equipment, such as a teleoperated robot, needs some human operations. The operator must train for various tasks in advance on the ground. Especially, collision and contact tasks beset with a lot of risks require thorough testing and training in advance. It is possible to test flight models of a contact area by using them as physical models for the hybrid simulation.

A key element for this class of simulation is a motion table that realizes high-frequency motion. The paper presents to exploits a parallel structure to implement such high-frequency motion. The motion table used in the robot (Inohira, Yukawa, Akima, et al., 1998) has a bandwidth of 40 [Hz], that is much broader than those of the conventional hybrid simulators. The feature makes it possible to combine seamlessly the numerical dynamics with the physical dynamics and consequently simulate the collision and contact motions with high precision and reliability.

3 Design of a Fast Parallel Robot

3.1 The HEXA Structure

Parallel structures fit the framework for high-speed operating robots, because all actuators for driving the structure can be placed on the base. Kokkinis and Stoughton have developed a spatial three-DOF (degrees of freedom) DD (direct drive) parallel structure robot (Kokkinis, Stoughton, 1988). This robot makes use of DD motors, but it has light moving parts and also affords high operating speed. Clavel has proposed a spatial three-DOF high-speed parallel robot called DELTA (Clavel, 1989).

However, these robots have only three DOF, so their use as a robot is limited. If three more DOF are given to the wrist then the robot has six DOF, but weight of the moving parts increases and high-speed operation is sacrificed. On the other hand, as a result of extension of the structure of DELTA, Pierrot et al. proposed a spatial six-DOF parallel robot called HEXA (Pierrot et al., 1990), (Pierrot et al., 1991). They thought that, keeping the good features of the DELTA robot, a very fast six-DOF parallel robot could be designed. Specifications considered in the design are

1. In order to design a low cost robot, the kinematic chains to make the robot should be all the same,
2. In order to place the motors on the ground, the joint to be driven should be the first (root) joint of each kinematic chain,
3. For electrically driving, the joint to be driven should be rotary one, and
4. For high-speed motion, the links in the kinematic chains should be as light as possible, and a mechanism that enables such design should be employed.

3.2 A Prototype Robot

Following the above specifications, a prototype of the HEXA robot was designed (Uchiyama et al., 1992). Figure 1 shows a structure of the prototype robot. Each kinematic chain in the mechanism is designed as follows:

1. The first link, named an arm, is driven by a motor placed on the base,
2. The arm is connected by a ball-joint (this could be replaced by a U-joint) to the second link, named a rod, and
3. The rod is connected by a ball-joint to a plate, named a traveling plate, to carry an end-effector.

Each kinematic chain is placed in a pair on the base. The same motion of each pair makes the motion of the HEXA like that of the DELTA. Since the rod receives no internal moments, and supports only compression and tension, it can be designed very light.

The motors used are the Yokogawa Precision DM1015B DD motor. The motor has rated speed of 2.0 [rps] and rated torque of 10 [Nm]. With this motor 90 degree point-to-point motion within 0.16 second can be achieved. Experiments using the prototype robot show that the traveling plate can move at the speed of 5.94 [m/s] and the acceleration of 22 [G].

3.3 Design of Ball-joints

We use a ball-joint to connect the arm and the rod and the rod and the traveling plate. The ball-joint is a key part in the HEXA robot. Of course, there are many kinds of ball-joints which are commercially available. Unfortunately, those ball-joints have only limited movable range that does not meet our design requirement. We asked, therefore, a manufacturer to make additional machining to enlarge the movable range to meet our design requirement. Design of a ball-joint with enlarged movable range is still a current research issue.

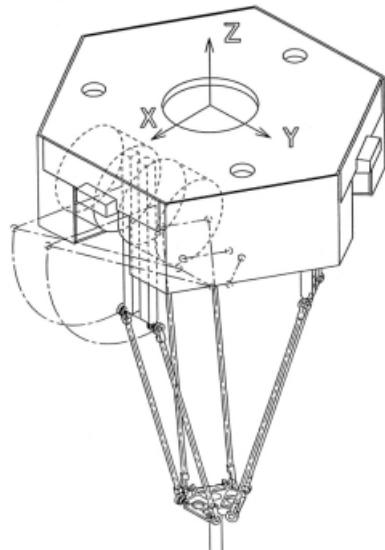


Fig. 1. HEXA

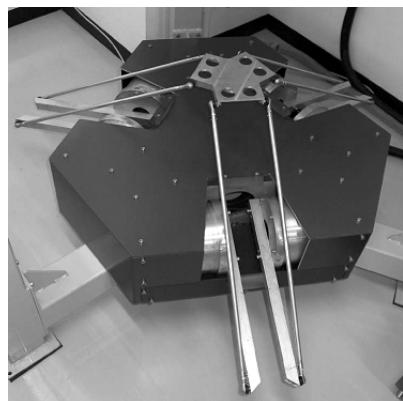


Fig. 2. HEXA97

4 The Second Prototype Robot

The second prototype robot was designed based on the experience of the first HEXA robot (Inohira et al., 1998). It was designed for faster motion with low friction. We carefully considered the selection of motors and eventually chose the NSK Mega-Torque DD motor, SSB045, with the maximum torque 45 [Nm]. Figure 2 shows a photograph of the second prototype robot, named HEXA97.

4.1 Fast Motion Experiment

An experiment was performed in order to investigate the maximum velocity and acceleration of HEXA97 (Akima, Tarao, Uchiyama, 1999).

Keeping the attitude of the traveling plate as $(0.0, 0.0, 0.0)$ [rad], it is moved from $(0.0, 0.0, 0.23)$ [m] to $(0.0, 0.0, 0.80)$ [m] and then down to $(0.0, 0.0, 0.30)$ [m] in a reference frame fixed on the base (xy -plane represents a horizontal plane and z -axis represents a vertical axis). A trajectory is generated by 4-1-4 polynomial interpolation. An experimental result for the z -axis motion is shown in Fig. 3. It is seen that the HEXA97 robot achieved maximum velocity of 10 [m/s] and maximum acceleration of 40 [G] at the center of traveling plate. The result shows that motion simulation of high velocity and acceleration is possible for the simulator that uses the HEXA97 robot.

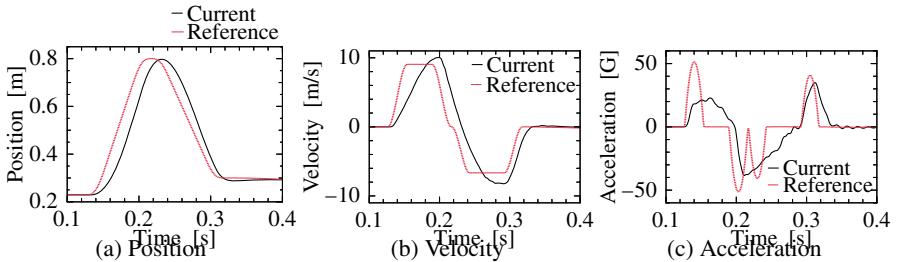


Fig. 3. Experimental results

4.2 Frequency Response Experiment

To estimate the response bandwidth of the simulator, a frequency response experiment was performed (Akima et al., 1999).

It is supposed that the impulsive motion which is caused by a contact or collision may be of high frequency but of low amplitude. Considering those features of motion to be simulated, we select as an amplitude of input sine wave to the motion table, 0.005 [m] for translational motion and 3 [deg] for rotational one, respectively. The amplitude of the rotational motion corresponds to the same amplitude of motion in the motor angle as that for translational one. The input sine trajectory is given for the traveling plate center. The output is computed from the motor angles using the forward kinematics. The frequency response experiment is performed at the point of (0.0, 0.0, 0.5) [m] in the base frame. The neutral orientation for rotational motion is zero at the point. The input frequency is varied from 1 [Hz], and from 2 [Hz] to 40 [Hz] with a step of 2 [Hz].

Experimental results with no load on the traveling plate are shown in the Bode plots of Fig. 4. The figure shows that a resonance point resides at around 160 [rad/s]. The bandwidth is around 260 [rad/s]. The shapes of the Bode plots are different from one another; this is because loading conditions on the motors varies according to the motion direction due to the arrangement of motors. There are two resonance points in each motion except in the z -axis motion and the motion around z -axis. It is considered that the lower resonance point is for the servo controller and that the higher one is due to the resonance of rods. This can be shown that the first bending mode of the rod matches with the higher resonance point in the experimental results.

An experiment with loads that are an F/T (force/torque) sensor and a small object on the traveling plate was also performed. The mass of the F/T sensor and the object are 0.710 [kg] and 0.288 [kg], respectively. Experimental results show that the robot still has a bandwidth of 200 [rad/s] even with the loads.

5 A High-Speed Hybrid Motion Simulator

We have developed a hybrid motion simulator adopting the HEXA97 robot (Kawabe, Inohira, Kubota, et al., 2001), (Tarao, Inohira, Uchiyama, 2000). Figure 5

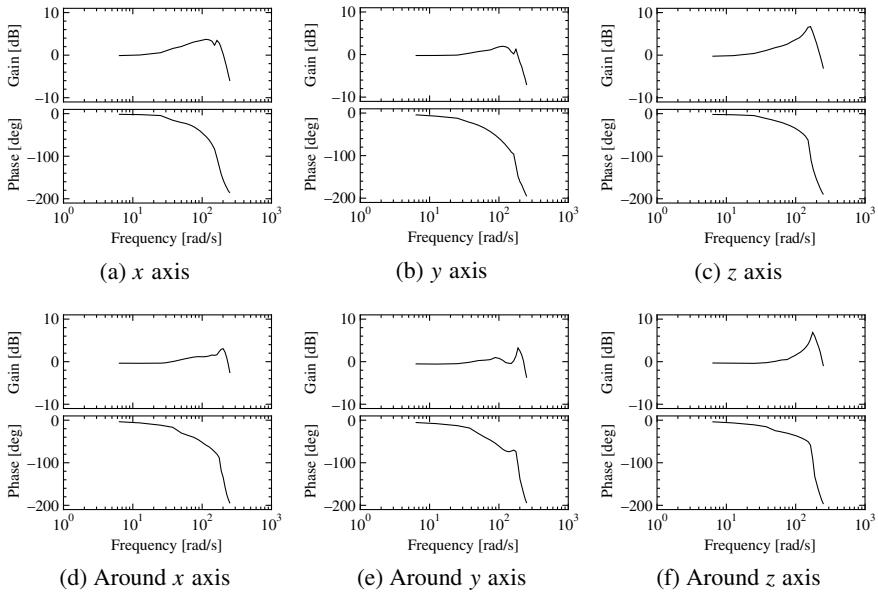


Fig. 4. Bode plots of frequency response

shows an overview of its physical part. As has been shown in the previous section, the simulator has a response bandwidth of up to around 40 [Hz], that is much larger than those of the conventional hybrid simulators.



Fig. 5. An overview of the simulator

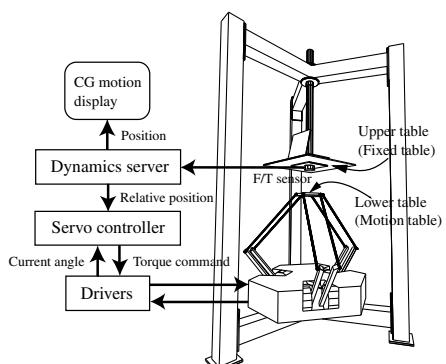


Fig. 6. System configuration

Figure 6 shows the system configuration of the developed simulator. The simulator has an upper fixed table and a lower motion table. Physical models are attached to each table. The motion table is the output link of the HEXA97 robot and possesses six DOF. Forces and torques interacting between the two physical models are measured by the F/T sensor mounted on the fixed table.

The dynamics server calculates the numerical model using the forces and torques obtained by the F/T sensor. The algorithm for the calculation is based on the Rosenthal's algorithm (Rosenthal, 1990). The server sends the calculated relative position and orientation between the two physical models to the controller of the robot (motion table). Motion of the numerical model is shown on the CG (computer graphics) motion display.

The controller of the robot derives the desired angles for the motors using the inverse kinematics of the HEXA97 robot to command torques $\tau \in \mathbb{R}^6$ to the motors under a PD control law:

$$\tau = \mathbf{K}_p \delta\theta + \mathbf{K}_d \delta\dot{\theta} \quad (1)$$

where $\mathbf{K}_p, \mathbf{K}_d \in \mathbb{R}^{6 \times 6}$ are diagonal matrices of proportional gains and differential gains, respectively, and

$$\delta\theta = \theta_{des} - \theta_{cur} \quad (2)$$

where $\theta_{des} \in \mathbb{R}^6$ are the desired angles and $\theta_{cur} \in \mathbb{R}^6$ are the current angles from the resolvers. When the two physical models make a contact with each other, the simulator system forms a feedback loop via the F/T sensor.

6 Experiments on the Simulator

6.1 A Model of Impact Dynamics

Considering that the numerical dynamics should correspond to the real dynamics in the real world, one can regard (3) as the simulator's characteristics for impact dynamics.

$$\mathbf{G}_{sim}(s) = \mathbf{G}_1(s) \mathbf{G}_2(s) \mathbf{G}_3(s) \quad (3)$$

where

$$\mathbf{G}_1(s) = \mathbf{J}^{-T} (\mathbf{K}_p + s\mathbf{K}_d) \mathbf{J}^{-1} \quad (4)$$

$$\mathbf{G}_2(s) = \frac{1}{1 + sT_l} \quad (5)$$

$$\mathbf{G}_3(s) = e^{-sT_d}. \quad (6)$$

$\mathbf{G}_1(s)$ represents the transfer function from the positional error $\delta\mathbf{p}$ at the motion table, to the forces/torques \mathbf{F} acting on the motion table. $\mathbf{G}_2(s)$ represents the time

lag in the simulation loop. It is mainly due to a low-pass filter for the F/T sensor and also due to the response characteristics of the DD motors. Though time lag of some elements may be of the second or higher order, these are regarded as of the first order, approximately. $G_3(s)$ represents time delay elements due to numerical calculations in the servo controller and the dynamics server. T_d is the total delay time.

Figure 7 shows the simulator's characteristics for the impact dynamics. The simulator's dynamics intervenes between the numerical and the real models. As shown in the figure $G_{sim}(s)$ is divided arbitrarily into $G_{sim1}(s)$ and $G_{sim2}(s)$.

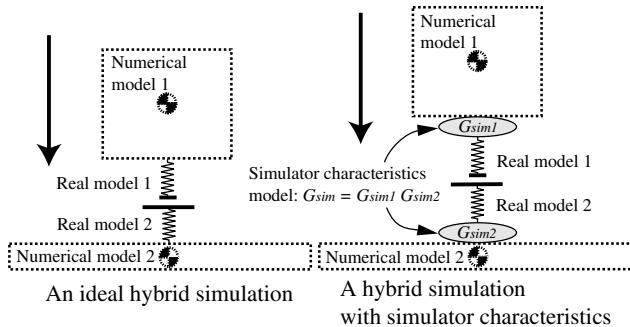


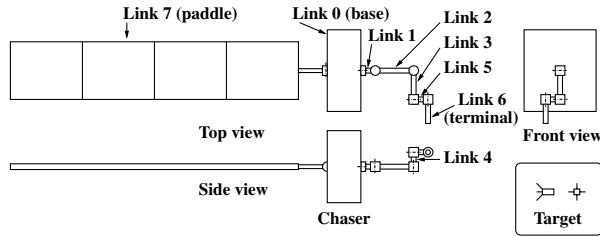
Fig. 7. A model of the simulator dynamic characteristics

6.2 Conditions for Experiments

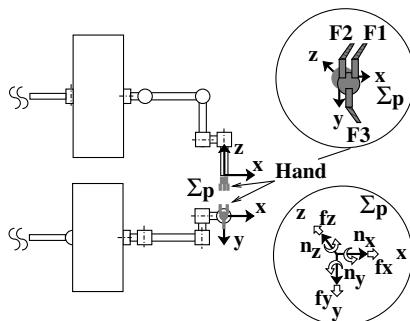
As a preliminary experiment to explore the feasibility of the hybrid simulation, motion of a space free-flying robot (chaser) capturing a nano-satellite (target) has been simulated.

The numerical models of the chaser and the target are designed as shown in Fig. 8 and are implemented by the dynamics server. The chaser consists of eight rigid links with seven revolute joints. The mass of those links, numbered from 0 to 7, are 300 [kg], 35 [kg], 22 [kg], 22 [kg], 17 [kg], 26 [kg], 18 [kg] and 32 [kg], respectively. The target consists of a single rigid body of 10 [kg]. Other necessary parameters are listed in Table 1 (the symbols r_1^k and r_2^k used in the table correspond to those used in (Rosenthal, 1990)).

Two physical models are combined with the above numerical models. Figure 9 shows the two physical models: The upper model is a robot hand (Barrett Technology Inc. BH8-255) with three fingers, F1, F2 and F3, mounted on the chaser terminal. Physically, it is attached to the fixed table via the force/torque sensor. The lower model is a part of the target to be gripped by the hand and is made of duralumin plates. It is attached to the motion table. To represent the components of the forces/torques \mathbf{F} , f_x , f_y , f_z , n_x , n_y and n_z , we use the local coordinate frame Σ_p which is fixed on the robot hand as shown in Fig. 10.

**Fig. 8.** A set of numerical models**Table 1.** Parameters of the numerical models

Link number k	Length [m]		Moment of inertia [kgm^2]		
	$ r_1^k $	$ r_2^k $	$I_k[1, 1]$	$I_k[2, 2]$	$I_k[3, 3]$
Chaser 0	—	—	128.5	84.5	68.5
“ 1	0.35	0.15	0.2844	0.2844	0.0440
“ 2	0.15	0.40	0.0275	1.1871	1.1871
“ 3	0.40	0.30	0.0275	0.6738	0.6738
“ 4	0.30	0.15	0.1381	0.1381	0.0213
“ 5	0.15	0.15	0.2113	0.2113	0.0325
“ 6	0.15	0.25	0.3863	0.3863	0.0225
“ 7	0.35	3.30	3.5267	119.69	116.16
Target 0	—	—	0.024	0.06	0.06

**Fig. 9.** A set of physical models**Fig. 10.** Local coordinate frame

Initially, the chaser remains stationary in the inertial frame, keeping each joint angle constant with an adequate control and the robot hand remains also stationary keeping each finger open. The target is stationary within the reach of the robot hand. With these initial conditions, the robot hand grasps target; see Fig. 11. The identical periodic time of 2 [ms] is adopted as the sampling time for the motion table control and also as the interval of the numerical model calculation time step. In addition, the parameter k_p which is the diagonal element of the gain matrix K_p is set at 200 [Nm/rad].

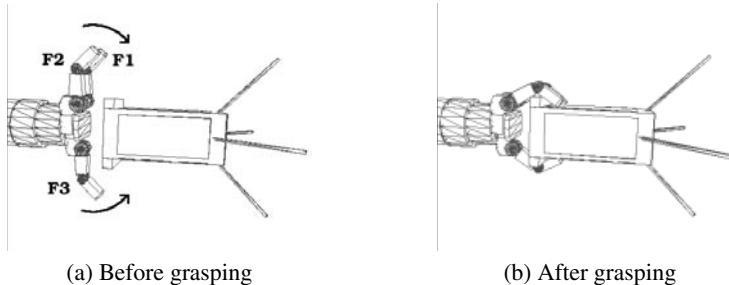


Fig. 11. Grasping motion

6.3 Experimental Results

Figure 12 shows a part of time series data of the finger tip velocity vector norms during the target capturing process. The CG outputs of the motion simulation for adequate time steps are shown in Fig. 13.

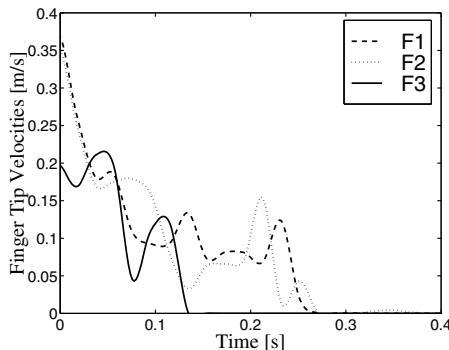


Fig. 12. Finger tip velocities

To evaluate the interaction forces/torques quantified by the hybrid motion simulator, we use (3) after neglecting the dynamic portion, that is, approximating it

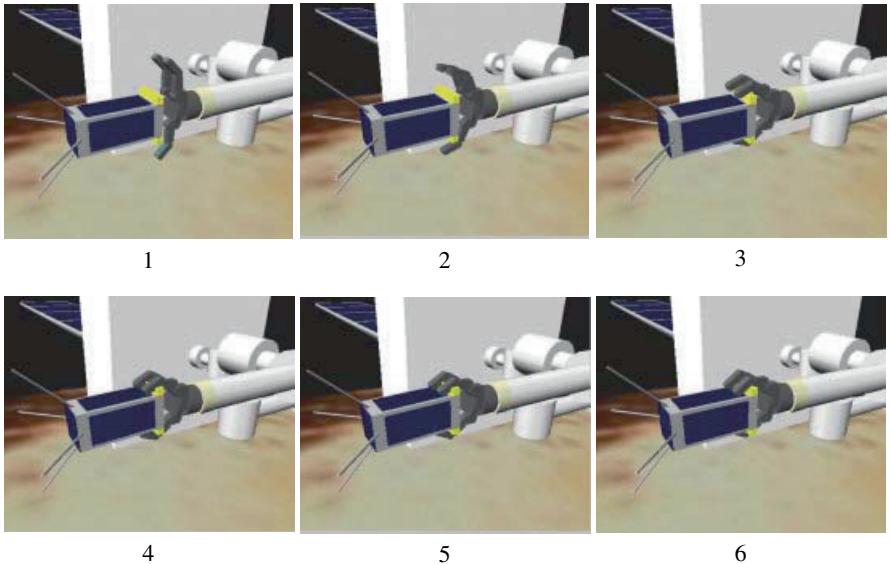


Fig. 13. Outputs of hybrid motion simulation on the CG motion display

by

$$\mathbf{G}_{sim}(s) = \mathbf{J}^{-T} \mathbf{K}_p \mathbf{J}^{-1}. \quad (7)$$

Using the equation we can calculate the interaction forces/torques \mathbf{F} from the error vector δp between the desired position vector and the current position vector of the motion table.

The calculated interaction forces/torques \mathbf{F} of which components are represented in the local coordinate frame Σ_p (see Fig. 10), are shown in Fig. 14 in dashed line. The interaction forces/torques \mathbf{F} quantified through the F/T sensor are shown in the figure in full line. The time axes in Fig. 14 are identical with the one in Fig. 12. Figure 14 shows evidently that, during the target capturing process, there is a causal relation between the impedance (stiffness) characteristics of the motion table and the interaction forces/torques quantified by the simulator.

7 Conclusions

The paper has presented a new concept of hybrid motion simulation in which most of the simulation has been done using a complex numerical model and hence, only a small but indispensable part is simulated by a real model. Thus the simulation is as flexible as a numerical one. The advantage of this type of simulation is that it exploits rapidly growing computational means for testing a real object in the simulation. A key element for such a class of simulation is a motion table that realizes motion with broad bandwidth and combines the results of the numerical simulation precisely with the motion of the physical model or vice versa. The paper has

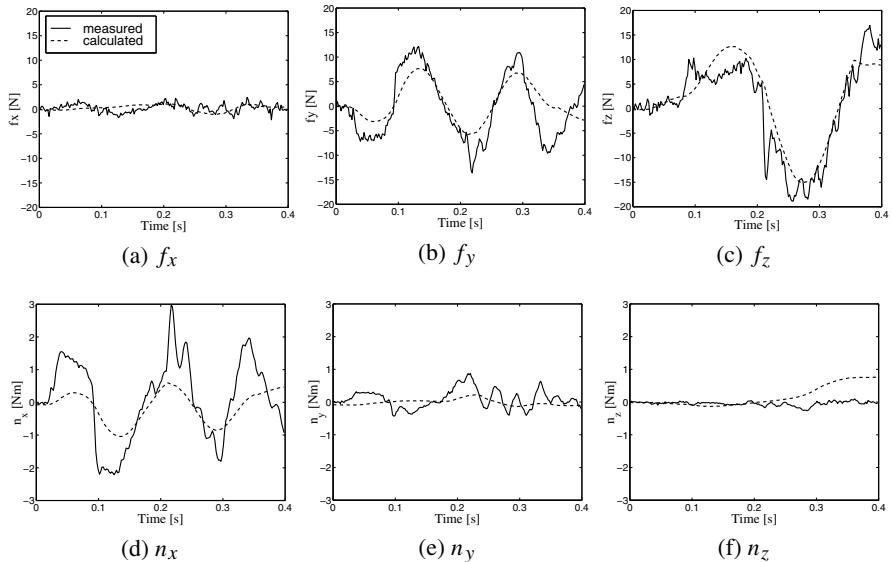


Fig. 14. Interaction forces/torques between the chaser and the target

presented a hybrid simulator that exploits a parallel structured robot effectively to implement such a high-speed motion table. Modeling of the impact dynamics in the simulator and its verification through experiments are also presented in the paper. Future research will be directed to more precise and consistent modeling of the simulation and identification of the limit of this simulation. Application of the method to more practical cases will also be promoted.

References

- Akima T, Tarao S, Uchiyama M (1999) Hybrid micro-gravity simulator consisting of a high-speed parallel robot. In Proc 1999 IEEE Int Conf on Robotics and Automation, pp 901–906.
- Clavel R (1989) Une nouvelle structure de manipulateur parallèle pour la robotique légère. APII, 23:501–519.
- Dubowsky S, Durfee W, Corrigan T, et al. (1994) A laboratory test bed for space robotics: The VES II. In Proc 1994 IEEE/RSJ Int Symp on Intelligent Robots and Systems, pp 1562–1569.
- García de Jalón J, Bayo E (1994) Kinematic and dynamic simulation of multibody systems. Springer-Verlag.
- Grimbert D, Noirault P, Marchal P, et al. (1989) DDTF improvement for more accurate space docking simulation. In Proc 2nd European In-Orbit Operations Technology Symposium, pp 239–244.
- Inohira E, Yukawa S, Akima T, et al. (1998) Design, development and evaluation of a high-speed parallel robot. In Proc 1998 JSME Conf on Robotics and Mechatronics. in Japanese.

- Kawabe H, Inohira E, Kubota T, et al. (2001) Analytical and experimental evaluation of impact dynamics on a high-speed Zero G Motion Simulator. In Proc 2001 IEEE/RSJ Int Conf on Intelligent Robots and Systems, pp 1870–1875.
- Kokkinis T, Stoughton R (1988) Optimal parallel actuation linkage for 3 DOF elbow manipulators. In The 1988 ASME Design Technology Conferences, pp 465–472.
- Merlet JP (2000) Parallel robots. Kluwer Academic Publishers.
- Pierrot F, Dauchez P, Fournier A (1991) Fast parallel robots. *J Robotic Systems*, 8:829–840.
- Pierrot F, Uchiyama M, Dauchez P, et al. (1990) A new design of a 6-DOF parallel robot. *J Robotics and Mechatronics*, 2:308–315.
- Rosenthal DE (1990) An order n formulation for robotic systems. *J Astronautical Sciences*, 38:511–529.
- Sakurai Y, Hirata H, Horie K, et al. (1997) XF-2 flight simulation tests. 35th aircraft symposium. In 35th Aircraft Symposium, pp 189–192. in japanese.
- Sawai S, Kawabe H, Asahina H, et al. (2000) MUSES-C touch-down test by robotics simulator. In 2000 ISAS 10th Workshop on Astrodynamics and Flight Mechanics, pp 286–291.
- Shimoji H, Inoue M, Inaba N, et al. (1995) Evaluation of space robot behavior using berthing dynamics simulator. *J Robotics Society of Japan*, 13:127–133.
- Stewart D (1965) A platform with six degrees of freedom. In Proc Institution of Mechanical Engineers, pp 371–386.
- Tarao S, Inohira E, Uchiyama M (2000) Motion simulation using a high-speed parallel link mechanism. In Proc 2000 IEEE/RSJ Int Conf on Intelligent Robots and Systems, pp 1584–1589.
- Uchiyama M (1994) Structures and characteristics of parallel manipulators. *Advanced Robotics*, pp 545–557.
- Uchiyama M, Iimura K, Pierrot F et al. (1992) A new design of a very fast 6-DOF parallel robot. In Proc 23rd Int Symp on Industrial Robots, pp 771–776.
- Wittenburg J (1977) Dynamics of systems of rigid bodies. B G Teubner Stuttgart.
- Yoshida K, Mavroidis C, Dubowsky S (1995) Experimental research on impact dynamics of spaceborne manipulator systems. In Experimental Robotics IV, Lecture Notes in Control and Information Sciences 233, pp 436–447. Springer-Verlag.

General Solution for Linearized Error Propagation in Vehicle Odometry

Alonzo Kelly

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213

Abstract. Although odometry is nonlinear, it yields sufficiently to linearized analysis to produce a closed-form transition matrix and a symbolic general solution for both deterministic and stochastic error propagation. Accordingly, error propagation in vehicle odometry can be understood at a level of theoretical rigor equivalent to the well-known Schuler dynamics of inertial navigation. While response to initial conditions is path-independent, response to input errors can be related to path functionals. These trajectory moments are integral transforms which functions like the moment of inertia or the Laplace transform – enabling many error propagation calculations to be performed by hand in closed-form.

1 Introduction

The problem of analytically computing the navigational error expected in odometry from a given set of sensor errors on a given trajectory seems to be both a fundamental and an unsolved problem. While a numerical solution to the problem of computing the resultant error is trivial, symbolic solutions yield dividends in the form of understanding the general case that numerical ones cannot. In this paper, the general solution is derived and validated.

1.1 Motivation

This work is motivated by a recurrent set of questions which arise for position estimation systems in mobile robots. Historically they have been answered numerically or in an ad hoc manner. How good do the sensors need to be? What kind of localization error can be expected if this particular sensor is used? Under what conditions do errors cancel out? What is the best way to calibrate the systematic or stochastic error model of this sensor? Unlike triangulation based position fixing whose error are algebraic, odometry and its errors evolve according to dead reckoning differential equations whose solutions are integrals. This essentially dynamic nature of odometry error propagation is unavoidable. However, the integral transforms derived here encapsulate the path dependent dynamic effects once the character of the trajectory is fixed. These moments thereby enable rapid analytical solutions to many of the design questions posed above.

1.2 Prior Work

The aerospace guidance community has enjoyed the benefits of a theoretical understanding of error propagation for at least five decades (Pinson, 1963). In inertial

guidance, the governing differential equations and their solutions explain the stabilizing influence of gravity – horizontal errors exhibit oscillation with the characteristic Schuler period of 84 minutes.

Likewise, the essentially geometric nature of satellite navigation system error relationships was known before the GPS satellites were in operation (Milliken, Zoller, 1978). Yet, the guidance community seems not to have provided the relevant analytical results for the land navigation systems which are typical of mobile robots – assemblies of wheel encoders, compasses, gyros, etc.

Analytical study of error propagation in mobile odometry appears only rarely in the literature. Early work in (Wang, 1988) concentrates on improving estimates for a single iteration of the estimation algorithm by incorporating knowledge of the geometry of the path followed between odometry updates. In (Borenstein, Feng, 1995), a geometric method is presented for the calibration of certain systematic errors on rectangular closed trajectories.

In (Chong, Kleeman, 1997), a recurrence equation solution is obtained for non systematic error on constant curvature trajectories. In (Nettleton, Gibbens, Durrant-Whyte, 2000), a one dimensional closed-form solution for a broader optimal estimation problem is presented. This paper presents the general solution for linearized systematic and random error propagation in odometry in the plane for any trajectory or error model.

1.3 Problem description

Odometry is a form of dead reckoning because the available measurements must be integrated to provide navigation solution. In the common “forced dynamics” formulation of odometry, the measurements, normally denoted $\underline{z}(t)$, are identified with the usual control inputs $\underline{u}(t)$ – the measurements act as a forcing function.

The state vector $\underline{x}(t)$ and input vector $\underline{u}(t)$ are:

$$\underline{x}(t) = [x(t) \ y(t) \ \theta(t)]^T \quad \underline{u}(t) = [V(t) \ \omega(t)]^T \quad (1)$$

where the state is the vehicle pose and the inputs are the linear and angular velocity.

The associated odometry equations in this case are those of the “integrated heading” case:

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} V(t) \cos \theta(t) \\ V(t) \sin \theta(t) \\ \omega(t) \end{bmatrix} \quad (2)$$

The x axis has been implicitly chosen as the heading datum as illustrated below:

Most of the notation used in subsequent sections follows that used in classic texts such as (Stengel, 1994). In particular, the operator δ will be used to signify perturbation and a fairly standard set of conventions of linear systems theory and

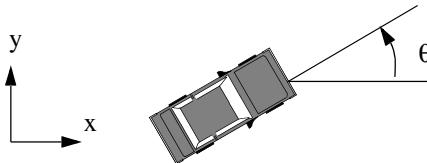


Fig. 1. Coordinates For Odometry

optimal estimation will be used. Generally, an observer equation can be introduced to model how sensor indications project onto the inputs and states and the general situation is described by:

$$\begin{aligned}\underline{x}(t) &= f(\underline{x}(t), \underline{u}(t)) \\ \underline{z}(t) &= h(\underline{x}(t), \underline{u}(t))\end{aligned}\quad (3)$$

Many alternative formulations of odometry are possible. This one and many others have the key properties of being homogeneous in the inputs, nonlinear in the states, and reduced to echelon form. Due to the latter property, the general solution is immediate:

$$\begin{aligned}\theta(t) &= \theta(0) + \int_0^t \omega(t) dt \\ x(t) &= x(0) + \int_0^t V(t) \cos \theta(t) dt \\ y(t) &= y(0) + \int_0^t V(t) \sin \theta(t) dt\end{aligned}\quad (4)$$

Closed form solutions to these equations are nevertheless available only for the simplest of inputs and nonlinear error propagation is equally intractable.

This paper addresses the following problem. Let the inputs (or equivalently, the sensors) to the system be corrupted by additive errors $\delta \underline{u}(t)$ as follows:

$$\underline{u}'(t) = \underline{u}(t) + \delta \underline{u}(t) \quad (5)$$

Using these input errors and the system dynamics, determine the behavior of the associated errors $\delta \underline{x}(t)$ in the computed vehicle pose:

$$\underline{x}'(t) = \underline{x}(t) + \delta \underline{x}(t) \quad (6)$$

The errors can be systematic or random in nature, and solutions for either case are sought.

2 Linearized Error Dynamics

Here, the governing equations of odometry error dynamics are developed and described.

2.1 Linearization

Perturbative techniques linearize nonlinear dynamical systems in order to study their first order behavior. With reference to (3), several Jacobian matrices are defined which may depend on the state and the input, and are evaluated on some reference trajectory:

$$\begin{aligned} F(t) &= \left. \frac{\partial}{\partial \underline{x}} f \right|_{\underline{x}, \underline{u}} & G(t) &= L(t) = \left. \frac{\partial}{\partial \underline{u}} f \right|_{\underline{x}, \underline{u}} \\ H(t) &= \left. \frac{\partial}{\partial \underline{x}} h \right|_{\underline{x}, \underline{u}} & M(t) &= N(t) = \left. \frac{\partial}{\partial \underline{u}} h \right|_{\underline{x}, \underline{u}} \end{aligned} \quad (7)$$

To be consistent with common notation, $G(t)$ will be used when discussing systematic error and $L(t)$ when discussing random error. $M(t)$ and $N(t)$ will be used similarly.

Equation 3 is linearized as follows:

$$\begin{aligned} \delta \underline{x}(t) &= F\{\underline{x}(t), \underline{u}(t)\}\delta \underline{x}(t) + G\{\underline{x}(t), \underline{u}(t)\}\delta \underline{u}(t) + L\{\underline{x}(t), \underline{u}(t)\}\delta \underline{w}(t) \\ \delta \underline{z}(t) &= H\{\underline{x}(t), \underline{u}(t)\}\delta \underline{x}(t) + M\{\underline{x}(t), \underline{u}(t)\}\delta \underline{u}(t) + N\{\underline{x}(t), \underline{u}(t)\}\delta \underline{w}(t) \end{aligned} \quad (8)$$

Parameters can be considered to be constant inputs. Subsequently, notational dependence on the state and the input will be suppressed for brevity but all of these matrices generally depend on both. These provide a linear approximation to the propagation of systematic error as well as a description of the mean of random error.

A second input vector $\delta \underline{w}(t)$ has been introduced to differentiate systematic from random error sources. $\delta \underline{w}(t)$ is simply the component of the original $\delta \underline{u}(t)$ which is random. By superposition, systematic and random error sources can be treated independently.

For the random errors $\delta \underline{w}(t)$, (8) is used only in a heuristic sense in stochastic calculus because direct integration of random signals is beyond the scope of traditional calculus.

2.2 Deterministic case

For deterministic error, the linearized dynamics take the form of:

$$\begin{aligned} \delta \underline{x}(t) &= F(t)\delta \underline{x}(t) + G(t)\delta \underline{u}(t) \\ \delta \underline{z}(t) &= H(t)\delta \underline{x}(t) + M(t)\delta \underline{u}(t) \end{aligned} \quad (9)$$

The first is the linear perturbation equation. The second, the linearized observer, is not a differential equation. It can be solved and substituted into the state equations to produce an equation of the same form as the original perturbation equation. It can be dispensed with for the balance of the paper and the deterministic case can be considered to be defined by the linear perturbation equation:

$$\delta \underline{x}(t) = F(t)\delta \underline{x}(t) + G(t)\delta u(t) \quad (10)$$

Once the matrices are filled in, and the error models are assumed, it is instructive to evaluate (10) for motion dependence, reversibility, etc.

2.3 Stochastic case

For practical treatment of random errors, the second moment or “covariance” of the error is considered and the state covariance and input spectral density matrices are defined:

$$\begin{aligned} P &= \text{Exp}(\delta \underline{x}(t)\delta \underline{x}(t)^T) \\ Q &= \text{Exp}(\delta \underline{w}(t)\delta \underline{w}(\tau)^T\delta(t - \tau)) \end{aligned} \quad (11)$$

In the second case, the Dirac delta signifies that the random sequence $\delta \underline{w}(t)$ is white and that the units of Q are a time rate of covariance. The solution for linear variance propagation is derived in several texts including (Stengel, 1994):

$$\begin{aligned} P(t) &= F(t)P(t) + P(t)F(t)^T + L(t)Q(t)L(t)^T \\ R(t) &= H(t)P(t)H(t)^T + M(t)Q(t)M(t)^T \end{aligned} \quad (12)$$

The first is the linear variance equation. The second, the linear stochastic observer, is not a differential equation. It can be substituted into the state equations to produce an equation of the same form as the original variance equation. It can be dispensed with for the balance of the paper and the stochastic case can be considered to be defined by the linear variance equation:

$$P(t) = F(t)P(t) + P(t)F(t)^T + L(t)Q(t)L(t)^T \quad (13)$$

3 Solution to Linearized Systems

The equations to be solved are (10) and (13). it is well known that the solution to these equations rests on the knowledge of a very important matrix called the

transition matrix, denoted $\Phi(t, \tau)$. One definition for this matrix is that it is the solution to:

$$\frac{d}{dt} \Phi(t, \tau) = F(t) \Phi(t, \tau) \quad (14)$$

The general solutions for the propagation of systematic and random error are respectively of the form of the vector and matrix convolution integrals:

$$\begin{aligned} \delta \underline{x}(t) &= \Phi(t, t_0) \underline{x}(t_0) + \int_{t_0}^t \Phi(t, \tau) G(\tau) \delta \underline{u}(\tau) d\tau \\ P(t) &= \Phi(t, t_0) P(t_0) \Phi^T(t, t_0) + \int_{t_0}^t \Phi(t, \tau) L(\tau) Q(\tau) L^T(\tau) \Phi^T(t, \tau) d\tau \end{aligned} \quad (15)$$

Once a trajectory and an error model are assumed, the only unknown in both of these equations is the transition matrix.

Error propagation behavior The important behaviors of error propagation are evident from the structure of these solutions. Both solutions consist of a state (initial conditions $\underline{x}(t_0)$ or $P(t_0)$) response and an input ($\delta \underline{u}(\tau)$ or $Q(\tau)$) response. The state response is always path independent and hence it vanishes on any closed trajectory. Perhaps surprisingly, this means both general solutions can exhibit extrema and even zeros.

Although the input response may be path dependent or otherwise vanish under conditions of symmetry, it is generally path dependent. In other words, error propagation is a functional defined on the input and it can, in certain cases, be expressed as a functional on the reference state space trajectory. The integrand in the systematic case is signed and can exhibit zeros. In the stochastic case, the integrand is positive semidefinite and its contribution is nondecreasing.

Input transition matrix It is very useful to define the (potentially nonsquare) input transition matrix as:

$$\tilde{\Phi}(t, \tau) = \Phi(t, \tau) L(\tau) = \Phi(t, \tau) G(\tau) \quad (16)$$

This matrix maps a given systematic or random error at time τ onto its net effect on the resultant state error occurring later at time t . Linearization for the purposes of studying error propagation amounts to treating errors occurring at different times independently of each other – their compounded effects being higher order. This matrix is the defining matrix for each form of odometry – capturing both the effects of system dynamics and of the input measurement errors.

Influence matrices Let \underline{i} denote the i th column of the input transition matrix. Notice that for a given element δu_i of $\delta \underline{u}$, its contribution to the solution integrands in (15) is:

$$\begin{aligned} d[\delta \underline{x}(t, \tau)] &= \underline{i} \tilde{\Phi} \delta u_i d\tau \\ dP(t, \tau) &= q_{ij} \left(\underline{i} \underline{j}^T \right) d\tau \end{aligned} \quad (17)$$

Hence, the influence vectors \underline{i} define the projection of each individual element of the input (measurement) error vector onto the entire output (state) error vector.

Similarly, the outer product influence matrices $\underline{i} \underline{j}^T$ define the projection of each element of the input (measurement) covariance matrix onto the entire output (state) covariance matrix.

Clearly:

$$\underline{i} \underline{i}^T = \text{symmetric} \quad \underline{i} \underline{j}^T = \left(\underline{i} \underline{j}^T \right)^T$$

In terms of influence matrices, (15) can be written as:

$$\delta \underline{x}(t) = \Phi(t, t_0) \delta \underline{x}(t_0) + \int_{t_0}^t \left(\sum_i \underline{i} \delta u_i \right) d\tau \quad (18)$$

$$P(t) = \Phi(t, t_0) P(t_0) \Phi^T(t, t_0) + \int_{t_0}^t \left(\sum_i \sum_j q_{ij} \left(\underline{i} \underline{j}^T \right) \right) d\tau$$

The columns of the input transition matrix constitute a basis for the time derivative of error because the result for any error source must reside in the column space of $\tilde{\Phi}(t, \tau)$.

Trajectory moment matrices The order of integration and summation can be reversed when convenient. Also, when the errors are constant or can be rendered so under a change of variable, they can be moved outside the integrals to produce:

$$\delta \underline{x}(t) = \Phi(t, t_0) \delta \underline{x}(t_0) + \sum_i \left[\int_{t_0}^t \underline{i} d\tau \right] \delta u_i \quad (19)$$

$$P(t) = \Phi(t, t_0) P(t_0) \Phi^T(t, t_0) + \sum_i \sum_j \left[\int_{t_0}^t \underline{i} \underline{j}^T d\tau \right] q_{ij}$$

This is the trajectory moment form of the error propagation equations. The two expressions in square brackets are respectively the trajectory moment vector and the trajectory moment matrix. Both are intrinsic properties of the trajectory – independent of the form of the errors. Their role in estimation is similar to the role of the Laplace Transform in control, the Fourier transform in circuits, the moment of inertia in mechanics and the moments of probability distributions (the variance is the second probabilistic moment).

When errors have simple forms, the effects of errors and trajectories can be decoupled, and the latter can be tabulated for specific trajectories and used like any other integral transform to convert problems in differential equations into equivalent algebraic ones.

These results represent the structure of the solution as a superposition very directly:

- Errors are always expressible as the sum of the isolated effects of each individual error source.
- Each effect is an integral transform of the trajectory weighted by the magnitude of the associated error source.
- The integral transform accounts for system dynamics and input transforms. It is an integral of the columns or outer products of all pairs of columns of the input transition matrix.

3.1 Solution for commutative dynamics

Given that the system Jacobian of odometry is a time varying matrix, linear systems theory shows that the transition matrix exists, but generally provides little guidance in finding it. Luckily, odometry is a special case.

Consider the following particular matrix exponential of a definite integral of the system Jacobian:

$$\Psi(t, \tau) = \exp[R(t, \tau)] = \exp\left(\int_{\tau}^t F(\xi)d\xi\right) = \exp[R(t, \tau)] \quad (20)$$

where the matrix exponential is defined as usual by the infinite matrix power series:

$$\exp(A) = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$$

When this exponential commutes (Brogan, 1974) with the system dynamics matrix thus:

$$\Psi(t, \tau)F(t) = F(t)\Psi(t, \tau) \quad (21)$$

then every power of $R(t, \tau)$ also commutes with and we can write the derivative as:

$$\frac{d}{dt} \Psi(t, \tau) = \frac{d}{dt} \left\{ I + R(t, \tau) + \frac{R(t, \tau)^2}{2!} + \frac{R(t, \tau)^2}{3!} + \dots \right\} = F(t) \Psi(t, \tau) \quad (22)$$

So it satisfies equation 14 and therefore it is the transition matrix which solves the associated time-varying linear system. This property of commutative dynamics is the key to generating a general solution.

4 Application To Odometry

This section derives the error propagation equations for one common form of odometry – integrated heading. Extensions to cases where a compass is used (direct heading) or where angular velocity is derived from two wheel velocities (differential heading) are very similar.

In integrated heading odometry, an angular velocity indication is available and a heading state is determined by integrating it. For example, a gyro could be used to measure heading rate and a transmission encoder, groundspeed radar, or fifth wheel encoder could be used to measure the linear velocity of the vehicle. This is the case given in (3):

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} V(t) \cos \theta(t) \\ V(t) \sin \theta(t) \\ \omega(t) \end{bmatrix} \quad (23)$$

Where:

$$\begin{aligned} \underline{x}(t) &= [x(t) \ y(t) \ \theta(t)]^T \\ \underline{u}(t) &= [V(t) \ \omega(t)]^T \\ \delta \underline{x}(t) &= [\delta x(t) \ \delta y(t) \ \delta \theta(t)]^T \\ \delta \underline{u}(t) &= [\delta V(t) \ \delta \omega(t)]^T \end{aligned} \quad P(t) = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{y\theta} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_{\theta\theta} \end{bmatrix} \quad Q(t) = \begin{bmatrix} \sigma_{vv} & \sigma_{v\omega} \\ \sigma_{v\omega} & \sigma_{\omega\omega} \end{bmatrix} \quad (24)$$

The Jacobians are:

$$F(t) = \begin{bmatrix} 0 & 0 & -Vs\theta \\ 0 & 0 & Vc\theta \\ 0 & 0 & 0 \end{bmatrix} \quad G(t) = \begin{bmatrix} c\theta(t) & 0 \\ s\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \quad (25)$$

So the linearized error dynamics are:

$$\frac{d}{dt} \begin{bmatrix} \delta x(t) \\ \delta y(t) \\ \delta \theta(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & -Vs\theta \\ 0 & 0 & Vc\theta \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta x(t) \\ \delta y(t) \\ \delta \theta(t) \end{bmatrix} + \begin{bmatrix} c\theta(t) & 0 \\ s\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta V(t) \\ \delta \omega(t) \end{bmatrix}$$

The rest of the matrices leading to the input transition matrix are therefore as follows:

$$R(t, \tau) = \int_{\tau}^t F(\zeta) d\zeta = \int_{\tau}^t \begin{bmatrix} 0 & 0 & -Vs\theta \\ 0 & 0 & Vc\theta \\ 0 & 0 & 0 \end{bmatrix} d\zeta = \begin{bmatrix} 0 & 0 & -\Delta y(t, \tau) \\ 0 & 0 & \Delta x(t, \tau) \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Phi(t, \tau) = \exp[R(t, \tau)] = \begin{bmatrix} 1 & 0 & -\Delta y(t, \tau) \\ 0 & 1 & \Delta x(t, \tau) \\ 0 & 0 & 1 \end{bmatrix} \quad (26)$$

$$\tilde{\Phi}(t, \tau) = \Phi(t, \tau) G(\tau) = \begin{bmatrix} c\theta(\tau) & -\Delta y(t, \tau) \\ s\theta(\tau) & \Delta x(t, \tau) \\ 0 & 1 \end{bmatrix}$$

The transition matrix follows from the fact that $R^2(t, \tau) = 0$ in this case. Again $\Psi(t, \tau)$ clearly satisfies (21), so $\Phi(t, \tau)$ was written in place of $\Psi(t, \tau)$. The following shorthand expressions for the coordinates of the endpoint from the perspective of the point $[x(\tau), y(\tau)]$ have also been used:

$$\Delta x(t, \tau) = [x(t) - x(\tau)] \quad (27)$$

$$\Delta y(t, \tau) = [y(t) - y(\tau)]$$

Integrals involving these quantities in particular must be manipulated with slightly more care to preserve the distinction between t (the endpoint of the trajectory) and τ (the variable of integration). Substituting into the general solution in (15) gives:

$$\underline{\delta x}(t) = IC_d + \int_0^t \begin{bmatrix} c\theta & -\Delta(t, \tau) \\ s\theta & \Delta x(t, \tau) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta V(\tau) \\ \delta \omega(\tau) \end{bmatrix} d\tau$$

$$P(t) = IC_s + \int_0^t \begin{bmatrix} c\theta & -\Delta y(t, \tau) \\ s\theta & \Delta(t, \tau) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{vv} & \sigma_{v\omega} \\ \sigma_{v\omega} & \sigma_{\omega\omega} \end{bmatrix} \begin{bmatrix} c\theta & -\Delta(t, \tau) \\ s\theta & \Delta x(t, \tau) \\ 0 & 1 \end{bmatrix}^T d\tau$$

Note, for example, that the translational error resulting from gyro bias is proportional to the product of excursion $\Delta x(t, \tau)$ and time. It is easy to show that the response to velocity scale errors is path independent. The initial state response for this case is:

$$\underline{IC}_d = \begin{bmatrix} 1 & 0 & -y(t) \\ 0 & 1 & x(t) \\ 0 & 0 & 1 \end{bmatrix} \delta \underline{x}(0) = \begin{bmatrix} \delta x(0) \\ \delta y(0) \\ 0 \end{bmatrix} + \begin{bmatrix} -y(t)\delta\theta(0) \\ x(t)\delta\theta(0) \\ \delta\theta(0) \end{bmatrix}$$

$$\underline{IC}_S = \begin{bmatrix} 1 & 0 & -y(t) \\ 0 & 1 & x(t) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{xx}(0) & \sigma_{xy}(0) & \sigma_{x\theta}(0) \\ \sigma_{xy}(0) & \sigma_{yy}(0) & \sigma_{y\theta}(0) \\ \sigma_{x\theta}(0) & \sigma_{y\theta}(0) & \sigma_{\theta\theta}(0) \end{bmatrix} \begin{bmatrix} 1 & 0 & -y(t) \\ 0 & 1 & x(t) \\ 0 & 0 & 1 \end{bmatrix}^T$$

This result is the general linearized solution for the propagation of systematic and random error in 2D integrated heading odometry for any trajectory and any error model.

5 Validation

Error propagation results were verified on the integrated heading case by comparing the linearized solutions of the paper with an exact nonlinear numerical solution for both systematic and random errors. The input error characteristics were as follows:

Table 1. Error Sources for Theory Validation

Error Source	Deterministic	Random
Linear Velocity	$\delta V = \alpha V$ $\alpha = 0.05$	$\sigma_v = \delta V$
Angular Velocity	$\delta\omega = b$ $b = 30^\circ/\text{hr}$	$\sigma_\omega = \delta\omega$

These models represent a systematic scale error of 5% on velocity and a motion dependent random walk stochastic velocity error of equal standard deviation. A systematic gyro bias of 30 degrees/hr. is used as well as a bias stability of equal standard deviation. Such error magnitudes are considerably larger than might be expected in a practical situation. The intention here is to stress the linearity assumption and provide a common error magnitude for both systematic and random sources which is large enough to be noticeable in the following figures.

In the systematic case, straightforward numerical quadrature can be used to integrate the nonlinear dynamics in both the perturbed and unperturbed case and the difference between the two is obtained as the exact nonlinear solution within the limits of time discretization.

For stochastic error, Monte Carlo simulation was used. Using 500,000 independent, unbiased, Gaussian random variables, 250 discrete time random signals for

linear and angular velocity errors were generated and used to corrupt the nominal inputs.

Figure 2 illustrates the result of nonlinear simulation of an arbitrary trajectory chosen to contain a loop but exhibit no particular symmetry. The velocity for the test is 0.25 m/s, the total time is 210 secs and the time step is 0.5 secs.

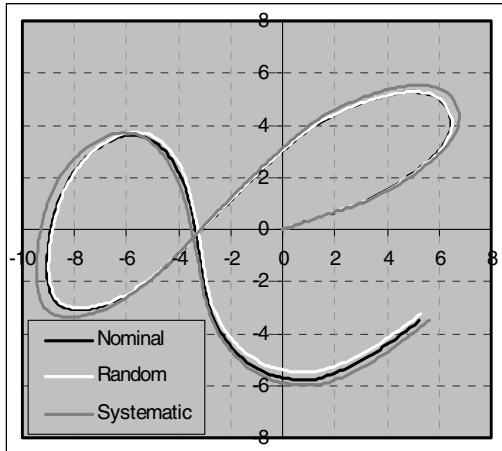


Fig. 2. Test trajectory

While there is a single systematic result to plot, only one representative of the 250 randomly perturbed trajectories is shown. Due to the tendency of random error to cancel in the short term, the effect of systematic error is locally more dramatic when compared to a random error. The effect of the systematic velocity scale error is evident in the increase in error magnitude with the radius from the origin. As predicted by theory, substantial accumulated systematic errors in the far left vanish when the loop closes. For random error, the overall growth rate is more subdued but it nonetheless accumulates to nontrivial levels over time.

The difference between linearized and nonlinear deterministic error is shown in Fig. 3.

Clearly, the linearized solution is an excellent approximation even for errors of this magnitude. The difference between exact and linearized solutions barely exceeds 1 cm or 2.5% of the actual maximum error magnitude of 40 cm. This result validates the systematic part of (28).

The results of the Monte Carlo simulation are provided in Fig. 4 for the translational variances and co-variances. Rotational variance is linear by construction and translational-rotational covariances (not shown) agree similarly with theory.

The agreement between theory and simulation is excellent. Overall, three classes of error can be expected in the stochastic case: linearization, discretization, and sampling error. The first two classes have been demonstrated to be relatively small

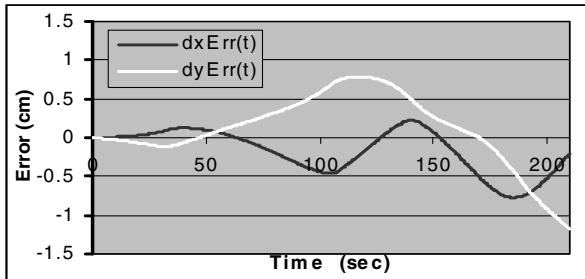


Fig. 3. Deterministic linearization error

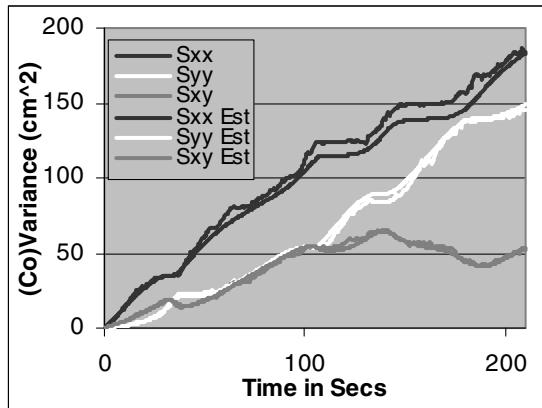


Fig. 4. Monte Carlo simulation compared to theory

by the systematic error results. These curves show that for a sample size of 250 pairs of input random processes, the sample variance of the state covariance matrix tracks the linearized theoretical population variance quite well. This result validates the stochastic part of (28).

6 Conclusions

While the general nonlinear solution of the equations of odometry in standard form is straightforward, it is only integrable in closed-form in the most trivial case of a constant heading error. The closed-form general solution to the linearized error dynamics also exists and it is integrable for a larger class of input errors.

When the equations are expressed in a forced dynamics manner and linearized, an (exponential of an integral) expression for the transition matrix is available for commonly encountered forms of odometry. The integral is often evaluable in closed-form and the upper-triangular nature of the system Jacobian implies that the infinite series expression for the exponential also terminates quickly. Subsequent applica-

tion of the theory of linear systems provides the integrals for error propagation in closed form.

Resultant state estimation error is always a combination of the state response and the input response. The former is always path independent and vanishes on any closed trajectory. The latter can often be reduced to expressions involving path functionals or moments which are analogous to the moments of mechanics and the Laplace transforms of control theory.

Trajectory moments are responsible for the path dependent behavior of odometry error. Over time, error evolves as the superposition of the trajectory moments corresponding to each input errors source.

Analytic expressions are important tools in the development of theory. The present results enable the symbolic application of linear systems theory, optimal control, calculus of variations, etc. to any application which attempts to account in some way for error propagation.

In addition to their pedagogic value, these results can be used in design to determine acceptable levels of sensor error. They can be used in development to accentuate response to individual error sources for on-line or off-line calibration or evaluative purposes. They can be used in operation to plan trajectories in order to minimize exposure to specific error sources, or determine optimal approaches to error compensation. The availability of an analytical theory of error propagation enables many problems to be solved rapidly by referring to a table of trajectory moments.

References

- Borenstein J, Feng L (1995) Correction of systematic odometry error in mobile robots. In Proceedings of IEEE/RSJ International Conference on Robotics and Systems (IROS 95), Pittsburgh, PA.
- Brogan WL (1974) Modern control theory. Quantum, New York, NY.
- Chong KS, Kleeman L (1997) Accurate odometry and error modelling for a mobile robot. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA 1997), Albuquerque, New Mexico.
- Milliken RJ, Zoller CJ (1978) Principle of operation of navstar and system characteristics. *Navigation*, 25(2):95–106.
- Nettleton EW, Gibbens PW, Durrant-Whyte HF (2000) Closed form solutions to the multiple platform simultaneous localization and map building (slam) problem. In Proceedings of AeroSense 2000, Orlando FL, USA.
- Pinson JC (1963) Inertial guidance for cruise vehicles. In Londes CT (ed) Guidance and Control of Aero-space Vehicles. McGraw-Hill, New York.
- Stengel R (1994) Optimal control and estimation. Dover, New York.
- Wang CM (1988) Location estimation and uncertainty analysis for mobile robots. In Proc IEEE Conference on Robotics and Automation (ICRA 88), pp 1230–1235.

Probabilistic Adaptive Agent Based System for Dynamic State Estimation using Multiple Visual Cues

Alvaro Soto and Pradeep Khosla

Robotics Institute, Carnegie Mellon University

Abstract. Most of current machine vision systems suffer from a lack of flexibility to account for the high variability of unstructured environments. Here, as the state of the world evolves the information provided by different visual attributes changes, breaking the initial assumptions of the vision system. This paper describes a new approach for the creation of an adaptive visual system able to selectively combine information from different visual dimensions. Using a probabilistic approach and uncertainty metrics, the system is able to take appropriate decisions about the more relevant visual attributes to consider. The system is based on an intelligent agent paradigm. Each visual algorithm is implemented as an agent, which adapts its behavior according to uncertainty considerations. The proposed system aims to achieve robustness and efficiency. By combining the outputs of multiple vision modules the assumptions and constraints of each module are factored out resulting in a more robust system. Efficiency is achieved through the online selection and specialization of the agents. An implementation of the system for the case of human tracking showed encouraging results.

1 Introduction

As the state of the art of computing technology is advancing, providing more powerful and affordable machines, computers are becoming widely used in diverse aspects of modern society. As computers start to perform new types of tasks in less structured and less predictable environments, there is an increasing need to provide them with a higher degree of awareness about the changing conditions of their virtual or natural surrounding.

In particular, the case of visual perception is a very attractive option to capture information from a natural surrounding. In contrast to other sensor modalities, vision can allow to perceive a large number of different features of the environment such as color, shape, depth, motion, and so on. This multi-dimensionality of visual information is one of the key strength that explains the great robustness observed in most advanced biological visual systems (Zeki, 1993).

Most current machine vision systems show a lack of flexibility to consider the wide variety of information provided by visual data. The typical approach relies on simplifications of the environment or on good engineering work to identify relevant visual attributes that allow solving a specific visual task. For example, consider the case of a robot localization system based on artificial visual landmarks. In this case, previous knowledge about the visual appearance of the landmarks provides strong constraints that allow constructing algorithms especially designed to detect the key visual attributes (Nourbakhsh, Bobenage, Grange, Lutz, Meyer, Soto, 1999).

The main problem with this approach is a lack of flexibility to account for the great variability of most dynamic unconstrained environments. Problems such as changes in the field of view, partial occlusion, changes in illumination, or different postures constantly modify the information provided by the different visual attributes. As a consequence, there is a high variability about the more adequate set of visual attributes to extract the knowledge needed to complete a task.

As an example, consider a stationary visual system designed to track people using an intensity based background subtraction algorithm (Kanade, Collins, Burt, Wixson, 1998). Under a slowly varying background the system will be very robust. Now in situations of heavy moving shadows, heavy wind, or people wearing clothes similar to the background, the system will perform poorly. In these cases an algorithm based on segmentation using depth information (Darrell, Gordon, Harville, Woodfill, 1998) can help to reduce the ambiguities, but at the expense of extra processing time. Now, if the system needs also to keep the identity of the targets, an algorithm based on color information can provide key information to identify each target, especially if they exhibit different colors. For the case of similar colors, it is possible that an algorithm based on shape or texture information can help to reduce any further ambiguities.

The key observation in the previous example is that as the state of the world evolves, the potential knowledge provided by different visual attributes can change dramatically, breaking the initial assumptions of the vision system. So in order to keep a balance between robustness and efficiency, it is crucial to incorporate suitable adaptation mechanisms that allow to combine and to select the more appropriate set of visual features.

In this paper we propose a new probabilistic approach for the creation of an adaptive visual system able to selectively combine information from different visual algorithms. The basic scenario is an agent embedded in an unpredictable and dynamic environment. The agent is able to receive different types of visual information from the environment. As new information arrives, the goal of the agent is to select the more adequate information in order to update the knowledge about its relevant part of the world.

The state of the world is characterized using a state space representation. For example, for the case of visual tracking of a single target, the state of the system is represented by 4 state variables (x, y, w, h) which determine a bounding box surrounding the target: (x, y) represent the center of the box in the image plane, (w) its width, and (h) its height. The goal of the system is to keep track of a joint probability density function (jpdf) over these state variables. The level of uncertainty in this state estimation is the key element used by the system to select the more adequate visual information sources. For example, if we are tracking a target using color information and there is a high level of uncertainty about the position of the target, the system will automatically activate new sources of visual information to reduce the current ambiguities.

Using the power of probability theory for representing and reasoning under uncertainty, and elements from information theory to lead the inference engine to

prominent hypotheses and information sources, the proposed system aims to achieve robustness and efficiency. By combining the outputs of multiple visual routines the assumptions and constraints of each module are factored out resulting in a more robust system. Efficiency is achieved through the online selection and specialization of the agents according to the quality of the information provided by them.

The research proposed in this work is particularly relevant for the case of dynamic visual tasks with a high variability about the subsets of visual attributes that can characterize relevant visual structures. This includes visual tasks such as dynamic target tracking, obstacle detection, and identification of landmarks in natural scenes. In particular, the advantages of the approach proposed here are demonstrated for the case of human target tracking using a mobile robot.

This paper is organized as follows. Section 2 describes our approach and its main components. Section 3 presents related work. Section 4 describes an implementation of the intended system. Section 5 describes the results of the implementation. Finally, section 6 presents relevant conclusions.

2 Probabilistic Adaptive Agent Based System

The system is based on an intelligent agent paradigm. Each visual information source is implemented as an agent that is able to adapt its behavior according to the relevant task and environment constraints. The adaptation is provided by local self-evaluation functions on the agents. These functions are based on considerations about the level of uncertainty present at each time in the state estimation. Cooperation among the agents is given by a probabilistic scheme that integrates the evidential information provided by them.

2.1 Intelligent Agents

There is a general agreement that the main features that distinguish an intelligent agent are *autonomy, sociability, and adaptation* (Jennings, Wooldridge, 1995). Autonomy provides the independence that allows the agent to exhibit an opportunistic behavior in agreement with its goals. Sociability provides the communication skills that allow the agent to interact with other artificial agents and humans. Adaptation provides the flexibility that allows the agent to change its behavior according to the conditions of the environment.

This work makes use of multiple agents that can simultaneously analyze different dimensions of the incoming information. Each agent is implemented as an independent visual algorithm. In this sense the agents act as a group of experts where each agent has a specific knowledge area.

The rest of section 2 describes the probabilistic representation used in this work, and how the system provides the agents with sociability and adaptation mechanisms. Autonomy is given by a distributed multi-agent software architecture briefly described in section 4.

2.2 Probabilistic Representation

In this work we use a probabilistic approach, which allows accounting for the inherent ambiguity of most unconstrained scenarios. The basic idea is to keep track of a probability distribution over a set of possible hypotheses. Each hypothesis represents a possible configuration of the objects of interest, for example the (x, y, w, h) state of a target.

In this work we use a Bayesian framework for reasoning under uncertainty. Assuming that at time t the current visual evidence e_t can be totally explained by the current hypothesis h_t , and that the dynamic of the system follows a first order Markov process, it is possible to obtain (1), which is the standard way to perform Bayesian inference for the dynamic case.

$$P(h_t / \vec{e}_t) = \beta * P(e_t / h_t) * \sum_{h_{t-1}} P(h_t / h_{t-1}) * P(h_{t-1} / \vec{e}_{t-1}) \quad (1)$$

Where \vec{e}_t contains all the historic evidence until time t , and β corresponds to a normalizing factor.

Equation (1) is a recursive formulation that requires knowledge about the observation model $P(e_t / h_t)$ and the system dynamics $P(h_t / h_{t-1})$. The observation model or likelihood function evaluates the fitness between each sample hypothesis and the observations. The system dynamics determines the level of exploration for new hypotheses as the system evolves.

In practice, except for the case of some finite state-space Hidden Markov models, full Bayesian inference is only possible when the models have suitable analytical expressions. The more typical case is linear-gaussian models. For this case, the pdf over the possible states of the system (state-pdf) remains Gaussian at all times, and the well-known Kalman Filter gives the optimal solution. For the case of nonlinear models, it is possible to use the Extended Kalman Filter, but still under a Gaussian assumption.

The Gaussian assumption severely limits the use of Bayesian inference for state estimation. High ambiguity is one of the inherent features that emerges in most unstructured environments. In this case the state-pdf can have a complex multi-modal shape that cannot be accurately modelled by a Gaussian density. Fortunately stochastic sampling provides an alternative and efficient estimation approach for these cases.

In stochastic sampling a pdf is represented through a set of samples, each with an associated weight representing its probability. The great advantage is that it is possible to approximate any functional non-linearity and system or measurement noise. In this paper we approximate (1) using a particle filter approach, also known in the literature as Bootstrap filter (Gordon, Salmon, Smith, 1993), Condensation algorithm (Isard, Blake, 1996), or sequential Monte Carlo (Kitagawa, 1996). Figure 1 shows pseudo code for the operation of the algorithm. Starting from an initial approximation of the state-pdf given by n sample hypothesis h_i , the algorithm uses the

system dynamic $P(h_i / h_{i-})$ and its current belief B_t to propagate the more prominent hypotheses. Then these candidate hypotheses are weighted according to the support received by new incoming evidence e_t represented as a likelihood function $P(e_t / h_i)$. The nice feature about the particle filter is the dynamic allocation of the sample hypotheses h_i according to the current belief. This helps to reduce the problem of sample depletion, and allows a great efficiency in the representation.

```

At t = 0
  Sample  $h_i |_{i=1}^n$  from initial prior  $P(h)$ 
  set  $B_t = \{ \pi_i, h_i \}_{i=1}^n$  with  $\pi_i = P(h_i)$ 

For t = 1,2,...N
  For i = 1 to n
    Sample  $h_{i-} \sim B_t$ 
    Sample  $h_i \sim P(h_i / h_{i-})$ 
    Evaluate  $\pi_i = P(e_t / h_i)$ 
  end
  Normalize weights  $\pi_i = \frac{\pi_i}{\sum_i \pi_i}$ 
  Set new  $B_t = \{ \pi_i, h_i \}_{i=1}^n$ 
end

```

Fig. 1. Pseudo code for particle filter algorithm

2.3 Sociability

The integration of information is performed using Bayes nets. Figure 2 shows the typical tree structure of the Bayes nets relevant to this work. Agent nodes directly measure different dimensions of the incoming visual information, such as color or shape. Abstraction nodes allow the integration of information and the updating of the state representation. Also, abstraction nodes allow introducing conditional independence relations among the visual agents.

Figure 2 can be considered as a hierarchical representation of the simpler case of just one abstraction node. In this case (1) can be expressed as:

$$\begin{aligned}
 P(h_t / \overrightarrow{e_{1,t}}, \overrightarrow{e_{2,t}}, \dots) &= \beta * \overbrace{P(e_{1,t} / h_t)}^{Agents} * \overbrace{P(e_{2,t} / h_t)}^{Agents} * \dots, \\
 &\dots * \sum_{h_{t-1}} \underbrace{P(h_t / h_{t-1}) * P(h_{t-1} / \overrightarrow{e_{1,t-1}}, \overrightarrow{e_{2,t-1}}, \dots)}_{\text{Abstraction node}} \quad (2)
 \end{aligned}$$

Where $\overrightarrow{e_{i,t}}$ corresponds to the historic evidence provided by agent i until time t .

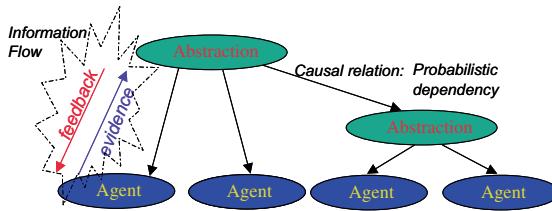


Fig. 2. Bayes Net

Equation (2) shows the decoupling between the evidence provided by each agent through a likelihood function, and the state updating performed by the abstraction node. The abstraction node acts as a central inference that keeps track of the state estimation represented by a set of sample hypotheses and their probabilities. Using the current estimations and the system dynamics, the abstraction node decides which hypotheses need further considerations and it sends this new set of hypotheses to each of the agents. According to its own local information sources, each agent evaluates the supporting evidence for each hypothesis, and it sends this evidence back to the abstraction node as a likelihood function. Finally the abstraction node uses this information to update its beliefs about the state of the world, and it starts a new iteration of the state estimation by predicting a new set of relevant hypotheses.

2.4 Adaptation

In contrast to most traditional applications of Bayes nets, where the structure of the net is fixed, the system intended in this research includes adaptation mechanisms that allow a dynamic reconfiguration of the net according to the characteristics of the incoming information.

The adaptation mechanisms are based on the evaluation of the level of uncertainty present in the state estimation and the evaluation of the quality of the information provided by each agent in terms of uncertainty reduction. The design goals are to perform robust estimation keeping uncertainty low, and also to perform an efficient estimation avoiding the processing of irrelevant, misleading, or redundant information. In order to achieve these goals it is needed to introduce two performance metrics.

The first metric called uncertainty deviation (UD) is intended to evaluate the level of uncertainty in the state representation. The intuition behind this metric is to quantify the dispersion of the state representation with respect to the most probable hypothesis known as maximum a posteriori hypothesis (MAP). Equation (3) shows this metric; here d corresponds to a distance metric between hypotheses:

$$UD = \sum_{h_i}^{|h|} d^2(h_i, MAP) * P(h_i) \quad (3)$$

For the case of Euclidean distance and state variables (x, y, w, h) , (3) can expressed as :

$$UD = \sum_{h_i} \sum_{l=x,y,w,h} (h_i(l) - MAP(l))^2 * P(h_i) \quad (4)$$

$$UD = \sigma_x^2 + \sigma_y^2 + \sigma_w^2 + \sigma_h^2 \quad (5)$$

Where $h_i(l)$ represents the value of the hypothesis h_i in the l dimension, and σ_l^2 represents the variance in the l dimension calculated with respect to the MAP estimator. So the UD metric is equivalent to the trace of the covariance matrix or sum of its eigenvalues (Fukunaga, 1972).

The second metric is intended to evaluate the quality of the information provided by each agent in terms of uncertainty reduction. The intuition is if an agent is providing good information its local likelihood should be close to the state-pdf maintained by the abstraction node. So the problem reduces to quantify similarity between distributions. In this work we compare distributions using the Kullback-Leibler divergence (Cover, Thomas, 1991), which is given by (7).

$$D(f, g) = \sum_i f(i) * \log \frac{f(i)}{g(i)} \quad (6)$$

$f(i)$ = pdf for the state estimation

$g(i)$ = local normalized agent likelihood

Using the previous performance metrics we introduce two adaptation schemes to the state estimation. The first scheme is performed by the abstraction node. Using the UD metric the abstraction node measures the level of ambiguity in its state representation. If this level exceeds a predefined threshold the central inference sends an activation signal to any inactive agent to start sending supporting evidence that can eventually reduce the current ambiguities. Also, in case that this level is lower than a predefined threshold, meaning that the uncertainty is low, the abstraction node stops the less informative agent in order to increase the efficiency of the state estimation. The selection of the less informative agent is performed in base to the relative values of the Kullback-Leibler divergence among the active agents.

The second adaptation scheme is carry out locally by each agent using the UD metric. In this case, given that each agent calculates a likelihood function, the MAP is replaced in (3) by the maximum likelihood hypothesis (ML). Using this metric each agent evaluates the local level of uncertainty in its information sources. If this level exceeds a predefined threshold the agent modify its own local actions in order to improve its performance. In case that after a number of cycles the agent still cannot improve its performance, it stops processing information becoming an inactive agent.

Section 4 describes an implementation of these adaptation mechanisms for the case of visual algorithms based on color, depth, and shape visual cues.

3 Related Work

The idea of reducing uncertainty by combining knowledge from difference sources is by no account new. In several fields it is possible to find studies that recognize the relevance of integrating information in order to create more robust and flexible systems. Although all the abundant literature, there have been a gap between the conceptual idea and the production of working systems for real problems. Important issues such as the organization and control of the pieces of knowledge, and in special the development of mechanisms that allow the adaptation and feedback among the knowledge sources have not been tackled in depth, and they are still open questions.

In the AI domain the blackboard model for problem solving is one of the first attempts to adaptively integrate different types of knowledge sources (Reddy, Erman, Richard, 1993). The blackboard conceptualization is closely related to the ideas presented in this work, but as a problem-solving scheme the blackboard model offers just a conceptual framework for formulating solutions to problems. This research aims to extent the blackboard conceptualization to a computational specification or working system, providing specific mechanisms to perform probabilistic inference and adaptive integration for the case of dynamic visual information.

In the computer vision literature, there have been an interest in the creation of systems that integrate information from different visual cues, but we are not aware of working systems that include mechanisms to adaptively select the most appropriate set of visual cues, or that incorporate mutual feedback between the individual visual algorithms. Among the works that have shown the gain in robustness of combining several visual modules, it is possible to mention (Darrell et al., 1998; Krotkov, Balacsy, 1993). Unfortunately, most of these works have not considered in their systems topics such as adaptation and general types of uncertainties, being the works by Isard and Blake (Isard, Blake, 1996), and Rasmussen and Hager (Rasmussen, Hage, 1998) some of the notable exceptions.

4 Implementation

Bounding boxes were used to describe the state of a tracked target. These bounding boxes were modelled by their center position (x , y), width (w), and height (h). The rectangular boxes in the right side of figure 6 are examples of hypotheses used to perform state estimation. In all the examples shown in this work the state-pdf was approximated using 1000 samples. Also for the dynamic of the system we used a stationary Gaussian model with sigma equal to 20 pixels

4.1 Visual Agents

At this moment the implementation is based on tree visual agents: Color, Stereovision, and Shape agents. These agents are implemented using a cooperative distributed multi-agent architecture (Dolan, Trebi-Ollennu, Soto, Khosla, 1999). Each agent is an independent process that runs concurrently and can be started on as needed basis in a remote or local machine

Color Agent The color agent uses as observation the hue histogram of the pixels inside each hypothesis in the sample set. Hue is calculated according to the relative influence of the dominant RGB component with respect to the other 2 components. This is achieved using a circular transformation, where the RGB components are equally spaced on the circle. Red at 0° , Green at 120° , and Blue at 240° . In this way, the hue value for an input point with dominant color Green is given by:

$$Hue(R, G, B) = 120 + \frac{B - R}{\max(R, G, B) - \min(R, G, B)} \quad (7)$$

Shifting (7) by 120° it is possible to obtain the hue for the cases of red and blue dominance. We prefer this heuristic transformation to the traditional RGB-HSI transformation based on a color pyramid, because it produces a smoother mapping.

In order to express its observations in terms of a likelihood function, the color agent calculates the similarity between each hypothesis and a reference histogram. A detailed description of this process can be found in (Soto, Khosla, 2002).

A special agent called a detector selects the reference histogram automatically. The idea here is to divide the agents in two types: detectors and specialists. Detector agents are general tools able to initialize the tracking of a target. Detectors explore the incoming information looking for possible structures without relying in specific illumination, views, posture, etc. Once possible candidate structures have been identified, it is possible to use specialist agents. These agents are more specific tools able to look for specific supporting evidence in order to provide more efficient and robust appearance models. More details can be found in (Soto, Khosla, 2002).

Stereovision Agent The stereovision agent uses as observation the depth values of the pixels inside each hypothesis in the sample set. The algorithm is based on depth segmentation and blob analysis, a detailed description of the algorithm can be found in (Soto, Saptharishi, Dolan, Trebi-Ollennu, Khosla, 1999).

To express its observations in terms of a likelihood function, the stereovision agent estimates four properties of the depth values of each hypothesis: depth variance, heights of the depth blobs, blob shape as a ratio between the width and height of the blob, and number of points with valid depth values. Using these depth features the stereovision agent estimate a likelihood function using a multi-variate Gaussian pdf given by (9):

$$\text{Likelihood}(\vartheta) = \frac{\exp\{-0.5 * (\vartheta - \mu)^T * B^{-1} * (\vartheta - \mu)\}}{(2\pi)^{1/2} |B|^{1/2}} \quad (8)$$

ϑ = feature vector (depth, height, blob shape, valid points)

μ = feature vector for the reference target

B = diagonal covariance matrix estimated by previous examples

In the case of depth information there is also a detector agent. This agent initializes candidate targets using the results of the depth based segmentation algorithm.

Also the reference feature vector is updated using expressions similar to equations (10) and (11).

Shape Agent The shape agent is based on a novel algorithm that uses as observation the silhouette of the targets. A set of training examples is used to generate a shape model. The training examples consist of a set of shape vectors. Figure 3 shows the different steps involved in the calculation of each shape vector. The input to the process is a gray level image consisting of a bounding box containing a target, as shown in Fig. 3 (a). A canny edge detector is applied in this area obtaining an edge image, as shown in Fig. 3 (b). Starting from a configuration similar to the one presented in Fig. 3 (c), a snake curve is fit to the edge information, as shown in Fig. 3 (d). Using the resulting snake points a uniform B-spline curve (Bartels, Beatty, Barsky, 1987) is fit to the target boundary using N=40 control points. After scaling these control points to a uniform size, they are used to generate a continuous close contour around the silhouette of the target, as shown in Fig. 3 (e). Finally, this close contour is used to generate the shape vector, which consists of the signed distance function [18] calculated from a regular grid of points inside the input image. Figure 3 (f) shows a gray level image of the signed distance function obtained for the example, each point encodes the distance to the nearest point on the boundary with negative values inside the boundary.

The training examples are used to build a point distribution model using principal component analysis. After the PCA decomposition only the eigenvectors corresponding to the largest eigenvalues are considered to build the shape space. This corresponds to the eigenvalues that accounts for more than 95% of the shape variations. In this way, the resulting model consists of a mean shape and an orthonormal shape space built with the more significant eigenvectors.

To express new observations in terms of a likelihood function, each hypothesis is transformed into a shape vector using the process described in Fig. 3. After subtracting the mean shape, the resulting vector is projected to the shape space obtaining a vector S . Assuming a Gaussian distribution of shape the likelihood of the shape vector is calculated by:

$$P(S) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2} * (S^T \sum^{-1} S)\right) \quad (9)$$

Where \sum is a $k \times k$ diagonal matrix containing the more significant eigenvalues.

It is important to note that the use of a signed distance function to build the shape vectors allows a greater tolerance to slight misalignment with respect to the shape model. Although in this work the shape model was learned offline, it is possible to think in a case where the model is automatically learned and updated online using as training examples the tracking results of other visual agents.

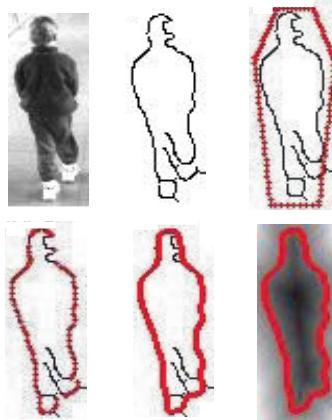


Fig. 3. a) Input image, b) Edge image after applying canny edge detector, c) Initial configuration to run snake algorithm, d) Resulting snake fitting the contour, e) Uniform B-Spline fitting the contour, f) Shape vector given by signed distance

5 Results

A version of the system proposed in this work has been developed for the case of person tracking. Figure 4 shows three frames of a video sequence used to evaluate the system (for more results check (Soto, Khosla, 2002)). In this sequence the system tracks 2 targets that move without overlap.



Fig. 4. Three frames of a video sequence used for target tracking

After an initial detection given by the stereovision agent detector, the system starts the tracking of each target using the color, stereovision, and shape agents. After 5 frames the system decides to track the left side target using just the simpler color agent. In the case of the right side target, its proximity to a large bright window and limitations in the dynamic range of the color CCD make the color information unreliable, and the system decides to track this target using the stereovision and shape agents. In this case, as it is shown in Fig. 5, the tracking using just stereovision is not totally reliable because the stereovision segmentation tends to link the target to the glass wall. In the same way, the tracking based just on shape tends to miss

parts of the silhouette when the image becomes too bright. As it is shown on Fig. 6, a combination of the two cues allows performing a proper tracking for this target.



Fig. 5. Result of stereovision segmentation. The close distance between the right side target and the glass wall confuses the stereovision agent

Figure 7 shows the evolution of the UD index for the case of the left target. The peak in the curve around frame 62 is due to the increase in the lighting coming from the left window, which produces an abrupt change in the color appearance mainly due to limitations in the dynamic range of the CCD camera. In this case the system activates the stereovision agent reducing the uncertainty. This shows how adaptation allows the system to operate successfully even when the assumptions of some of the algorithms, in this case color constancy, are not applicable at all times.

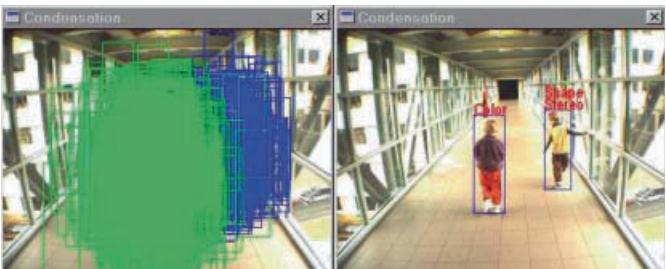


Fig. 6. (Left) hypotheses for state estimation at frame 7. (Right) MAP hypotheses frame 7

6 Conclusions

This paper presented a new approach for the creation of an adaptive visual system. Using an intelligent agent paradigm in combination with a probabilistic approach and uncertainty metrics, we developed a sound methodology to adaptively combine dynamic visual information.

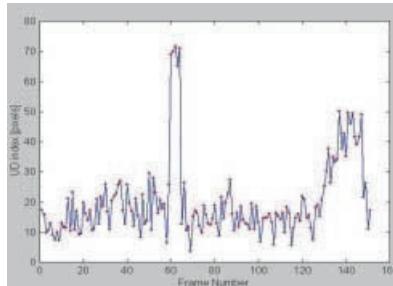


Fig. 7. Evolution of the UD index at the abstraction node for the left side target

Taking into account the level of uncertainty in the information provided by the agents and the uncertainty in the state estimation, the system was able to take appropriate decisions about the more suitable use of the incoming information.

The implementation of the system for the case of human tracking showed encouraging results. The comparative analysis with respect to the case of operation without adaptation and/or integration shows that the adaptive integration of information increases the robustness and efficiency of the system in terms of accuracy and output rate.

There are still further research avenues to improve the system. At this moment we are currently adding more visual cues. We are also adding alternatives algorithms for the case of color and depth cues that differ in terms of assumptions and complexity, so the system can select the most appropriate. In the case of target tracking we are adding reasoning schemes for the case of target occlusion.

We believe that a synergistic combination of elements from computer vision, intelligent agents technology, probabilistic reasoning, and information theory is a viable way for the creation of *flexible, robust and efficient* vision system for tasks like target tracking.

References

- Bartels R, Beatty J, Barsky B (1987) An introduction to splines for use in computer graphics and geometric modeling. Morgan Kaufmann.
- Cover T, Thomas J (1991) Elements of information theory. telecommunications. Wiley series.
- Darrell T, Gordon G, Harville M, Woodfill J (1998) Integrated person tracking using stereo, color, and pattern detection. In Proc. of the conf on computer vision and pattern recognition, pp 601–609.
- Dolan J, Trebi-Ollennu A, Soto A, Khosla P (1999) Distributed tactical surveillance with ATVs. In Proc. of SPIE, vol 3693, Orlando Fl.
- Fukunaga K (1972) Introduction to statistical pattern recognition. Academic Press.
- Gordon N, Salmon D, Smith A (1993) A novel approach to nonlinear/ non gaussian bayesian state estimation. In IEE Proc. on Radar and Signal Processing, vol 140, pp 107–113.
- Isard M, Blake A (1996) Visual tracking by stochastic propagation of conditional density. In Proc. of 4th European conf on computer vision, pp 343–356.

- Jennings N, Woolridge M (1995) Applying agent technology. *Journal of Applied Artificial Intelligence*. special issue on Intelligence Agents and Multi-Agent Systems.
- Kanade T, Collins R, Burt P, Wixson L (1998) Advances in cooperative multisensor video surveillance. In *Darpa Image Understanding Workshop*, pp 3–24.
- Kitagawa G (1996) Monte Carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5:1–25.
- Krotkov E, Bajcsy R (1993) Active vision for reliable ranging: cooperating focus, stereo, and vergence. *International Journal of computer vision*, 11(2):187–203.
- Nourbakhsh I, Bobenage J, Grange S, Lutz R, Meyer R, Soto A (1999) An affective mobile educator with a full-time job. *Artificial Intelligence*, pp 95–124.
- Rasmussen C, Hage G (1998) Joint probabilistic techniques for tracking multi-part objects. In Proc. of the conference on computer vision and pattern recognition, Santa Barbara.
- Reddy R, Erman D, Richard N (1993) A model and a system for machine recognition of speech. In *IEEE transactions on audio and electroacoustic theory*, vol AU-21.
- Soto A, Khosla P (2002) Adaptive agent based system for state estimation using dynamic multidimensional information sources. *Lecture notes in computer science*, inprint.
- Soto A, Saptharishi M, Dolan J, Trebi-Ollennu A, Khosla P (1999) CyberATVs: dynamic and distributed reconnaissance and surveillance using all terrain ugv's. In Proc. of Intl conf on field and service robotics.
- Zeki S (1993) *A vision of the brain*. Blackwell scientific publications, Oxford.