

Springer Tracts in Advanced Robotics

Volume 24

Editors: Bruno Siciliano · Oussama Khatib · Frans Groen

Springer Tracts in Advanced Robotics

Edited by B. Siciliano, O. Khatib, and F. Groen

Vol. 23: Andrade-Cetto, J.; Sanfeliu, A.
Environment Learning for Indoor Mobile Robots
130 p. 2006 [3-540-32795-9]

Vol. 22: Christensen, H.I. (Ed.)
European Robotics Symposium 2006
209 p. 2006 [3-540-32688-X]

Vol. 21: Ang Jr., H.; Khatib, O. (Eds.)
Experimental Robotics IX
618 p. 2006 [3-540-28816-3]

Vol. 20: Xu, Y.; Ou, Y.
Control of Single Wheel Robots
188 p. 2005 [3-540-28184-3]

Vol. 19: Lefebvre, T.; Bruyninckx, H.; De Schutter, J.
Nonlinear Kalman Filtering for Force-Controlled
Robot Tasks
280 p. 2005 [3-540-28023-5]

Vol. 18: Barbagli, F.; Prattichizzo, D.; Salisbury, K. (Eds.)
Multi-point Interaction with Real and Virtual Objects
281 p. 2005 [3-540-26036-6]

Vol. 17: Erdmann, M.; Hsu, D.; Overmars, M.;
van der Stappen, F.A (Eds.)
Algorithmic Foundations of Robotics VI
472 p. 2005 [3-540-25728-4]

Vol. 16: Cuesta, F.; Ollero, A.
Intelligent Mobile Robot Navigation
224 p. 2005 [3-540-23956-1]

Vol. 15: Dario, P.; Chatila R. (Eds.)
Robotics Research – The Eleventh International
Symposium
595 p. 2005 [3-540-23214-1]

Vol. 14: Prassler, E.; Lawitzky, G.; Stopp, A.;
Grunwald, G.; Hägele, M.; Dillmann, R.;
Iossifidis, I. (Eds.)
Advances in Human-Robot Interaction
414 p. 2005 [3-540-23211-7]

Vol. 13: Chung, W.
Nonholonomic Manipulators
115 p. 2004 [3-540-22108-5]

Vol. 12: Iagnemma K.; Dubowsky, S.
Mobile Robots in Rough Terrain –
Estimation, Motion Planning, and Control
with Application to Planetary Rovers
123 p. 2004 [3-540-21968-4]

Vol. 11: Kim, J.-H.; Kim, D.-H.; Kim, Y.-J.; Seow, K.-T.
Soccer Robotics
353 p. 2004 [3-540-21859-9]

Vol. 10: Siciliano, B.; De Luca, A.; Melchiorri, C.;
Casalino, G. (Eds.)
Advances in Control of Articulated and Mobile Robots
259 p. 2004 [3-540-20783-X]

Vol. 9: Yamane, K.
Simulating and Generating Motions of Human Figures
176 p. 2004 [3-540-20317-6]

Vol. 8: Baeten, J.; De Schutter, J.
Integrated Visual Servoing and Force Control
198 p. 2004 [3-540-40475-9]

Vol. 7: Boissonnat, J.-D.; Burdick, J.; Goldberg, K.;
Hutchinson, S. (Eds.)
Algorithmic Foundations of Robotics V
577 p. 2004 [3-540-40476-7]

Vol. 6: Jarvis, R.A.; Zelinsky, A. (Eds.)
Robotics Research – The Tenth International Symposium
580 p. 2003 [3-540-00550-1]

Vol. 5: Siciliano, B.; Dario, P. (Eds.)
Experimental Robotics VIII
685 p. 2003 [3-540-00305-3]

Vol. 4: Bicchi, A.; Christensen, H.I.;
Prattichizzo, D. (Eds.)
Control Problems in Robotics
296 p. 2003 [3-540-00251-0]

Vol. 3: Natale, C.
Interaction Control of Robot Manipulators –
Six-degrees-of-freedom Tasks
120 p. 2003 [3-540-00159-X]

Vol. 2: Antonelli, G.
Underwater Robots – Motion and Force Control of
Vehicle-Manipulator Systems
209 p. 2003 [3-540-00054-2]

Vol. 1: Caccavale, F.; Villani, L. (Eds.)
Fault Diagnosis and Fault Tolerance for Mechatronic
Systems – Recent Advances
191 p. 2002 [3-540-44159-3]

S. Yuta · H. Asama · S. Thrun · E. Prassler · T. Tsubouchi (Eds.)

Field and Service Robotics

Recent Advances in Research and Applications

With 393 Figures

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

Editors

Prof. Shin'ichi Yuta
University of Tsukuba
Intelligent Robot Laboratory
Tennoudai 1-1-1
305-8573 Tsukuba, Japan

Prof. Hajima Asama
The University of Tokyo
Research into Artifacts
Center for Engineering (RACE)
Kashiwanoha 5-1-5, Kashiwa-shi
Chiba 277-8568, Japan

Dr. Erwin Prassler
Universität Ulm
FAW Forschungsinstitut für anwendungs-
orientierte Wissensverarbeitung
Helmholtzstr. 16
89081 Ulm, Germany

Prof. Takashi Tsubouchi
University of Tsukuba
Department of Information, Interaction
Technologies
Tennoudai 1-1-1
305-8573 Tsukuba, Ibaraki, Japan

Dr. Sebastian Thrun
Stanford University
Department of Computer Science
94305-9045 Stanford, USA

ISSN print edition: 1610-7438

ISSN electronic edition: 1610-742X

ISBN-10 3-540-32801-7 **Springer Berlin Heidelberg New York**

ISBN-13 978-3-540-32801-8 **Springer Berlin Heidelberg New York**

Library of Congress Control Number: 2006923559

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. Violations are liable to prosecution under German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2006
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 editors.

Data-conversion and production: PTP-Berlin Protago- \LaTeX -Production GmbH, Germany (www.ptp-berlin.com)

Cover-Design: design & production GmbH, Heidelberg

Printed on acid-free paper 89/3141/Yu - 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, Stanford University, USA

ASIA/OCEANIA

Peter Corke, CSIRO, Australia

Makoto Kaneko, Hiroshima University, Japan

Sukhan Lee, Sungkyunkwan University, Korea

Yangsheng Xu, Chinese University of Hong Kong, PRC

Shin'ichi Yuta, Tsukuba University, Japan

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



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 the new series of *Springer Tracts in Advanced Robotics (STAR)* 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 wider dissemination of research developments will stimulate more exchanges and collaborations among the research community and contribute to further advancement of this rapidly growing field.

Since its inception in 1996, FSR, the *International Conference on Field and Service Robotics* has published archival volumes of high reference value. With the launching of STAR, a more suitable home is found for this and other thematic symposia devoted to excellence in robotics research.

The Fourth edition of *Field and Service Robotics* edited by Shin'ichi Yuta, Hajime Asama, Sebastian Thrun, Erwin Prassler and Takashi Tsubouchi offers in its twelve-chapter volume a collection of a broad range of topics in advanced robotics. The contents of these contributions represent a cross-section of the current state of robotics research from one particular aspect: field and service applications, and how they reflect on the theoretical basis of subsequent developments. Pursuing technologies aimed at realizing skilful, smart, reliable, robust field and service robots is the big challenge running throughout this focused collection.

Rich by topics and authoritative contributors, FSR culminates with this unique reference on the current developments and new directions in field and service robotics. A fine addition to the series!

Naples, Italy
December 2005

Bruno Siciliano
STAR Editor

Preface

In the beginning of 1980's, the robot technology had started to be popular for the automation in many factories. Since then, the role of the robotics has spread wider and wider. Now, robots are expected to take over many human tasks, and to work in various environments. And, robots are also expected to perform in the circumstances, where human cannot stay, such as, deep sea, space or the extremely hazardous places. The robot that performs such tasks is called "field robot". These abilities have brought new activities to human. Beside such an indirect assist to human, the robots are also requested to help people more directly. This concept is called "service robot". Various service tasks for daily life have been investigated and many prototypes have already been implemented.

The advanced technology will be the base of people's daily life and society in 21 century, and we believe that the robotics will take an important and dominant part. However, we know that huge amounts of technical advancement are still necessary to realize the reliable and useful robots, which work in various real environments or support our daily life. It is still the big challenge to realize skillful, smart, reliable, and robust field and service robots, and we have to pursue this technology.

The series of the International Conference on Field and Service Robotics started 1996 in Australia and have been held biannually in different continents, to stimulate and make progress of the research on this important subject.

The conferences are steered by FSR Permanent Organizing Committee (POC), which have the members of:

Hajime Asama, John Bares, Raja Chatila, Peter Corke, Aarne Halme, John Hollerbach, Oussama Khatib, Christian Laugier, John Leonard, Eduardo Nebot, Roland Siegwart, Chuck Thorpe, Kazuya Yoshida, Shin'ichi Yuta, and Alex Zelinsky.

The fourth FSR conference was held in Lake Yamanaka where is on the foot of Mt. Fuji in Japan, on 14-16 July 2003.

In this conference, 49 original papers were selected and presented in single-track way, among 68 originally submitted papers, besides 4 invited/special talks. These presented papers and invited talks have demonstrated the recent advances in research and applications of field and service robotics in the world. More than 70 participants had actively discussed on all presentations and exchanged their opinion on both this particular theme and the direction of the robotics research, at: sessions, coffee breaks, mealtimes and on the occasion of excursion, banquet or in Japanese hot-spring public bath.

For the scientific program of this conference, the program committee requested to authors to provide information of their research and their opinion such as:

Exact targeted service and field,
Realized basis (Theoretical/Experimental/Applied),
Complexity of the working environment and achieving task, and
Estimated years until real use,

for selecting the proper papers, when they submitted their papers.

This book is the collection of the papers, which are presented at the conference and revised after these presentation and discussions. We believe that they are informative and useful to see the state of the art in this important and interesting subject.

At last, we are grateful to all participants for their contribution to have made this conference meaningful. Also, for their great help, we thank program committee members, which include POC members and,

Erwin Prassler, Paolo Fiorini, Gisbert Lawitzky, Bruno Siciliano, Gerd Hirzinger, Ben Kroese, Carlos Balaguer, and Friedrich Wahl, from Europe, Dieter Fox, Frank Dellaert, Matthew Deans, Roberto Manduchi, Howie Choset, Alonzo Kelly, Gurav Sukhatme, Daniela Rus, Christoph Mertz and Vijay Kumar, from USA, and

Makoto Mizukawa, Kazuhiro Kosuge, Tamio Arai, Satoshi Tadokoro, Shigeo Hirose, Toshio Fukuda, Yasushi Nakauchi, Yoshiki Shimomura, Koichi Osuka, Fumitoshi Matsuno, Takashi Tsubouchi, Alex Zelinsky, Kiyoshi Komoriya, and Seung-ho Kim, from Asia.

Finally we sincerely appreciate Dr. Kuniaki Kawabata, Dr. Shigeru Sarata and Dr. Hironori Adachi for their great contribution to operate the conference and edit the Proceedings.

May 2005

*Shin'ichi Yuta
Hajime Asama
Sebastian Thrun
Erwin Prassler
Takashi Tsubouchi*

Contents

Part 1 – Invited Papers

Service RT Systems	3
<i>Kazuhiro Kosuge</i>	
A Small Biped Entertainment Robot Creating Attractive Applications	13
<i>Yoshihiro Kuroki</i>	
Mobile Robots Facing the Real World	21
<i>Roland Siegwart</i>	
Breakthroughs in Human Technology Interaction	31
<i>Bernd Reuse</i>	

Part 2 – Indoor Navigation

Indoor Navigation for Mobile Robot by Using Environment-Embedded Local Information Management Device and Optical Pointer	41
<i>Tsuyoshi Suzuki, Taiki Uehara, Kuniaki Kawabata, Daisuke Kurabayashi, Igor E. Paromtchik, and Hajime Asama</i>	
Wall Following with Constrained Active Contours	51
<i>Elliot S. Duff and Jonathan M. Robert</i>	
Landmark-Based Nonholonomic Visual Homing	61
<i>Kane Usher, Peter Corke, and Peter Ridley</i>	
Recursive Probabilistic Velocity Obstacles for Reflective Navigation	71
<i>Boris Kluge and Erwin Prassler</i>	

Part 3 – Rough Terrain Navigation

Learning Predictions of the Load-Bearing Surface for Autonomous Rough-Terrain Navigation in Vegetation	83
<i>Carl Wellington and Anthony Stentz</i>	
A Terrain-Aided Tracking Algorithm for Marine Systems	93
<i>Stefan Williams and Ian Mahon</i>	

Experimental Results in Using Aerial LADAR Data for Mobile Robot Navigation 103

Nicolas Vandapel, Raghavendra Donamukkala, and Martial Hebert

Autonomous Detection of Untraversability of the Path on Rough Terrain for the Remote Controlled Mobile Robots 113

Kazuma Hashimoto and Shin'ichi Yuta

Part 4 – Localization

Mobile Robot Navigation Based on DGPS and Odometry in Campus Environment 125

Kazunori Ohno, Takashi Tsubouchi, Bunji Shigematsu, Shoichi Maeyama, and Shin'ichi Yuta

Vehicle Localization Using Inertial Sensors and GPS 135

Libor Přeučil and Roman Mázl

An Experimental Study of Localization Using Wireless Ethernet 145

Andrew Howard, Sajid Siddiqi, and Gaurav S. Sukhatme

MALOC - Medial Axis LOCalization in Unstructured Dynamic Environments 155

Michael Fiegert and Charles-Marie De Graeve

Part 5 – Mapping and Tracking

Market-Based Multirobot Coordination Using Task Abstraction 167

Robert Zlot and Anthony Stentz

Decentralised SLAM with Low-Bandwidth Communication for Teams of Vehicles 179

Eric Nettleton, Sebastian Thrun, Hugh Durrant-Whyte, and Salah Sukkarieh

Case Studies of a Borehole Deployable Robot for Limestone Mine Profiling and Mapping 189

Aaron Morris, Derek Kurth, Daniel Huber, William Whittaker, and Scott Thayer

Bayesian Programming for Multi-target Tracking: An Automotive Application 199

Christophe Coué, Cédric Pradalier, and Christian Laugier

Optimal Search of a Lost Target in a Bayesian World 209

Frédéric Bourgault, Tomonari Furukawa, and Hugh F. Durrant-Whyte

Part 6 – Rough Terrain Locomotion**Terramechanics-Based Analysis and Traction Control of a Lunar/Planetary Rover 225***Kazuya Yoshida, Toshinobu Watanabe, Noriyuki Mizuno, and Genya Ishigami***Topological Analysis of Robotic N-Wheeled Ground Vehicles 235***Michel Lauria, Steven Shooter, and Roland Siegwart***Development of a Control System of an Omni-directional Vehicle with a Step-Climbing Ability 245***Daisuke Chugo, Kuniaki Kawabata, Hayato Kaetsu, Hajime Asama, and Taketoshi Mishima***Sensor-Based Walking on Rough Terrain for Legged Robots 255***Yasushi Mae, Tatsuhi Mure, Kenji Inoue, Tatsuo Arai, and Noriho Koyachi***Part 7 – Helicopters and Air Vehicles****Experiments in Learning Helicopter Control from a Pilot 267***Gregg Buskey, Jonathan Roberts, and Gordon Wyeth***Landing on a Moving Target Using an Autonomous Helicopter 277***Srikanth Saripalli and Gaurav S. Sukhatme***Scan Alignment and 3-D Surface Modeling with a Helicopter Platform 287***Sebastian Thrun, Mark Diel, and Dirk Hähnel***Real-time Navigation, Guidance, and Control of a UAV Using Low-Cost Sensors 299***Jong-Hyuk Kim, Salah Sukkarieh, and Stuart Wishart***A Compact Millimeter Wave Radar Sensor for Unmanned Air Vehicles 311***Ali Haydar Göktoğan, Graham Brooker, and Salah Sukkarieh***Part 8 – Mobility and Manipulation****Motion Analysis of a Parallel Mobile Robot 323***Shraga Shoval and Moshe Shoham***Teleoperation System for Two Tracked Mobile Robots Transporting a Single Object in Coordination Based on Function Allocation Concept 333***Hiroki Takeda, Zhi-Dong Wang, and Kazuhiro Kosuge*

Development of a Terrain Adaptive Stability Prediction for Mass Articulating Mobile Robots	343
<i>Antonio Diaz-Calderon and Alonzo Kelly</i>	

ROBHAZ-DT2: Passive Double-Track Mobile Manipulator for Explosive Ordnance Disposal	355
<i>Sungchul Kang, Changhyun Cho, Changwoo Park, Jonghwa Lee, Dongseok Ryu, and Munsang Kim</i>	

Part 9 – Human-Robot Interaction

Towards Safer Roads by Integration of Road Scene Monitoring and Vehicle Control	367
<i>Lars Petersson and Alexander Zelinsky</i>	

Performing Skilled Work with an Interactively Operated Service Robot	377
<i>Aarne Halme, Jouni Sievilä, Ilkka Kauppi, and Sami Ylönen</i>	

A Multi-purpose Eight-Legged Robot Developed for an Evaluation of a Neural Interface	385
<i>Takashi K. Saito, Itsuro Saito, Nobuyuki Nemoto, Koki Takiura, Toshinaga Ozeki, Naoto Kakuta, Takahiro Tohyama, Takashi Isoyama, and Tsuneo Chinzei</i>	

Online Interactive Building of Presence	395
<i>Jussi Suomela, Jari Saarinen, Aarne Halme, and Panu Harmo</i>	

Part 10 – Health Care and Service Tasks

Light Weight Autonomous Climbing Robot for Elderly and Disabled Persons' Services	407
<i>Carlos Balaguer, Aantonio Giménez, Alberto Jardón, Raúl Correal, Ramiro Cabas, and Pavel Staroverov</i>	

Planning under Uncertainty for Reliable Health Care Robotics	417
<i>Nicholas Roy, Geoffrey Gordon, and Sebastian Thrun</i>	

Development of a Personal Service Robot with User-Friendly Interfaces	427
<i>Jun Miura, Yoshiaki Shirai, Nobutaka Shimada, Yasushi Makihara, Masao Takizawa, and Yoshio Yano</i>	

An Enhanced Robotic Library System for an Off-Site Shelving Facility	437
<i>Jackrit Suthakorn, Sangyoon Lee, Yu Zhou, Sayeed Choudhury, and Gregory S. Chirikjian</i>	

International Contest for Cleaning Robots: Fun Event or a First Step Towards Benchmarking Service Robots	447
<i>Erwin Prassler, Martin Hägele, and Roland Siegart</i>	

Part 11 – Mining

Dragline Automation: Experimental Evaluation Through Productivity Trial	459
<i>Peter Corke, Graeme Winstanley, Matthew Dunbabin, and Jonathan Roberts</i>	

Shearer Guidance: A Major Advance in Longwall Mining	469
<i>David C. Reid, David W. Hainsworth, Jonathon C. Ralston, and Ronald. J. McPhee</i>	

Development of an Autonomous Conveyor-Bolting Machine for the Underground Coal Mining Industry	477
<i>Jonathon C. Ralston, Chad O. Hargrave, and David W. Hainsworth</i>	

A Case Study in Robotic Mapping of Abandoned Mines	487
<i>Christopher Baker, Zachary Omohundro, Scott Thayer, William Whittaker, Mike Montemerlo, and Sebastien Thrun</i>	

Automatic 3D Underground Mine Mapping	497
<i>Daniel F. Huber and Nicolas Vandapel</i>	

Part 12 – Rescue and Agricultural Applications

Development of Pneumatically Controlled Expandable Arm for Search in the Environment with Tight Access	509
<i>Daisuke Mishima, Takeshi Aoki, and Shigeo Hirose</i>	

Development of Mobile Robots for Search and Rescue Operation Systems	519
<i>Akihiro Ikeuchi, Toshi Takamori, Shigeru Kobayashi, Masayuki Takashima, Shiro Takashima, and Masatoshi Yamada</i>	

Distributed Search and Rescue with Robot and Sensor Teams	529
<i>George Kantor, Sanjiv Singh, Ronald Peterson, Daniela Rus, Aveek Das, Vijay Kumar, Guilherme Pereira, and John Spletzer</i>	

Spraying Robot for Grape Production	539
<i>Yuichi Ogawa, Naoshi Kondo, Mitsuji Monta, and Sakae Shibusawa</i>	

Path Planning for Complete Coverage with Agricultural Machines	549
<i>Michel Taïèx, Philippe Souères, Helene Frayssinet, and Lionel Cordesses</i>	
Author Index	559

Service RT Systems

Kazuhiro Kosuge

Department of Bioengineering and Robotics
Tohoku University
Sendai 980-8579, Japan
kosuge@irs.mech.tohoku.ac.jp
<http://www.irs.mech.tohoku.ac.jp>

Abstract. In this article, we first introduce a new research model proposed by the Science Council of Japan in 1999[1][2]. The model was proposed based on how research is carried out and how the research is integrated with our society as a culture. We consider what the Field and Service Robotics is, and introduce the Service RT Systems as a type of the field and service robots. Several examples illustrate the relation between the RT Systems and the Robotics Research.

1 Introduction

More than thirty years have passed since robots have been utilized in industries as industrial robots. The Robot Technology, which has been developed through the development of the industrial robots, is applicable to many fields outside of factories. The “Robot Technology” is abbreviated as “RT” in this article[3][4][5]. The RT includes technologies relating to physical interactions between the systems and their environments, while IT(Information Technology) mainly deals with the information.

In this article, we first introduce a research model, which explains how research is carried out and how the research is integrated with our society as a culture. The research model was proposed by the Third Committee of the 17th Science Council of Japan in 1999. The robotic research seems to be explained by the model very well. The RT systems are intelligent systems, which have interactions with the real world and which the RT is embedded in. The Service RT Systems is one of the outcomes of the robotic research in the real world.

We have been developing robot helpers and robot technologies necessary for the realization of the robot helpers in future. We introduce some of the robot helpers and robot technologies developed for them, then, discuss how the robot technologies have been applied to real world problems as the Service RT Systems, which includes an application in a construction site and several assist RT systems. Finally, we discuss a future direction of the Service RT Systems necessary for the aging society which we are facing now.

2 New Research Model and Field and Service Robotics

As mentioned above, a new research paradigm has been proposed by The Third Committee, The 17th Science Council of Japan, in the report[1], entitled as “Toward a new research paradigm” in April, 1999. The chair of the committee was Professor Shun-ichi Iwasaki who proposed the perpendicular magnetic recording is 1975. The research model is said to have been developed based on his experience as a researcher when he invented the perpendicular magnetic recording. The concept is also included in his own article[2]. In the report, the research has been classified into three phases based on the mental process of researchers in performing actual research as shown in fig. 1.

1. Creation Model Research(First Model Research)
Proposal of hypothesis and verification,
which is characterized by the words, “Original”, “Unconventional”, “Recognize/Discover”, and “Noncompetitive.”
2. Development Model Research(Second Model Research)
Standardization and Popularization of the Creation Model Research,
which is characterized by the words, “Precise”, “Objective”, “Design/Make” and “Competitive.”
3. Integration Model Research(Third Model Research)
Integration with the real world,
which is characterized by the words, “Social”, “Humanity”, “Ethical” and “Co-operative.”

In the report, it is also noted that practical research might be the Creation Model Research and scientific research might be the Development Model Research. It depends on the nature of the research which model the research belongs to.

Most of the research in robotics seems to be categorized as the Creation Model Research and the Development Model Research. Some of researches in robotics,

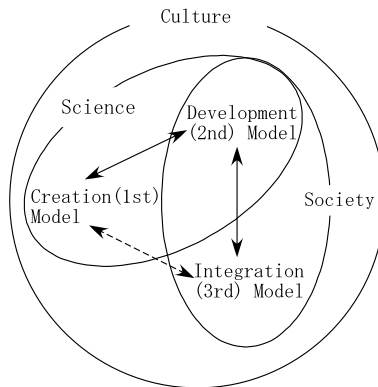


Fig. 1. Model research proposed in [1] and [2]

such as industrial robots, robots assisting surgery, intelligent transport systems, and so on, have reached to the Integration Model Research and some of them have integrated with the real world.

The Field and Service Robotics is the robotics which covers from the Creation Model Research through the Integration Model Research. The new model research is useful for the researchers in robotics to position what they are trying to do and identify what they would like to do.

3 Robotics and RT Systems

In general, the word “Robotics” means both Robot Science and Robot Technology. When we do research in robotics, we may have some typical image of robots which the research results will be integrated with. In most cases, however, when the robotics is integrated with a real system, the system does not always look like a robot. The robotics is integrated with a real system as the robot technology.

The word “RT systems” defines what the system is. As mentioned in the previous section, the RT systems are intelligent systems, which have interactions with the real world and which the RT is embedded in. Looking around ourselves, we have a lot of RT systems as the integration of the Robot Technology with a real world system. The followings are some of examples of RT systems;

- Power Steering System of Automobile

The power steering system is a device of a vehicle that facilitates the turning of the steering wheel by the driver. An electric power steering system has been developed recently and used in most of recent automobiles for less fuel consumption. Several types of the electric power steering system exist, and some of them use an electric motor to generate steering assist torque based on the driver’s steering torque and the vehicle’s state. A simple force control system is used in the systems.

- Driving Support System

Some of recent automobiles have a lane keeping support system. Mobile robot technologies and robot vision technologies are used in the system together with the electric power steering system, although the automobile does not look like a robot.

- Power Assist Bicycle

The power assist bicycle has been initially proposed and commercialized by YAMAHA MOTOR CO., LTD. Similar types of bicycles are sold in Japan by many companies. An electric actuator assists the pedaling force according to its rider’s pedaling force which is measured by a torque sensor. A simple force control scheme is used in this system as in the case of the electric power steering system.

- Robot Cleaners

Recently, robot cleaners, such as the Trilobite and the Roomba, have been commercialized by the Electrolux and the iRobot respectively. Other companies are also planning to commercialize similar systems. The robot cleaners have

been developed based on the mobile robot technologies together with cleaning systems optimized for them.

- etc.

There are many systems which the robot technologies are embedded in. Some of them will be also introduced in the following sections.

4 From Robotics to RT Systems

We have proposed several types of robot systems doing tasks in cooperation with a human/humans[7]-[10]. The robot-human collaboration system was one of the first robots carrying an object in cooperation with a human[8] (fig. 2). In this system, the dual manipulators were controlled so that the apparent impedance of the manipulated object is specified. The operator maneuvers the object by applying its intentional force to the object with specified impedance. We have extended this concept to mobile robots. The Mobile Robot Helper, referred to as MR Helper[9], is a mobile manipulator system which consists of an omni-directional mobile base and dual manipulators(see fig. 3). A stereo camera system and other sensors have been used to control the system in addition to the object impedance control system which we have proposed for the robot-human collaboration system.

The Distributed Robot Helpers referred to as DR Helpers[10] consists of omni-directional mobile robots(fig. 4). By utilizing multiple small robots in coordination, handling of a heavy object has been realized. Several problems relating to coordination of multiple mobile robots have been solved through this research, such as how to eliminate the effect of motion errors among robots, and so on. MR Helper and DR Helpers are the platforms for research of robot helpers in human environments.

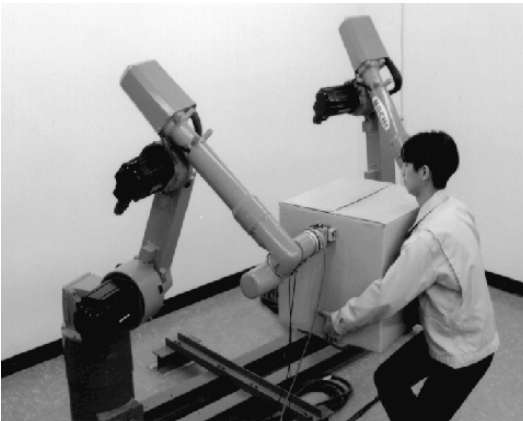


Fig. 2. Human robot coordination[8]

Through experiments using real robots, we have found and solved problems relating to real applications.

Recently, Matsushita Electric Works Ltd. has commercialized an electrically power assisted food delivery system for hospital use, referred to as “Delicart”[14]. The system has omni-directional mobile base and moves according to human intentional force applied to the system. The system is a typical RT system in which robot technologies have been implemented, although they have developed the system independently.



Fig. 3. Mobile robot helper[9]

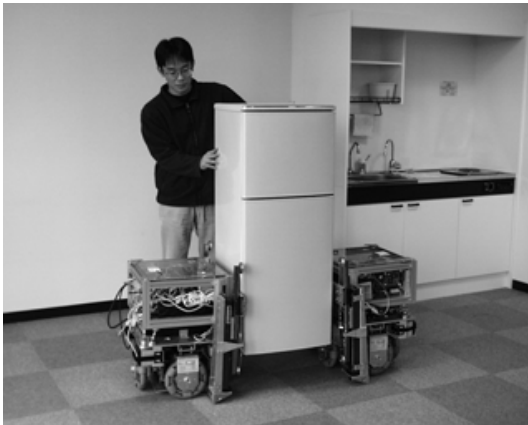


Fig. 4. Distributed robot helpers[10]

Yamada et al. have proposed a power assist system for an automobile assembly system, which is referred to as “Skill Assist”[15]. The concept of the system has been proposed by the Hardyman project in 1960’s by the Army and Navy in U.S.A. The concept has been explored by several researchers[6][7], etc. Yamada et al. have succeeded the application of the concept to a real production system by adding a system for the worker’s safety. It is reported that many systems have been already used in production lines worldwide, although the real system is not open to the public. The real system does not look like a robot, but is a real RT system.

Consider another example of the RT systems, which we have developed based on the result of the robot helper project. We have proposed MR Helper and DR Helpers for handling an object in cooperation with an operator based on the intentional force applied by the operator to the object. The weight of the manipulated object is fully supported by or shared with the robot helper/helpers. To handle the object, the operator is required to apply its intentional force to the object when the object is fully supported and both its intentional force and a part of the weight when the weight of the object is shared with the robot/robots. We apply this concept to the power assist bicycle which we introduced in the previous section.

In case of the existing power assist bicycle, the actuator is controlled so that the pedaling force is doubled. When climbing up and down a slope, the system might not work appropriately especially for the elderly. Anyone has to support a half of the pedaling force required to ride the bicycle even if the slope is very steep. The actuator is controlled to double the pedaling force even when climbing down a slope, which accelerates the bicycle

We have proposed a new type of power assist system for a bicycle with which the drag force is cancelled[11]. The rider of the new bicycle could ride it as if he rides the bicycle on a horizontal surface even on a slope. Fig. 5 shows both types of the bicycles when the rider was climbing down the slope. The left hand side picture of each frame shows the result with the proposed power assist system and the right hand side shows the result with conventional power assist system.

The bicycle with the proposed power assist system could be accelerated only when the rider would like to do so even when climbing down the slope. When climbing up the slope, anyone could ride the bicycle as if he/she rides it on a horizontal road. With the new power assist system, even elderly could ride the bicycle in any slope. We, however, could not sell the system at this moment in Japan because of a regulation relating to the power assist bicycle. This is another factor which we have to consider when we try to integrate a new system with our society.

5 From RT Systems to Robotics

In this section, we consider how the development of a RT system is fed back to the research of the Robotics based on our experience. We have developed an assembly system of tunnel wall segments for a shield tunnel excavation system[13] in cooperation with a company in Japan which is shown in fig.6. The system has been developed assuming that a compliant motion control scheme is available.

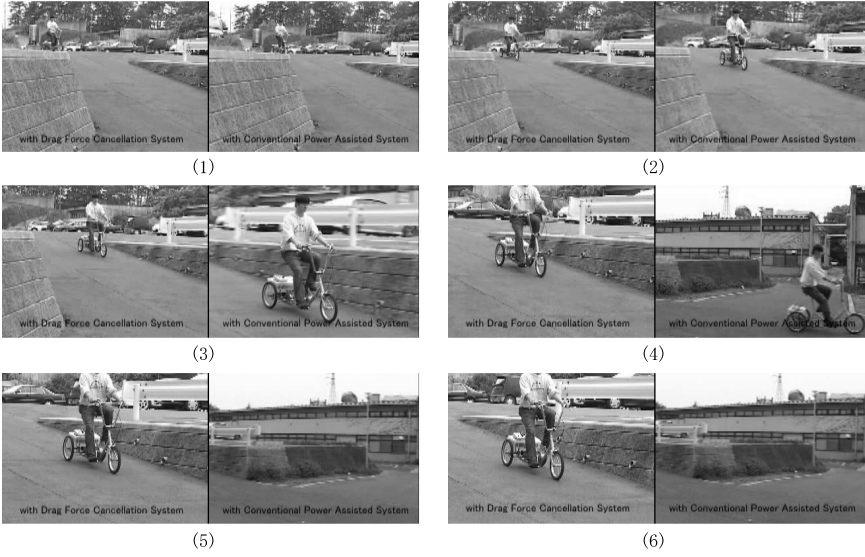


Fig. 5. Proposed power assist bicycle [11]

We first defined a generalized coordinate system for the compliant motion control system so that no jamming occurs during the assembly of the honeycomb shaped segments. We carefully and intuitively defined the coordinate system through a lot of case studies. At that time, we did not know how to select the coordinate system theoretically. The result was very interesting in that there is no center of rotation for the assembled segment. Three axes of rotations, which are used to control the orientation of the assembled segment, do not cross at one point.

With the developed segment assembly system, we could assemble the tunnel segments without measuring position of each segment precisely before assembling the segment to the existing ones. The assembled segment is fit to the existing ones without explicitly controlling the position and orientation. The interface force between the assembled segment and the existing ones guide the assembled segment into the right position and orientation as shown in fig. 6

To generalize the result which we have got from the development, we have started research of how to design a generalized coordinate system for parts-mating. In some sense, the selection of a coordinate system introduces a kind of a structure in the compliant motion of the assembled object with respect to the interface force/moment applied to the object from its environment.

The planar case has been completely solved[12]. Fig. 7 shows how a part is assembled by pushing it to its environment which the part is assembled to. The motion of the compliantly supported assemble part is generated based on the interface force between the part and the environment so that the part is lead to its right position and orientation assuming a certain bounded initial pose error and friction between

them. Please note that, in the planar case, the selection of the coordinate system is equivalent to the selection of the center of rotation.

This example shows how the development of the RT system is fed back to the robotics research. Through the development of real world RT systems, we could identify what problems we need to solve in the Robotics.

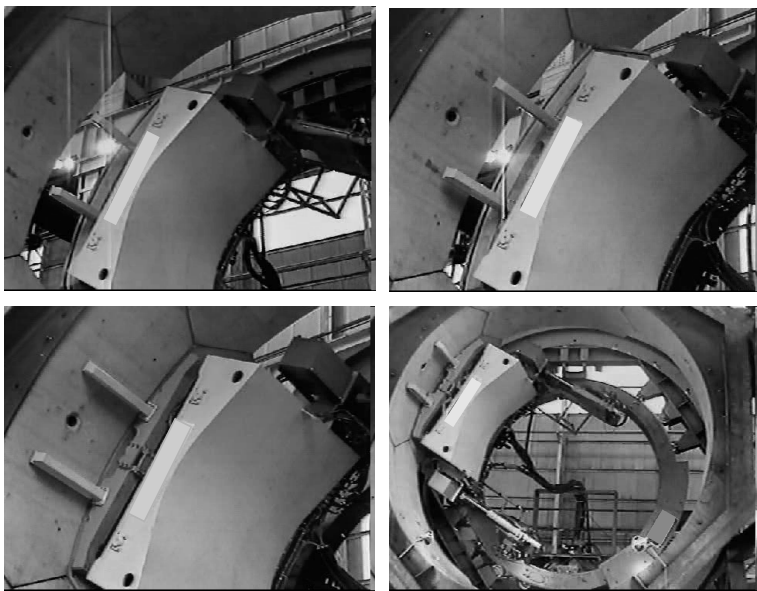


Fig. 6. Prototype of assembly system of tunnel Segments[13]

6 Conclusions

In this article, we first reviewed the New Research Model proposed by the 17th Science Council of Japan in 1999. According to the model, the research consists of three phases;

1. Creation Model Research(First Model Research)
2. Development Model Research(Second Model Research)
3. Integration Model Research(Third Model Research)

The Field and Service Robotics covers from the 1st model through the 3rd model. When we consider the Integration Model Research, a robot may not always look like a robot which everybody imagines. The RT system is defined to explain such a system.

We then consider several examples of the RT systems in the real world. We also introduced examples of how the robot technology is utilized in a real system

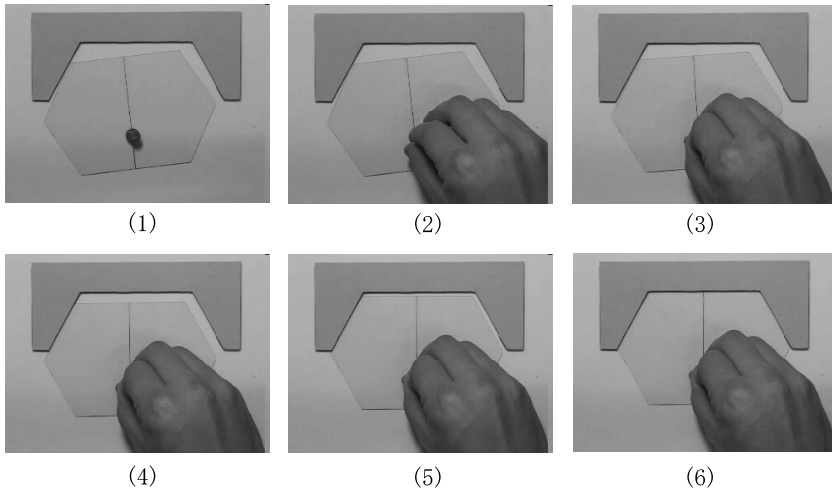


Fig. 7. Assembly system of planar parts based on parts-mating theory [12]

and how the research result of the RT system could be fed back to the research of Robotics. The Field and Service Robotics will be developed through the interactions among the research models. The interactions among the research models seems very important for developing the Service RT Systems for the aging society which we are facing now.

References

1. The Third Committee, The 17th Science Council of Japan, "Towards a new research paradigm," National Council of Japan, April, 1999 (in Japanese).
2. Shun-ichi Iwasaki, "A New Direction of Science in Japan - from Strategic Research to Model Research," Seminar of the Academy of Finland, February 5, pp.12-19, 1999.
3. Japan Robot Association, "Strategy for Creation of Society with Robots in 21st," Japan Robot Association, May, 2001 (in Japanese).
4. Japan Robot Association, "Open System Architecture for Developing Robots in New Fields," Japan Robot Association, March, 2003 (in Japanese).
5. Japan Robot Association, "RT Open Architecture and its Strategy," Japan Robot Association, March, 2003 (in Japanese).
6. H. Kazerooni, "Human-Robot Interaction via the Transfer of Power and Information Signals," *IEEE Transactions on System, Man and Cybernetics*, Vol.20, No.2, pp. 450-463, 1990.
7. K.Kosuge, Y.Fujisawa and T.Fukuda, "Control of Mechanical System with Man-machine Interaction," *Proceesings of 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 87-92, 1992.
8. K.Kosuge, H.Yoshida, D. Taguchi and T.Fukuda, "Robot-Human Collaboration for New Robotic Applications," *Proceesings of The IECON'94, 20th International Conference on Industrial Electronics Control and Instrumentation*, pp. 713-718, 1994.

9. Kazuhiro Kosuge, Manabu Sato and Norihide Kazamura, "Mobile Robot Helper, " *Proceesings of 2000 IEEE International Conference on Robotics and Automation*, pp. 583–588, 2000.
10. Yasuhisa Hirata and Kazuhiro Kosuge, "Distributed Robot Helpers Handling a Single Object in Cooperation with a Human, " *Proceesings of 2000 IEEE International Conference on Robotics and Automation*, pp. 458–463, 2000.
11. Hidenori Yabushita, Yasuhisa Hirata, Kazuhiro Kosuge, and Zhi-Dong Wang, "Environment-Adaptive Control Algorithm of Power Assisted Cycle, " *Proceeding of IECON2003* , pp. 1962–1968, 2003.
12. Kazuhiro Kosuge and Masayuki Shimizu, "Planer Parts-mating Using Structured Compliance, " *Proceesings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1477–1482, 2001.
13. Kazuhiro Kosuge, Koji Takeo, Daiji Taguchi, Toshio Fukuda and Hiroki Murakami "Task-Oriented Force Control of Parallel Link Robot for the Assembly of a Shield Tunnel Excavation System, " *IEEE/ASME Transactions on mechatronics*, Vol.1., No., pp. 250–258, 1996.
14. http://www.mew.co.jp/epm/pmd/delica/food/delicart_01.html
15. Yoji Yamada, Hitoshi Konosu, Tetsuya Morizoni and Yoji Umetani, "Proposal of Skill-Assist: A System of Assisting Human Workers by Reflecting Their Skills in Positioning Tasks, " *Proceesings of 1999 IEEE International Conference on Systems, Man and Cybernetics* , pp. IV–11–IV–16, 1999.

A Small Biped Entertainment Robot Creating Attractive Applications

Yoshihiro Kuroki

Technology Development Department 2
Entertainment Robot Company
Sony Corporation
Shinbashi Sumitomo Building
5-11-3 Shinbashi, Minato-ku,
Tokyo, 105-0004
Japan

Abstract. After the debut of SDR-3X, SDR-4X made a stage appearance in 2002. It remains a small humanoid type robot and is expanding its capabilities of adaptability in home environment. Some of the new key technologies have been developed for SDR-4X. One is a Real-time Integrated Adaptive Motion Control. Another technology is a motion creating software system called “SDR Motion Creator” which allows to create and develop SDR’s attractive motion performances. In addition, speech synthesis and singing voice production are also developed for enhancement of entertainment applications. A cappella chorus performance and high-tempo dance performance are introduced as the attractive applications.

1 Introduction

Some leading studies concerning Whole Body Cooperative Motion Control for humanoid robots have been proposed[1][2][3]. In the progress of such studies, in November, 2000 we proposed a small biped entertainment robot SDR-3X (Sony Dream Robot, a prototype) which realized the dynamic and elegant motion performances by the advanced and integrated robot actuator ISA (Intelligent Servo Actuator) and the Whole Body Cooperative Dynamic Motion Control [4][5]. SDR-4X is an advanced model and has a capability of a Real-time Integrated Adaptive Motion Control using the enhanced ISA and sensors. It enables real-time adaptive biped walking on unbalanced surface and real-time adaptive motion control against external force. In addition, real-time adaptive and controlled falling over and standing up on a floor are realized. The cooperation of the motion control system and the motion creating system can be applied to realize attractive applications for Motion Entertainment.

2 Basic Configuration of SDR-4X

Fig. 1 shows attractive poses of both SDR-3X and SDR-4X. SDR-3X with a 50 cm height and 5 kg weight, has 24 DOF in its major joints. SDR-4X with a 58 cm height and 6.5 kg weight, has 28 DOF in its major joints. Each leg has 6 DOF, trunk has 2 DOF, each arm has 5 DOF and the neck has 4 DOF. In addition, five independent driving fingers are also attached to each hand. SDR-4X has been designed not to injure humans and also not to get damaged itself when it falls over.

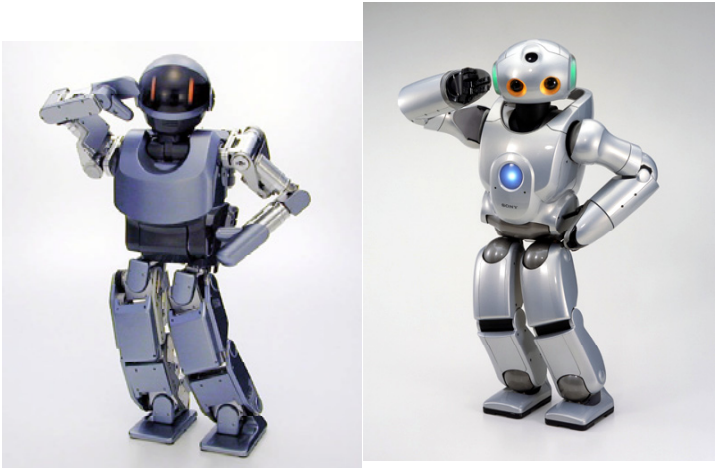


Fig.1. Attractive Poses of SDR-3X and SDR-4X

CPU	64 bit RISC Processor (× 2)
Memory	64MB DRAM (× 2)
Operating Sys. & Architecture	Aperios & OPEN-R 6)
Robot Control Supplying Media	16MB Memory Stick
Input/output	PC Card Slot (Type II)/MS Slot
Image Input (color/stereo)	110,000 pixels CCD
Sound Input/output	7 Microphones/Speaker
Walking Speed	6m/min. max (unleveled surface)
Ability of Adaptive Walking (on non-slip condition)	Approx. 10mm uneven surface
	Approx. 10degrees tilted surface
Weight (including battery)	Approximately 6.5Kg
Dimensions (height/width/depth)	Approximately 580/260/190mm

Safe design including a joint structure that does not trap hands and fingers in between joints has been adopted for safe interaction with humans. **Table 1** shows the basic specifications of SDR-4X.

3 Key Devices

The newly developed robot actuators named ISA-4; Intelligent Servo Actuator for SDR-4X are applied to the driving joints except joints in head and of fingers. Compared with ISA-3 used in SDR-3X, an approximately 30% improvement in the start-up torque, an approximately 15% improvement in rated torque, and an approximately 20% improvement in efficiency are achieved for ISA-4. **Fig. 2** shows the overview of ISA-4. Some typical sensors are used for motion control. They are 3 axes accelerometer and gyro in the trunk, 2 axes accelerometer and 4 force sensors in each foot, and contact sensors in the head and both shoulders. Two small CCD color cameras are used for stereo image recognition detecting the distance between itself and an object. SDR-4X can also detect the direction of a sound source and recognizes an individual speaking by utilizing 7 microphones located inside its head (**Fig. 3**).



Fig.2. Robot Actuator ISA

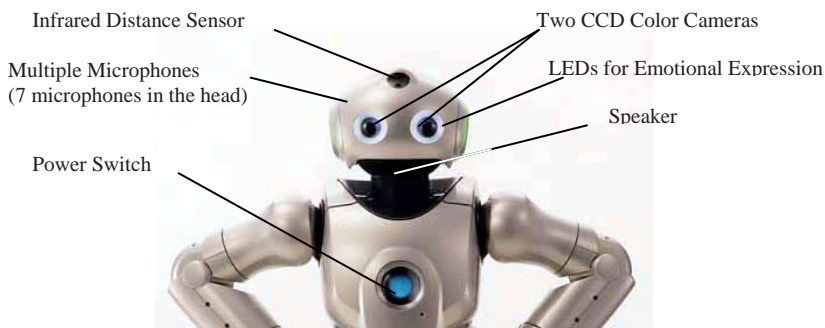


Fig.3. SDR's Typical Embedded Devices

4 Real-Time Integrated Adaptive Motion Control

We developed sensor based Real-time Integrated Adaptive Motion Control System which can be applied to a rough and unlevelled terrain, and retaining posture under external forces. The control system is composed of Real-time Whole Body Stabilizing Motion Control, Real-time Terrain Adaptive Motion Control, External Force Adaptive Motion Control, Real-time Adaptive Falling over Motion Control and Lifting up Motion Control.

4.1 Real-Time Whole Body Stabilizing Motion Control

The Real-time Whole Body Stabilizing Motion Control enables restabilization of whole body motion while biped walking such as asynchronously generated obstacle avoidance and alteration of the upper-body motion in walking. The function includes a real-time solver for the ZMP equation and a gait pattern generator. **Fig. 4** shows the autonomous obstacle avoidance using this function with the implemented stereo image recognition system.

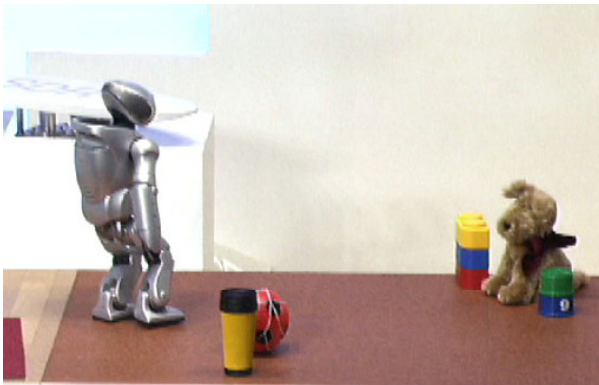


Fig.4. Autonomous Obstacle Avoidance

4.2 Real-Time Terrain Adaptive Motion Control

Using the internal sensors such as force sensors, posture of the robot and inclination of the terrain could be calculated and detected. The Real-time Terrain Adaptive Motion Control has been realized by cooperative control of such sensor based motion control and mechanically adaptive capability. **Fig. 5** shows the ability to walk on inclined and seesaw terrain.

4.3 External Force Adaptive Motion Control

Inclination of the robot caused by external force and moment could be also detected using the internal sensors such as force sensors. The same function described in **4.1** is employed for restabilization of whole body motion. To prevent falling over

under external forces, SDR-4X performs footfall and step back and forth to maintain its stable posture. **Fig. 6** shows the external force adaptive motion performance by intercepting biped walking. It is based on the collaboration of the whole body stabilizing motion control and the external force adaptive motion control.

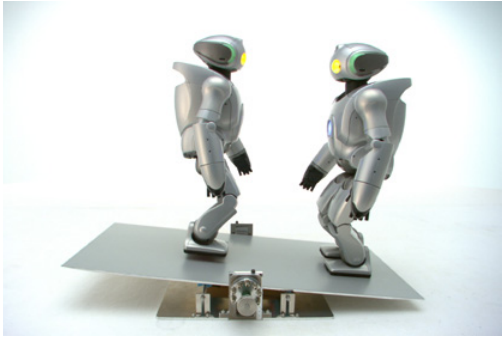


Fig.5. SDR-4X Terrain Adaptive Motion Control



Fig.6. External Force Adaptation

4.4 Real-time Adaptive Falling over Motion Control

As soon as SDR-4X detects the adaptive limitation of the terrain adaptive motion control and the external force adaptive motion control, restabilization of the whole body motion control is suspended and the control is switched to the Real-time Adaptive Falling over Motion Control. When the system detects direction of falling over, the robot is controlled to get into the pose which can adapt and secure against contacting with environment. To reduce the impact of the ground contact the cooperative motion control with the performance of ISA's back-drivability enables compliant control of the joints. **Fig. 7** shows a controlled falling backward motion. After falling over SDR-4X can make standing-up motion.

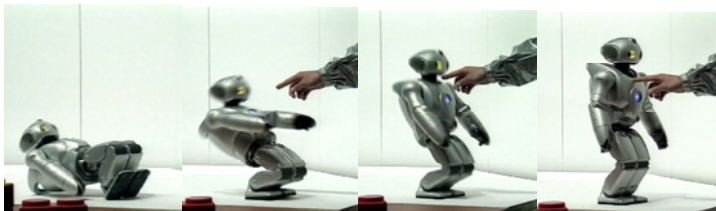


Fig.7. Falling Backward Motion Control

4.5 Lifting up Motion Control

A grip with a touch sensor is utilized on the back of SDR-4X. Lifting up of SDR-4X is detected by this sensor of the grip and the force sensors on the feet. Once it is detected the control is switched to the Lifting up Motion Control and the joints of the lower limbs and the both arms are controlled to make compliant (**Fig. 8**).



Fig.8. Lifting up Motion Control

5 SDR Motion Creator

We have developed a motion creating software system, SDR Motion Creator (**Fig. 9**), which enables the easy production and editing of a variety of motions. It runs on PCs. SDR Motion Creator includes the motion creating and editing user interfaces, the Real-time Whole Body Stabilizing Motion Control module, the same module applied to SDR-4X. The designed motions can be determined to perform by 3D browsing function. SDR Motion Creator has a capability to realize a motion and music synchronized creation using the motion creating track and the music data track.

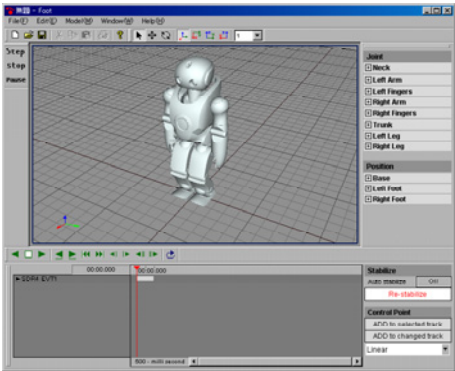


Fig.9. SDR Motion Creator

6 SDR Motion Performances

6.1 High-tempo Dance Performance

To ensure effectiveness of the newly developed robot actuators, ISA-4 and the Real-time Integrated Adaptive Motion Control System, we developed a high-tempo dance performance. Using the SDR Motion Creator the dynamic, stable and music synchronized performance based on 4-unit formation has been realized (**Fig.11**). **Fig.10** shows a high-tempo dance performance by 3 SDR-3X units.



Fig.10. Dance Performance by SDR-3X

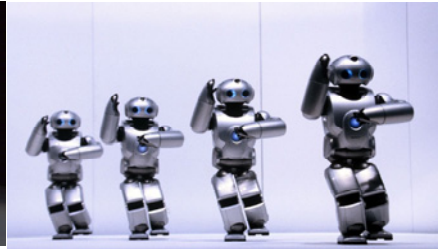


Fig.11. Dance Performance by SDR-4X

6.2 A Cappella Chorus Performance

Based on the speech synthesis the emotionally expressive speech and the production of a singing voice with vibratos have been developed. By cooperative control of this function with the Real-time Whole Body Stabilizing Motion Control enables speech and singing performances synchronized with whole body motion. **Fig.12** shows a cappella chorus performance by 4 SDR-4X units.

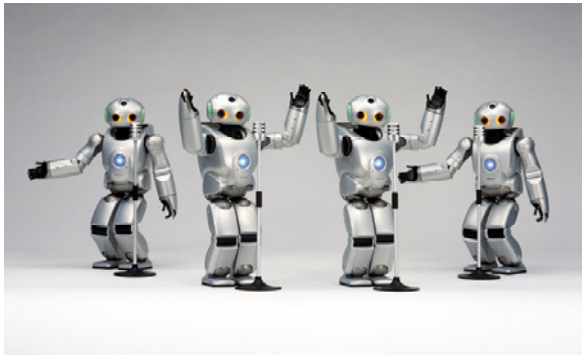


Fig.12. A Cappella Chorus Performance

7 Summary and Conclusions

We have described the core technologies to perform a novel application of a small biped entertainment robot SDR-3X and SDR-4X. SDR-4X is the enhanced prototype model that can adapt its performance to the environment and situations found in the home. Safe design for safe interaction with humans is applied to this SDR-4X. We have developed and enhanced new robot actuators, ISA to realize the dynamic and elegant motion performances. The development of significant technologies for SDR-4X includes the Real-time Whole Body Stabilizing Motion Control, the Real-time Terrain Adaptive Motion Control, the External Force Adaptive Motion Control, the Real-time Adaptive Falling over Motion Control and the Lifting up Motion Control. We have also developed a motion creating software system, SDR Motion Creator. One of the most attractive applications is the Motion Entertainment we have been proposing. We tried to develop a high-tempo dance performance and a cappella chorus performance using the described core technologies and we realized the attractive entertainment applications.

References

1. Jin'ichi Yamaguchi, A. Takanishi and I. Kato "Development of a Biped Walking Robot Compensating for Three-Axis Moment by Trunk Motion", *Proc. of IROS'93*, pp. 561-566 (1993).
2. Jin'ichi Yamaguchi, E. Soga, S. Inoue and A. Takanishi "Development of a Bipedal Humanoid Robot - Control Method of Whole Body Cooperative Dynamic Biped Walking-", *Proc. of the 1999 ICRA*, pp. 368-374 (1999).
3. Ken'ichiro Nagasaka, H. Inoue and M. Inaba "Dynamic Walking Pattern Generation for a Humanoid Robot Based on Optimal Gradient Method", *Proc. of the 1999 SMC*, pp. VI908-VI913 (1999).
4. Ishida, T., Kuroki, Y., Yamaguchi, J., Fujita, M., Doi, T.T. "Motion Entertainment by a Small Humanoid Robot Based on OPEN-R", *IROS*, pp. 1079-1086 (2001).
5. Kuroki, Y., Ishida, T., Yamaguchi, J., Fujita, M., Doi, T.T. "A Small Biped Entertainment Robot", *Proc. of the IEEE-RAS International Conference on Humanoid Robots*, pp. 181-186 (2001).
6. Fujita, M. and Kageyama, K., "An Open Architecture for Robot Entertainment", *Proc. of International Conference on Autonomous Agents*, pp. 435-440 (1997).

Mobile Robots Facing the Real World

Roland Siegwart

Autonomous Systems Lab
Swiss Federal Institute of Technology Lausanne (EPFL)
roland.siegwart@epfl.ch
<http://asl.epfl.ch>

Abstract. Autonomous mobile robots navigating in the real world are facing major challenges. They are expected to adapt best to different environments, interact with them and cope with sensor noise and incomplete information. Some important competencies of mobile robots, including locomotion, environment representation and navigation are discussed in this paper. Two research examples of our Lab demonstrate the application of this concepts: Innovative wheel-based locomotion concepts for rough terrain and a family of 11 tour guide robots with 5 month of operational experience.

1 Introduction

Robots are machines that are getting closer and closer to humans. One way to classify robots is through their closeness to humans, especially if their potential social and psychological impact is of interest (fig. 1). Today around one million

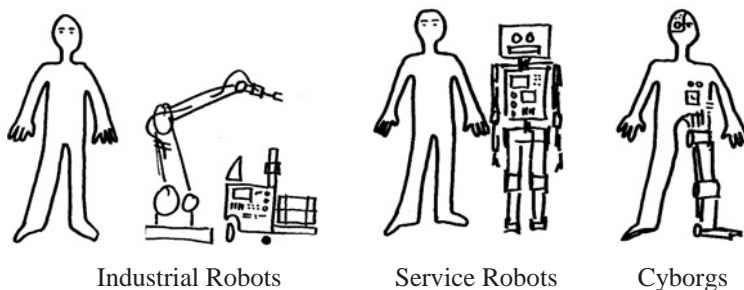


Fig. 1. Classification and Evolution of Robotics: The evolution of Robotics can be characterized by the increasing physical and psychological closeness and interaction between man and machine. *Industrial Robots* are characterized by their power, endurance, speed, precision productivity, and efficiency. *Service* and *Personal Robots* are not physically bound to a fixed position. Their reduced distance between man and machine, their mobility, interactivity, autonomy, adaptability and intelligence characterizes them. Typical applications are in elderly care, housekeeping and entertainment. *Cyborg Robots* are characterized by the fusion of organic-human and technical subsystems. The distance between human and machine vanishes totally. Their characteristics are bio-compatibility, bio-interfacing, distributed intelligence and symbiosis. Cyborg devices can either augment or replace functionality of humans.

industrial robots are doing their work in close cooperation with humans, 24 hours a day, 7 days a week. The second generation of robots, so-called personal or service robots, just started gradually to appear on the market, offering their service in our daily life or just for fun. In contrast to industrial robots, service robots are not physically bound to a fixed position. This offers them a much wider field of activities and applications, but implies also an exponential growth in system complexity. For autonomous operation in real world environments, mobile service robots have to be optimally adapted to their environment, they have to perceive and interpret the surroundings and to interact with it. In many senses, service robots might be considered as the most sophisticated machines, combining and integrating latest technologies from different engineering fields and beyond.

Service robots have to develop their own intelligence in order to interact with highly complex real world environments. The key requirements and research issues for future evolution of service robots are therefore in environment perception and scene interpretation, navigation (localization and map-building), multi-modal interaction, adaptivity and learning. Furthermore, the mechanical and electrical design of the robot system is of key importance, in order to best adapt to various environments and situations.

In this paper some key design issues for real world operation of service robots are presented and discussed, and demonstrated through two research examples of our Lab.

2 Design Issues for Real World Operation

The interaction with the real world can have many facets, from pure mechanical interaction up to probabilistic reasoning about the environment. Here we would like to raise some of the most important issues and questions, and discuss some design considerations as locomotion concepts, environment representation and navigation.

2.1 Locomotion in Real World Environment

Locomotion in different type of terrains can quickly become a major challenge for a mobile robot. Whereas wheels are very well adapted for flat and hard ground, novel concepts are required for rough terrain, where slippage, unequal and soft ground, and various kind of obstacles are present. One well established research axis concentrates on walking mechanisms with various numbers of legs. Even if nature demonstrates efficient and impressive walking concepts for rough terrain, artificial systems suffer still from very high complexity and require, depending on the number of legs, quite a good knowledge of the environment. Various walking robots on one, two or more legs have shown the feasibility of walking, but are still limited on relatively flat and hard ground. Furthermore energy efficiency, systems complexity and weight are an additional obstacle for their application.

A probably more adapted approach for locomotion in rough terrain is to apply the advantages of wheels, and to combine them with appropriate passive or active suspension mechanisms. This is actually a research direction we are following up

in our Lab, that leads to locomotion concepts that perform extremely well in rough terrain, still being efficient and not very complex (fig. 5 and 6).

2.2 Environment Representation

Environment perception and representation can be model- or behavior-based and might involve different levels of abstraction (fig. 2). Whereas behavior-based approaches are often combined with bio-inspired algorithms for adaptation and

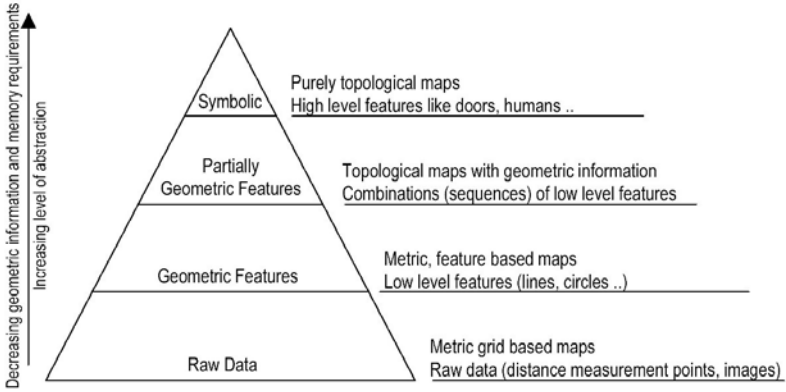


Fig. 2. This figure depicts the hierarchy of abstraction levels. More we go up in the hierarchy, more we reduce the geometric information and more we increase the distinctiveness. For global localization and mapping, high distinctiveness is of importance, whereas for local action, precise geometric relations with the environment come forward.

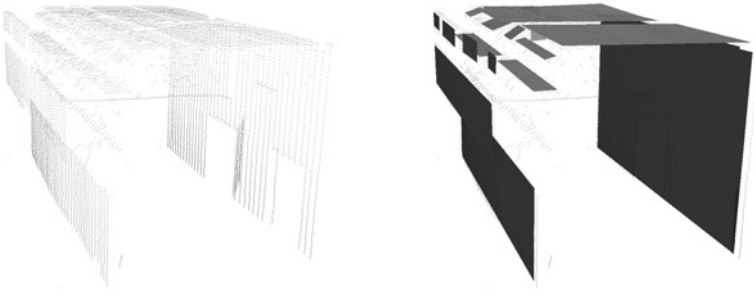


Fig. 3. 3D representation of an indoor environment by planes. The raw data (left) are registered with an upward looking SICK laser scanner, whereas a horizontally arranged laser scanner is used for probabilistic localization of the robot during the measurement. Through the extraction of plans (right) from the 13157 raw data points, the representation can drastically be simplified to 27 planar regions, thus reducing memory requirement and complexity [15], and filtering unimportant information.

learning, they hardly scale with more complex task and have only shown their feasibility in simple experiments. Model based approaches make use of a priori knowledge by means of environment models, thus allow a very compact and task-dependant environment interpretation. They scale much better with the complexity of the system and task, but have some limitations with highly nonlinear systems and unmodeled environment features. However, today only model based control approaches enable predictivity of the systems behavior, thus guaranteeing safety. Models can have different level of abstraction, from raw-data based grid representations up to highly symbolic descriptions of the environment (fig. 2). For real world navigation, the complexity is a major issue. Complexity, especially for localization and mapping, is strongly linked with the environment representation. We therefore strongly believe in a model based environment representation, that drastically reduces the memory requirements and complexity. Furthermore, the use of models of expected environment features enable to filter most measurement that are not relevant to the given navigation task, e.g. people around the robot that are not appropriate features for localization. Representing a typical office room in 2D by raw data based occupancy grids easily requires hundreds of k-bytes if a fine grid of some centimeters is used. However, for localization purposes the same room can typically be represented by around 10 lines, thus requiring only around 20 to 40 bytes dependent of the requested resolution. The same applies even more drastically for 3D representations as presented in figure 3 [15]. If lower precision is required, one might even use a topological map using more abstract and distinct features as the fingerprints presented in [16,17].

A major challenge for a useful environment perception and representation is the inherent noise of sensor systems and the feature ambiguities that are present in most environments. Therefore probabilistic algorithms are adopted in order to extract useful information and to fuse the different signals and modalities to the best estimate of the environment and the robots situation within it. The currently most successful approaches employ Kalman filters [3] or Hidden Markov Models [1] for fusion and estimation. The Kalman filter is well adapted if the environment is represented in a continuous form by geometric features whereas Hidden Markov Models are typically used with metric grid maps or topological maps. Both approaches have their advantages and disadvantages. We therefore propose hybrid approaches, using a topological representation for the global map and a metric representation based on geometric features for the local maps [18]. These approaches enable to combine global consistence, robustness, precision and applicability for large environments.

2.3 Navigation and Control Architecture

Navigation in real world environments is a very complex task. It requires an appropriate control architecture implementing various parallel tasks in real time. A typical autonomous mobile robot system requires at least five control levels, running at different cycle times. The main tasks ordered by importance are motor control, emergency supervision, obstacle avoidance, localization and planning of the task. Apart the motor controller, all other control tasks require information about the local or even global environment of the robot. As discussed in the

pervious section, the perception and representation of the environment can become very complex. Thus the processing power to run these algorithms can be extremely high and therefore real-time implementation a real challenge. Various research projects address this problem with the goal to find new concepts and algorithms for robust and practical navigation in real world environments [1,3]. Today, feasible solutions for typical indoor or flat outdoor environments are available (e.g. RoboX presented below). However, navigation in unstructured and rough terrain, where the environment has to be modeled in real 3D, is still a very open research area [4,5,6].

3 Examples

At the Autonomous Systems Lab at EPFL we conduct focused research in mobile robotics for autonomous operation in real world environments. Major axes are in the field of mobile robot design for rough terrain, navigation and interaction, and in mobile micro-robotics. Among our most recent findings are enhanced feature based localization concepts [3,9,8], obstacle avoidance for highly dynamic and human-cluttered environments, wheeled robots for high performance in rough terrain [12,4,10] and a mobile micro-robot of the size of a sugar cube [19]. Recently we run one of the worlds largest mobile robot installations with eleven fully autonomous and interactive mobile robots at the Swiss exhibition *expo.02*. It represents a milestone in our mobile robotics research and allowed us a long-term evaluation of our recent findings. Furthermore it was used to investigate social and psychological issues of mobile robotics.

In the following two research results are briefly presented and discussed.

3.1 Wheel-Based Mobile Robot Locomotion for Rough Terrain

Wheels enable efficient motion on flat ground, and, if equipped with an appropriate suspension, can reach excellent climbing abilities. In our Lab we therefore investigate new passive and active wheel-based locomotion concepts. Passive means, that the articulations of the suspension have no actuators, whereas active concepts use motors for at least some of the articulations of the suspension system.

Shrimp, presented in figure 4, is a passive system with 6 wheels [12]. It can effortlessly overcome obstacles up to two times of its wheel diameter and climbs regular steps of stairs that are about of its height. Within a running research project for the European Space Agency, the system is extended for full energetic autonomy using solar cells, and equipped with a navigation system for autonomous long-range operation [4].

Octopus, shown in figure 5, features an active locomotion concept on 8 wheels. Its 6 active and one passive articulations enables the robot to keep all wheels in optimal ground contact at any time [10]. A specially developed tactile wheel measures the contact point and force of each wheel.



Fig. 4. The robot Shrimp is an all-terrain rover based on a passive locomotion concept. It is characterized by 6 wheels suspended by parallel mechanisms, one fixed wheel in the rear, two boogies on each side and one front wheel with spring suspension. The robot sizes around 60 cm in length and 20 cm in height, is highly stable in rough terrain and overcomes obstacles up to 2 times its wheel diameter with a minimal friction coefficient.



Fig. 5. Octopus features an active locomotion concept with 8 motorized and tactile wheels, 6 active and 1 passive DOF for ground adaptation and on-board integration of all control elements, joint sensors and inclinometers. (Photo © Bramaz)

3.2 RoboX, the Tour-Guide Robot with Long-Term Experience [11]

The Swiss National Exhibition takes place once in 40 years. The 2002 edition, *expo.02*, ran from May 15 to October 21, 2002. It hosted the exhibition Robotics that was intended to show the increasing closeness between man and robot technology (fig. 1). The central visitor experience of Robotics was the interaction with eleven autonomous, freely navigating mobile robots on a surface of about 315 m². Their main task was giving guided tours but included also a robot taking pictures of visitors. The exhibition was scheduled for five hundred persons per hour. For this task, the main specifications can be summarized as follows:

- Navigation in an unmodified, highly populated environment with visitors and other freely navigating robots
- Bi-directional multi-modal interaction using easy-to-use, intuitive yet robot-typical interaction modalities.
- Speech output in four languages: French, German, Italian and English
- Safety for visitors and robots at all time.
- Reliable operation during around eleven hours per day, seven days per week, during five months
- Minimal manual intervention and supervision
- Adaptive multi-robot coordination scenarios in function of the number of visitors and their interests
- Control of visitor flow through the robots
- Development of ten robots within tight budgets and schedules

The RoboX robot was designed and developed at our Lab by a multi-disciplinary team of around 15 young engineers and artists (fig. 6 and 7). It was then realized by our spin-off company BlueBotics. It features fully autonomous navigation, including feature-based localization [3], highly adaptive and dynamic obstacle avoidance [8] and multi robot coordination on the path planning level [2]. The main interaction functions are face and people tracking, speech output, facial expression through the two pan-tilt eyes and the eye-integrated LED matrix [11]. Four touch buttons were used as input devices and two robots were equipped with a directional microphone and speech analysis for simple answers.

The navigation and interaction software, with around 20 main tasks, was running on two embedded computers. The safety-critical navigation software runs on a XO/2 operating system based on Oberon [13] and the interaction software on an industrial PC running Windows 2000. An additional security controller was running on a PIC micro-controller, guaranteeing visitors safety at all time.

The specially developed interaction software SOUL [20] aimed at composing the scenarios like a theater or a music composition. It enables through a convenient interface to combine different basic behaviors with synthesized speech, motion, sensory inputs and much more.

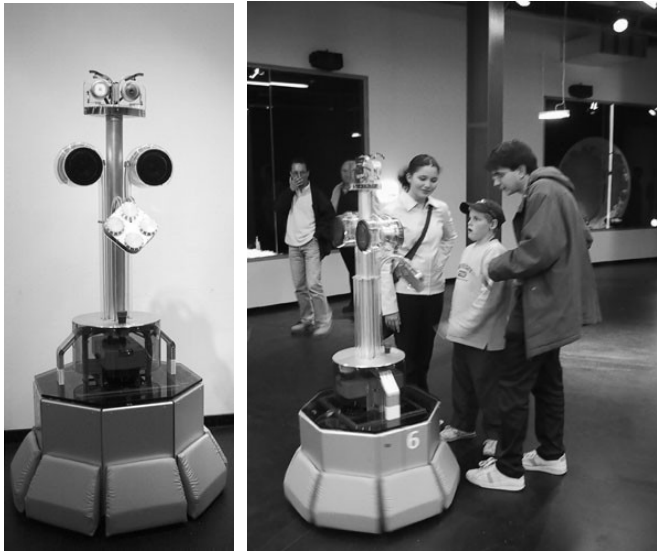


Fig. 6. a) The autonomous exhibition robot RoboX. b) RoboX number 6 with visitors in the pavilion at expo.02

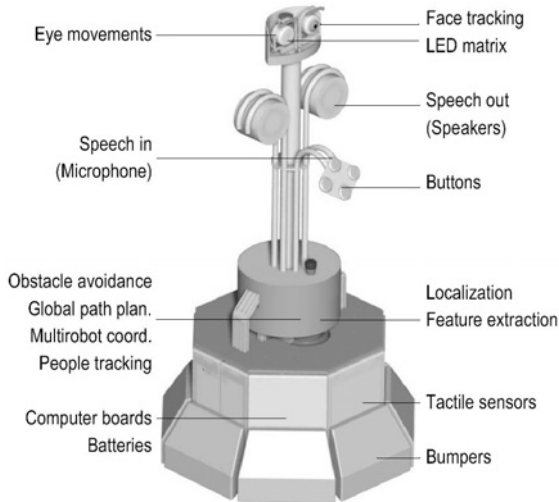


Fig. 7. Basic elements and functionalities of the tour-guide robot RoboX.

Eleven robots were guiding visitors through the exhibition and interacting with them in an environment cluttered by hundreds of visitors. During the five-month exhibition, the RoboX family was in contact with 686'000 visitor and traveled a total distance of 3'315 km. The installation served also as research

platform and technology demonstration. Throughout the five-month operation period, the navigation system was close to 100% reliable. This was especially due to the localization system that was based on line feature [2], thus filtering out all the dynamics in the environment coming from the visitors in the vicinity of the robot. More details on the hardware design, the navigation system and the reliability can be found in [2,8,11,14].

4 Conclusions and Outlook

New concepts for wheeled locomotion, feature based environment representation and navigation have been presented and discussed in this paper. Their potential was shown by two examples of mobile robots system of our Lab, facing the complexity of the real world. They represent our recent findings, but are still only a very first step towards intelligent and socially interactive robots. In order to realize really intelligent mobile robots, able to scope with highly complex real world environments, enormous research efforts in various fields like environment representation, cognition and learning are still required.

Acknowledgments

The author would like to thank all the current and past collaborators of the Autonomous Systems Lab for their contributions and inspiring work, their curiosity and dedication for mobile robotics research. The presented projects were mainly funded by EPFL, the European Space Agency (ESA) and expo.02.

References

1. S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo Localization for Mobile Robots," In Artificial Intelligence (AI), 2001.
2. Arras, K.O., Philippsen, R., Tomatis, N., de Battista, M., Schilt, M. and Siegwart, R., "A Navigation Framework for Multiple Mobile Robots and its Application at the Expo.02 Exhibition," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03), Taipei, Taiwan, 2003
3. Arras, K.O., Castellanos, J.A. and Siegwart, R., Feature-Based Multi-Hypothesis Localization and Tracking for Mobile Robots Using Geometric Constraints. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'02), Washington DC, USA, May 11 - 15. 2002.
4. Lamon, P. and Siegwart, R., "3D-Odometry for rough terrain – Towards real 3D navigation." In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03), Taipei, Taiwan, 2003.
5. Singh S., Simmons R., Smith T., Stentz A., Verma V., Yahja A., Schwehr K., "Recent Progress in Local and Global Traversability for Planetary Rovers", Proceedings of IEEE International Conference on Robotics and Automation (ICRA'00), p1194-1200, San Francisco, April 2000.
6. A. Mallet, S. Lacroix, and L. Gallo, "Position estimation in outdoor environments using pixel tracking and stereovision", Proceedings of IEEE International Conference on Robotics and Automation (ICRA'00), pages 3519-3524, San Francisco, CA (USA), April 2000.
7. Fong T., Nourbakhsh I., Dautenhahn K., "A survey of social interactive robots," Journal of Robotics and Autonomous Systems, 42, 143-166, 2003

8. Philippsen, R. and Siegwart, R., "Smooth and Efficient Obstacle Avoidance for a Tour Guide Robot," In Proceedings of IEEE International Conference on Robotics and Automation, (ICRA'03), Taipei, Taiwan, 2003.
9. Tomatis, N., Nourbakhsh, I. and Siegwart, R., "Hybrid Simultaneous Localization and Map Building: Closing the Loop with Multi-Hypotheses Tracking," In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'02), Washington DC, USA, May 11 - 15, 2002.
10. Lauria, M., Piguet, Y. and Siegwart, R., "Octopus - An Autonomous Wheeled Climbing Robot," In Proceedings of the Fifth International Conference on Climbing and Walking Robots. Published by Professional Engineering Publishing Limited, Bury St Edmunds and London, UK, 2002.
11. Siegwart R., et al., "Robox at Expo.02: A Large Scale Installation of Personal Robots," Special issue on Socially Interactive Robots, Robotics and Autonomous Systems 42 (3-4), 31 March 2003
12. Siegwart R., Lamon P., Estier T., Lauria M., Piguet R., "Innovative Design for Wheeled Locomotion in Rough Terrain," Journal of Robotics and Autonomous Systems, Elsevier Sep. 2002, Vol. 40/2-3, pp 151-162
13. Brega, R., N. Tomatis, K. Arras, and Siegwart R., "The Need for Autonomy and Real-Time in Mobile Robotics: A Case Study of XO/2 and Pygmalion," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00), Takamatsu, Japan, 2000.
14. Siegwart, R., Arras, K.O., Jensen, B., Philippsen, R. and Tomatis, N., "Design, Implementation and Exploitation of a New Fully Autonomous Tour Guide Robot," In Proceedings of the 1st International Workshop on Advances in Service Robotics (ASER'2003), Bardolino, Italy, 13-15 March 2003.
15. Weingarten, J., Gruener, G. and Siegwart, R., "A Fast and Robust 3D Feature Extraction Algorithm for Structured Environment Reconstruction," Proceedings of 11th International Conference on Advanced Robotics, Portugal, July 2003.
16. Lamon, P., I. Nourbakhsh, et al., "Deriving and Matching Image Fingerprint Sequences for Mobile Robot Localization," Proc. of IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea, 2001
17. Lamon, P., Tapus A., et al., "Environmental Modeling with Fingerprint Sequences for Topological Global Localization" - submitted at IROS'03, Las Vegas, USA, 2003.
18. Tomatis N., Nourbakhsh I. and Siegwart R., "Hybrid Simultaneous Localization and Map Building: Closing the Loop with Multi-Hypotheses Tracking," In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'02), Washington DC, USA, May 11 - 15, 2002.
19. Caprari G., Estier T., Siegwart R.: "Fascination of Down Scaling - Alice the Sugar Cube Robot, Journal of Micro-Mechatronics," VSP, Utrecht 2002, Vol. 1, No. 3, pp. 177-189.
20. Jensen, B., Froidevaux, G., Greppin, X., Lorotte, A., Mayor, L., Meisser, M., Ramel, G. and Siegwart, R. (2003) "Multi-Robot Human-Interaction and Visitor Flow Management," In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03), Taipei, Taiwan, 2003.

Breakthroughs in Human Technology Interaction

Bernd Reuse

Federal Ministry of Education and Research, Germany

Abstract. In 1999 the German Federal Government launched six major strategic collaborative research projects on Human Technology Interaction, which involved 102 research partners and a funding volume of 82 million. The results of these projects were expected to allow people to control technical systems multimodally by using natural forms of interaction such as speech, gestures, facial expressions, touch and visualization methods and to apply such systems for the most varied purposes in their private and working environments. The ambitious research goals were achieved with prototypes for real-world applications. Research activities have resulted in 116 patent applications, 56 spin-off products and 13 spin-off companies as well as 860 scientific publications.

1 Introduction: Trends in Human Technology Interaction

Together with the Federal Ministry of Economics and Labour (BMWA), the Federal Ministry of Education and Research (BMBF) organized an international status conference in Berlin in June 2003, where the results of four years of government-funded research on Human Technology Interaction (HTI) were presented (www.dlr.de/pt-dlr/sw).

Distinguished personalities from science, research and industry participated in this conference. The roughly 350 conference participants from Germany and abroad agreed on the following trends in Human Technology Interaction:

- Human Computer Interaction is turning into Human Computer Cooperation and will support many trends in I&C. The number of transactions is increasing dramatically and requires new modalities in HTI. Access to any information with any device at any place and at any time will be supported by HTI interfaces. Agents will take over routine work.
- A simple and easy-to-handle Human Computer Interface is an important precondition for marketing the products of the IT industry.
- Human Computer Interaction can help solve the problems of the future, namely the problems of the aging society. An intelligent human life has to be supported by science and technology.

- Human Computer Interaction will have a strong influence on society as a whole as a result of the convergence of the use of computers in private and business environments, of working life and leisure time, and of paid and unpaid work.

2 The Verbmobil Project

After 20 years of largely unsuccessful research in the field of speech recognition, which had only produced simple dialogue systems, the BMBF decided in 1993 to provide a total of approximately 60 million for an eight-year research project entitled **Verbmobil** which was to deal with the automatic recognition and translation of spontaneous speech in dialogue situations. This was not in line with the then prevailing trend in research, and even some of the international experts who were involved took the view that the goal could not be achieved at that time.

But in the Verbmobil project we pursued new paths in research: first of all, mastering the complexity of spontaneous speech with all its phenomena such as vagueness, ambiguities, self-corrections, hesitations and disfluencies took priority over the envisaged vocabulary. Another novel feature was that researchers used the information contained in the prosody of speech for speech recognition purposes. In addition, the transfer included knowledge processing, which is indispensable in translation. The 135 different work packages and 35 research groups distributed throughout Germany were linked by a network management led by Professor Wahlster of the German Research Centre for Artificial Intelligence (DFKI) in Saarbrücken.

As a result of Verbmobil it was possible to demonstrate in July 2000 the translation of spontaneous speech for the domain of the remote maintenance of PCs using 30,000 words and for a telephone translation system with 10,000 words for translation from German into English and with 3,000 words for translation from German into Japanese. In addition Verbmobil generated 20 spin-off products, 8 spin-off companies and about 800 scientific publications. In 2001, Verbmobil received Federal President Rau's German Future Award, the highest German research award.

3 Lead Projects on Human Technology Interaction

In 1999 the Federal Government started an initiative on Human Technology Interaction, the central goal being to extend the findings and models of the Verbmobil project concerning speech-based human interaction with computers to cover the full range of human forms of interaction.

It was expected that the consideration and integration of several forms of interaction would allow a much better interpretation of the user's intention than one

modality alone. This initial view has now been confirmed on a global scale at relevant international conferences.

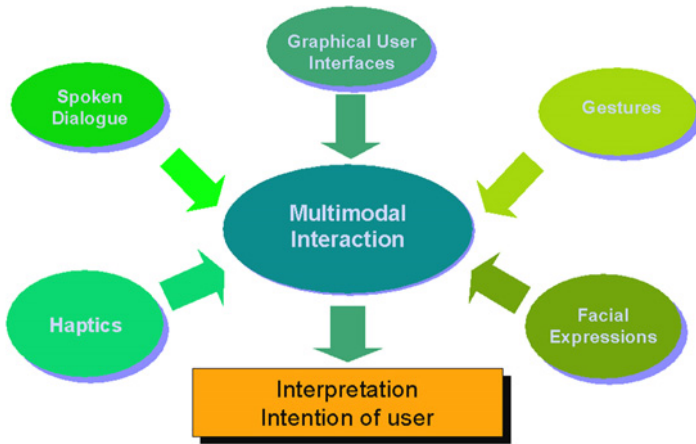


Fig. 1. Aspects of multimodal interaction

The Federal Government therefore staged an ideas competition on Human Technology Interaction. Altogether 89 outline proposals involving 800 cooperation partners from science and industry were submitted. In a two-tier process, international experts selected six major strategic and interdisciplinary collaborative research projects (lead projects) involving 102 partners from science and industry. These projects were supported with 82.4 million in government funds (to which industry added 69.7 million of its own funds) between July 1999 and September 2003.

The results of the projects were expected to allow people to control technical systems multimodally by using natural forms of interaction such as speech, gestures, facial expressions, touch and visualization methods and to apply such systems for the most varied purposes in their private and working environments. The aim was to adapt technology to people and not vice versa as has been the case in the past. Ergonomics and user acceptance of the various forms of interaction were major criteria for the development of prototypes which should not only be highly attractive from the scientific viewpoint but should also have a great market potential.

The following is an overview of the projects that were carried out. It should be noted that a basic-science project entitled SMARTKOM succeeded in generalizing the advanced discourse models for spoken dialogue to cover the full spectrum of multimodal discourse phenomena. The other projects covered the entire range from basic to applied research but were clearly more application-oriented.

SmartKom: Dialogue-based Human Computer Interaction through Coordinated Analysis and Generation of Multiple Modalities (www.smartkom.org).

Computer without keyboard and mouse - integrated processing of speech and gestures for natural interaction with the system. Even vague, ambiguous and incomplete input is understood. The computer is controlled by means of gestures and facial expressions and recognizes frowning as a sign of non-understanding on the part of the user (10 project partners; project leader: German Research Centre for Artificial Intelligence, DFKI). Highlights:

- Situation-based understanding of vague, ambiguous or incomplete multimodal input at the semantic and pragmatic level
- Development of a Meaning Representation Language M3L.

ARVIKA: Augmented Reality (AR) for Development, Production and Services (www.arvika.de). The computer in your spectacles - mobile action in mixed real and virtual future-oriented working environments. Situation-related information is displayed to service engineers on the spot (22 project partners; project leader: Siemens AG, Nürnberg). Highlights:

- First prototypes of mobile Augmented Reality systems for industrial applications in development, production and service
- Remote Augmented Reality support for service processes.

EMBASSI: Multimodal Assistance for Infotainment and Service Infrastructures (www.embassi.de). Multimodal remote control for all electronic appliances in everyday life ensures clear and intelligent user interfaces and operating instructions. Individually adaptable access to public terminal systems is possible (18 project partners; project leader: Grundig AG, Nürnberg). Highlights:

- Living room of the future – multimodal assistance in the selection of entertainment programmes and the control of living room equipment
- Adaptive driver assistance for significantly increasing traffic safety.

INVITE: Intuitive Human Computer Interaction for the Interlaced Information World of the Future (www.invite.de). Multimedial, multimodal and multi-local team work - tools for an innovative exchange of information and knowledge. Implicit recording of information for customer advice and support (20 project partners; project leader: ISA GmbH, Stuttgart). Highlights:

- Exploration and extraction of knowledge structures by integration of speech recognition, text mining and ontology building
- Interaction and collaborative data representation in immersive and distributed 3D-environments.

MAP (BMW): Multimedia Workplace of the Future (www.map21.de). New technical solutions for mobile activities through integration of multimodal interaction functions, new assistance systems, agent technologies and multimedia methods (15 project partners; project leader: Alcatel SEL AG, Stuttgart). Highlights:

- Secure mobile agent systems for personal assistance in stationary and mobile working situations
- Delegating tasks, appointment negotiations by personal agents.

MORPHA: Intelligent service robots to interact and collaborate with human users (www.morpha.de). Mobile service robots are advancing - mainly in private households and in long-term care. Robots working in human environments or interacting with people must be able to recognize people and adapt their movements to suit them. It must be possible to teach robots quickly and intuitively through gestures and demonstration (17 project partners; project leaders: Delmia GmbH, Fellbach, FAW, Ulm). Highlights:

- New interactive programming paradigms for robots through seamless integration of tactile, gesture and speech channels, programming by touch
- Reactive, collision-free motion generation for housekeeping and manufacturing assistance.

3.1 Selected Demonstrators

The projects produced a total of 150 demonstrators; four of them are presented in the following:

SMARTKOM developed systems which allow the speech-based sending of e-mails and selection of music on demand using MP3 over the Internet from a running car.



Fig. 2. SMARTKOM: Sending email from a driving car

ARVIKA developed an innovative cabling method for the Airbus. Instead of connecting the hundreds or thousands of cables in aircraft or other technical systems by using cable lists, engineers will in future be able to find the right connection via Augmented Reality by speech control of cable numbers.

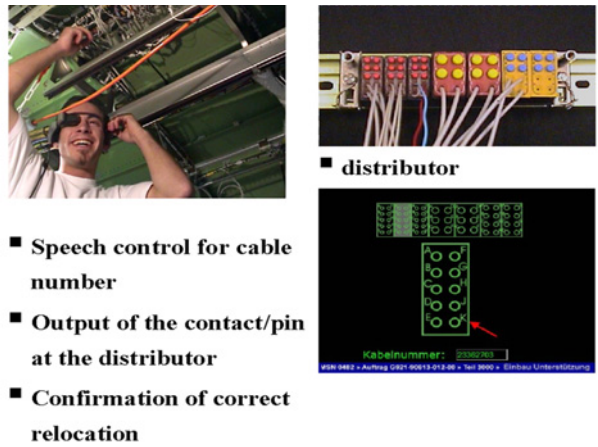


Fig. 3. ARVIKA: Connecting cables in airplanes with AR support

INVITE developed a system which is based on speech processing and can extract the content of a conversation between several participants and present it in a graph. This graph is generated while the device is listening; the information it contains can be used for further purposes, e.g. for identifying the points of the conversation which have or have not been settled.

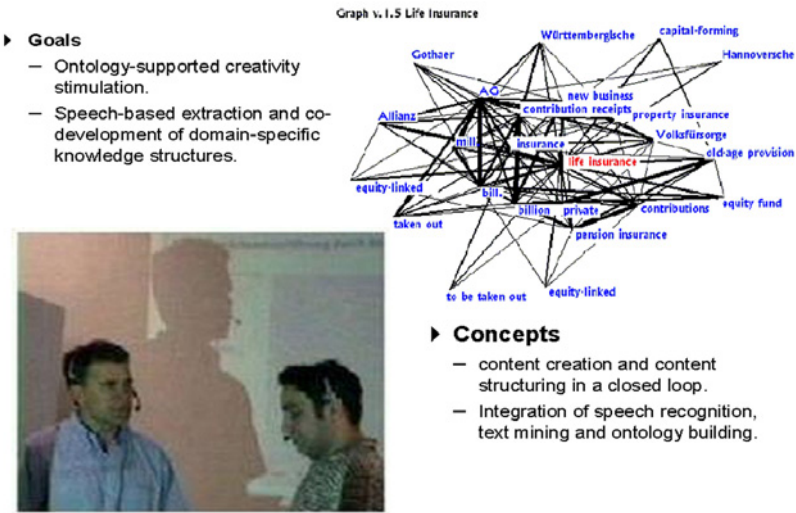


Fig. 4. INVITE: Exploration and Extraction of Knowledge Structures: Goals and Concepts

MORPHA developed a service robot for the domestic environment which can fulfil many functions that are needed by elderly or disabled persons in their private sphere, e.g. it can offer drinks, assist them in walking and carrying, and provide communication support.



Fig. 5. MORPHA: Interaction and communication with robot assistants

4 Results of the Lead Projects on Human Technology Interaction

The lead projects on Human Technology Interaction have produced a considerable number of results which are of great scientific and commercial importance. A scientific advisory board with international membership confirmed the overwhelming success of the HTI programme and pointed to the considerable progress made by research in the field of Human Technology Interaction (<http://informatiksysteme.pt-it.de/mti-2>).

■ Commercial Results

- 150 demonstrators
- 116 patent applications
- 56 spin-off products
- 13 spin-off companies
- 154 newly created high-tech jobs (up to 7/2003)

■ Scientific Results

- 862 scientific publications
- 15 appointments to university chairs
- 208 academic degrees (masters, PhDs, habilitations)
- 1 European Information Technology Prize 2003 (EU)

Fig. 6. Commerical and scientific results

The BMBF has so far organized four events together with German industry in order to promote the quick transfer of the results yielded by the research projects. The BMBF and BMWA will present these results to an international public at the CeBIT 2004 in Hannover (March 18 to 24), where they have reserved an area of about 1000 square meters in exhibition hall 11 (research).

Owing to this big success, the topics of the German lead projects on Human Technology Interaction have been included in the EU's Sixth Research Framework Programme.

Expression of thanks:

My special thanks go to the DLR project management group in Berlin Adlershof, above all to Dr. Grote and Dr. Krahle.

Indoor Navigation for Mobile Robot by Using Environment-Embedded Local Information Management Device and Optical Pointer

Tsuyoshi Suzuki¹, Taiki Uehara², Kuniaki Kawabata³, Daisuke Kurabayashi⁴, Igor E.Paromtchik³, and Hajime Asama³

¹ Department of Information and Communication Engineering, Tokyo Denki University, 2-2, Kanda-Nishiki-Cho, Chiyoda-ku, Tokyo, 101-8457, JAPAN

tszk@ieee.org

² EPSON KOWA Corp., Ueda, Nagano, JAPAN

³ The Institute of Physical and Chemical Research (RIKEN), Wako, Saitama, JAPAN

⁴ Tokyo Institute of Technology, Meguro, Tokyo, JAPAN

Abstract. The paper discusses a new hybrid navigation strategy for mobile robots operating in indoor environment using the Information Assistant (IA) system and the Optical Pointer (OP). For intelligent navigation, the robots need a static and global information describing a topological map such as positional relation from any starting position to any goal position for making a path plan as well as dynamic and local information including local map, obstacles, traffic information for navigation control. We propose a method for managing the information. The robot has only rough path information to the goal, and the IAs, which are small communication devices installed in the environment, manage real environment information, locally. The OP is used for guidance of a robot in the junctions such as crossing, which communicates with mobile robots through IA and indicates their target positions by means of a light projection from a laser pointer onto the ground. The mobile robot allows it and run after the laser light beacon and reaches the destination. The robot can navigate to the goal efficiently by using these systems.

1 Introduction

In recent years, an application range of a robot is spreading to general indoor environment with many indefinite elements from limited condition environment such like in a factory where the industrial robots are used. In such a background, it is important subject for a mobile robot research to be navigated a robot to a goal correctly. Although, there are many research for robot navigation[e.g.1], the robot is beforehand given a detailed map in almost all researches. In such methods, however, it is difficult to cope flexibly according to problems which occur in the situation of changing every moment. Management of the information according to every change of the environment is also difficult.

The paper discusses a new navigation method for multiple mobile robots operating in indoor environment. In order to realize a flexible navigation, a management method of environment information is considered. For intelligent navigation, the robots need two types of environment information. That is, they are static and global information

describing a topological map[2] such as positional relation from any starting position to any goal position for making a path plan and local information including local map, obstacles, traffic information for dynamic navigation control. The robot has only simple path information to the goal and the Information Assistant (IA), which is small communication devices installed in the environment, manage real environment information, locally. In addition, the Optical Pointer (OP) is used for guidance of a robot in the junctions such as crossing, which communicates with mobile robots through IA and indicates their target positions by means of a light projection from a laser pointer onto the ground. The mobile robot allows it and run after the laser light beacon and reaches the destination. The robot can navigate to the goal efficiently by using these systems. There is research which is the robot itself judging all information in the environment for navigation by using improved topological map[3]. However, it is difficult to share the information with other robots. By using IA and OP, more flexible navigation and information management are expected. Section 2 presents the proposed management method of environment information. A navigation system by using the method is developed in a section 3. Section 4 illustrates experimental results. The conclusions are given in section 5.

2 Map Information Management for Navigation

In this research, an indoor environment enclosed with the wall which consists of a passage and a junction is assumed. For flexible robot navigation, we classify two types of environment information. One is global information which is used path planning roughly. The other is local information which is detailed information depending on the place. The local information is used for accurate navigation control with in a path. The management methods of these information are described in this section.

2.1 Global Map Expression for Path Planning

In order to plan the path to the goal, map information that shows the position relation of a present location, goal and passing point roughly are needed. We decided that such global information is given as a topological map. For example, topological map of the environment shown in Fig. 1 (a) is represented as Fig. 1 (b) by using graph expression. In this map, accuracy of the environment information such as position distance etc. is disregarded. Only the simplified position relation of the environment is expressed with this map. A robot can plan a rough path for navigation by using graph search method on the basis of this global map information given beforehand.

2.2 Local Information for Actual Navigation

Although a rough path is planned by the global environmental map, it is necessary for the robot to determine the trajectory to which a robot actually runs in the path. For example, the robot can easily runs along wall in a passage. However, the robot

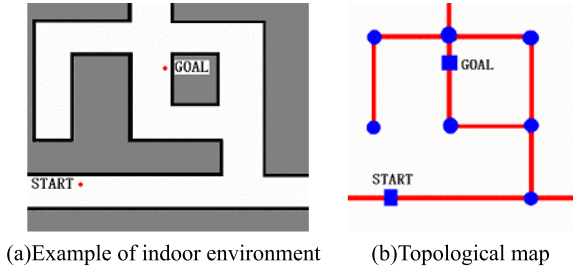


Fig. 1. Global map expression

cannot run easily on a junction. Since the form of junction is not uniform, it cannot give the trajectory beforehand. Moreover, it is needed a traffic control because passing of other robots and man's crowd into the junction. Furthermore, since a wall does not exist in a crossing, the run position control accompanied by self-position identification of a robot is difficult. For smooth navigation in a junction, the robot needs to know the state of information of junction which will run. However, such information depends on the each place and state. Then, we define such information as local information and make the environment itself to manage it. The robot can plan the suitable run trajectory in the local place by acquiring local information from the environment.

3 Local Information Management for Navigation

3.1 Information Assistant and Optical Pointer

In this research, the local information is managed a Information Assistant (IA) which is radio accessing device. IA is the small device with functions, such as managing local information, communication with a robot, and control of connected another device such as sensor. IAs are embedded the junction or other place with local information. By using the IA Reader/Writer, the robot can read the information from IA, and also write the information which the robot has to IA. Furthermore, communication is also possible by other robots reading the information which the robot wrote in IA.

IA gives a information according to a demand of a robot for trajectory planning. However, it is difficult for the robot to run the trajectory correctly only on the basis of this information. For example, Figure 2 shows a arrival point that the robot run a junction by using it's odometer. This shows that arrival points are not fixed. The following things can be considered as this cause.

1. Since the robot detects IA by the electric wave, variation arises at time to detect. Therefore, start position which the robot moves in junction is not fixed.

2. Detection error of the angle between wall and robot occurs when a robot starts revolution to the trajectory direction.
3. The run error at the time of crossing occurs.

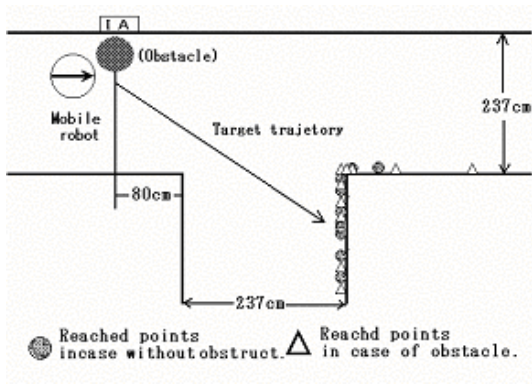


Fig. 2. Experiment of mobile robot operation using IA

In addition, the error is larger when an obstacle exists near a start point. Then, we use a Optical Pointer (OP)[4]. OP has pan-tilte mechanism, and can points the sub-goal of the robot by a laser spot. OP is controlled by IA. The robot can detect laser spot by using CCD camera and image processing, then move to near by laser spot. OP can give a trajectory to the robot by moving laser spot. Figure 3 shows a concept of this optical guidance system.

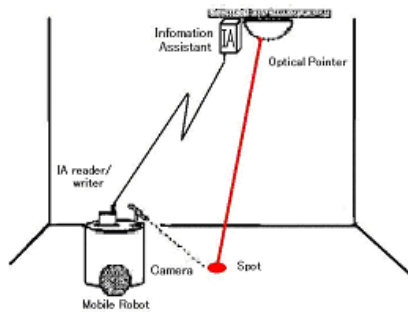


Fig. 3. Concept of the optical guidance system

3.2 Navigation Algorithm of the System

The navigation algorithm by using the above method is shown as follows:

1. The robot is given the topological map of the whole environment and a present location.

2. The robot is given a command of goal information through wireless LAN from an operator.
3. On basis of start and goal information, the robot plans a rough path with referring a topological map by using graph search method.
4. The robot starts to move.
5. The robot runs along wall in a passage direction of the goal according to a path planning. The robot can also detect a obstacle by infrared sensor, and avoid it.
6. When the robot arrives at a junction, it will detect the response from IA. Then, the robot communicates with IA and requests the guidance in the junction. Figure 4 shows a communication process between the robot and the IA.
7. The IA receives a start and goal information from the robot, and give a trajectory information to the robot with referring a local information.
8. The IA gives a laser spot as a target point for the robot movement by using OP on basis of the trajectory.
9. The robot detects a laser spot from OP which controlled by IA, then moves to the spot. IA guides a robot with moving a laser spot along the trajectory in a junction.
10. When the robot arrives a another passage, the robot sends a arrival message to the IA. Then, the guidance by IA is finished.
11. The robot starts an along wall run again.
12. When the robot puts in another junction, the optical guidance is executed. If the robot detect a goal information form IA, the task is finished.

In addition, in the case of many robot move into the junction simultaneously, the robot which communicated first with IA receives guidance service. Other robots receives a busy signal from IA, then wait on current position until service becomes possible. Therefore, navigation is possible even if two or more robots exist in the environment.

4 Navigation System

In order to realize a proposed navigation, the prototype system is developed. This section describes a details of the system.

4.1 Omni-Directional Mobile Robot

The omni-directional mobile robot ZEN has been already developed for realizing flexible action[5](Fig. 5 (a)). The control system is mounted on the robot. Batteries are also mounted on the robot for electrical devices and actuators. The robot can behave autonomously and independently.

The robot has an infrared sensor system called LOCISS (Locally Communicable Infrared Sensory System)[6] shown in Fig.5 (b). The robot can detect a wall and run along wall by using LOCISS, and can also detect a obstacle and avoid it.

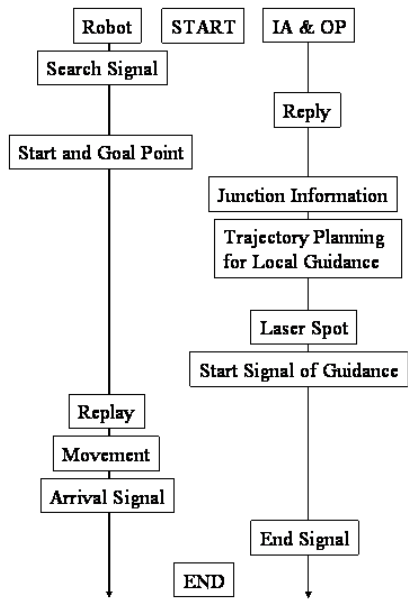
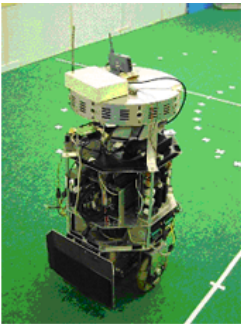


Fig. 4. Communication between robot and IA



(a) Omni-directional mobile robot ZEN



(b) LOCISS

Fig. 5. Omni-directional mobile robot

4.2 Information Assistant and Optical Pointer

Figure 6 (a) shows a Intelligent Data Carrier (IDC)[7] and IDC Reader/Writer. IDC is a small device which consists of radio communications part, CPU, memory and battery. The robot can communicate with IDC through IDC Reader/Writer. In this research, we used this IDC ver.4 as IA, and installed in the environment. For local navigation in a junction, OP (Optical Pointer) was developed. Figure 6 (b) shows a OP which is installed in the ceiling. OP consists of two stepping motors for pan-

tilte motion, two motor driver for driving stepping motor and a laser pointer. OP is connected to IA by the cable. IA controls the actuators and laser pointer of OP. It is enabled to give a laser spot flexibly on the floor surface.

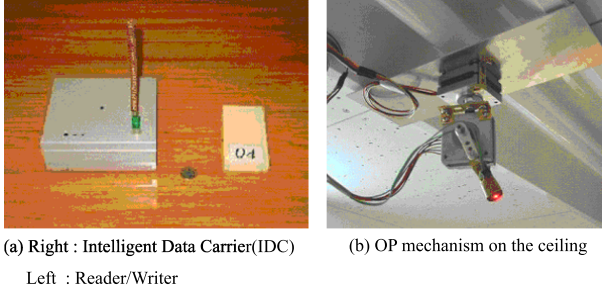


Fig. 6. IA and OP

5 Experiments

The proposed system was installed to an indoor environment where was the second floor of our building in RIKEN. Figure 7 (a) shows the floor map of experimental environment. The task of the robot is movement from its present location to a goal which is given by the human. IA and OP are installed in all junction of the floor. They have information about own position and junction state. The circles A, B and C of Fig. 7 (a) show junctions where the IA-OP system is installed and also show communication and local navigation range of each IA-OP system.

Experimental result is also shown in Fig. 7. The robot had information of current position as START, and was given a goal such as shown in Fig. 7 (a). Then, the robot planed a rough path such like START - A - B - C - GOAL with referring a topological map by using graph search method. After that, the robot run along the wall based on a rough path by using LOCISS, and can also avoid a obstacle. When the robot approached to a junction, the robot detected and communicated with IA. The IA projected a laser spot onto a floor surface as a target point for the robot movement by using OP. The IA provided a trajectory to the robot by moving a laser spot. Figure 7 (b) shows a trajectory which is given a laser spot movement in junction A. Likewise, the robot could reach the goal by using both of an along wall run based on rough path planning by using topological map and the local navigation in junction B and C. The dotted line in Figure 7 (a) shows the actual trajectory of the robot. Figure 8 shows a situation of local navigation in junction.

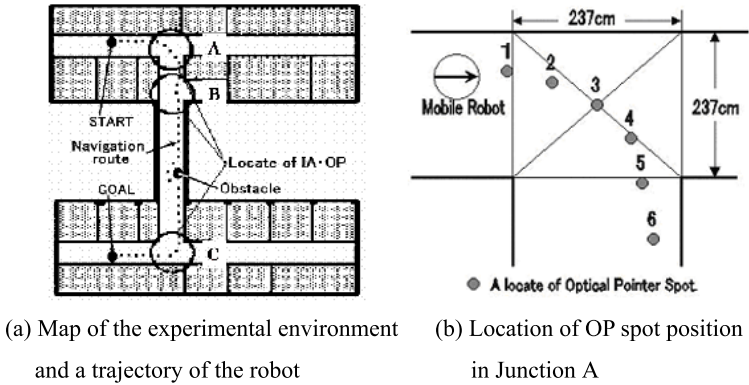


Fig. 7. Experimental Result

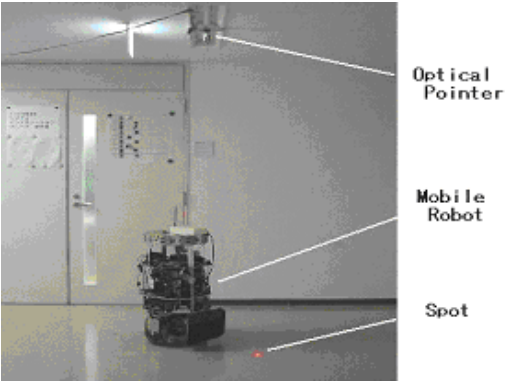


Fig. 8. Robot navigation experiment using IA and OP

6 Conclusion

The new navigation strategy for mobile robots operating in indoor environment is proposed. The robots need a static and global information describing a topological map such as positional relation from any starting position to any goal position for making a rough path plan as well as dynamic and local information including local map, obstacles, traffic information for accurate navigation control. For intelligent navigation in indoor environment, we classified the information into two types such as global information and local information and proposed the method for managing each type using the IA system and the OP. The navigation algorithm by using the system was presented. The experimental example of navigation by using this system is shown. The robot could navigate to the goal efficiently by using the systems.

References

1. M.J.Matarib, "Environment Learning Using a Distributed Representation," *Proc of IEEE Int. Conf on Robotics and Automation*, pp. 402–406, 1990.
2. M.J.Matric, "Integration of Representation into Goal-Driven Behavior-Based Robots," *IEEE Trans. On Robotics and Automation*, vol. 8, pp. 304–312, 1992.
3. M. Tomono and S. Yuta, "Mobile Robot Localization based on an Inaccurate Map," *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS2001)*, pp. 399–405, 2001.
4. Igor E.Paromtchik, "Optical Guidance System for Multiple Mobile Robots.," *IAV Design of Multi-Robots*, City, State, pp. 306–310, 2001.
5. H. Asama, et. al., "Development of an Omni-Directional Mobile Robot with 3 DOF Decoupling Drive Mechanism," *IEEE Int. Conf. on Robotics and Automation*, City, State, pp. 1925–1930, 1995.
6. Y. Arai, et. al., "Collision Avoidance among Multiple Autonomous Mobile Robots using LOCISS (Locally Communicable Infrared Sensory System).," *Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation*, City, State, pp. 2091–2096, 1996.
7. Y. Arai, et. al., *Self-Localization of Autonomous Mobile Robots using Intelligent Data Carriers*, Distributed Autonomous Robotic Systems 2, Springer-Verlag, pp.401–410, 1996.

Wall Following with Constrained Active Contours

Elliot S. Duff and Jonathan M. Roberts

CSIRO Manufacturing & Infrastructure Technology
Queensland Centre for Advanced Technologies
PO Box 883 Kenmore QLD 4069 Australia
{firstname}.{lastname}@csiro.au
<http://www.cat.csiro.au/automation>

Abstract. This paper presents a new wall-following algorithm for reactive navigation. It is based upon constrained active contours in a repulsive potential field. The development of this algorithm was initiated by the need for robust and efficient software for the navigation of large underground autonomous vehicles.

1 Introduction

One of the key issues for underground automation is navigation. Both absolute and reactive navigation techniques have been, and are being developed to guide autonomous vehicles in mine tunnels. Absolute navigation techniques that rely on *a priori* maps of the mine are not suitable in the underground environment because, by their very nature, mines change every day. To rely on a potentially erroneous map information is hazardous. The most appropriate navigation technique at this stage is reactive navigation [1].

Although the underground mining environment is unstructured in a robotic sense, it is topologically highly structured. It consists of well defined paths (tunnels with walls), intersections, muck piles and ore passes. The navigation of a vehicle in such an environment can be compared to the task of rally driving, where the navigation of the vehicle is shared between two people. One person is concerned simply with moving the vehicle forward and staying on the road, whilst the other person uses pace notes to sequence his/her attention for landmarks, and to provide instructions to their partner about the path to take at intersections.

Just as the human task of rally driving can be split between two people, the task of autonomous navigation can be split between two software agents: tactical (wall-following) and strategic (localization). At the lower level, the tactical agent is responsible for the steering and speed of the vehicle, whilst at the higher level, the strategic agent is responsible for receiving instructions from the operator and for passing hints down to the tactical agent. These hints are designed to influence the behaviour of the tactical software (i.e. turning and speed) at appropriate locations in the mine. Thus bridging the gap between hierarchical and reactive schools of navigation [2]. This paper describes the implementation of the tactical software onto a Load-Haul-Dump (LHD) truck — a 30 tonne underground mining vehicle.

2 Implementation

The task of the tactical software (wall-following) is to generate a steering demand that will drive the vehicle forward through the mine without hitting the walls. It does not require any knowledge of the location of the vehicle with respect to a global coordinate frame, it need only be aware of its local surroundings. From previous field trials [3], scanning lasers were found to be an ideal primary navigation sensor, providing 180° coverage up to a range of 30m. With this data, the wall-following algorithm needs to solve two problems. Firstly, it needs to construct a desired vehicle path from the free space observed in front the vehicle, and secondly, to use this path to estimate a steering demand.

2.1 Desired Vehicle Path

There are many ways to generate a desired vehicle path. Potential field methods have been described by robotics researchers for a number of years [4]. The general idea behind this method is to treat the vehicle as a particle that is attracted by a potential field radiating from its goal and repulsed by fields radiating from obstacles. A local path plan may then be formed by applying a force, based on the sum of the potential fields, to a general desired path whose end is fixed to the vehicle. Such schemes are normally iterative in nature and are hence amenable to real-time implementation. Potential field methods are very similar in principle to the *active contour* methods [5] which are commonly used in image processing and computer vision applications.

In this work, we have chosen active contours (often called snakes) to generate the desired vehicle path. A snake is a set of vertebra that define a line, or spline in the vehicles frame of reference: (s_x^i, s_y^i) for $i = 0$ to ns . An example, is shown in Figure 1. The total length of the snake is a function of the speed of the vehicle. The position of each vertebra is determined iteratively by minimising the energy of the snake. The snake has three energy terms:

$$E = E_e + E_i + E_p$$

where, the external energy, E_e manifests itself as a repulsive force between the vertebra of the snake and the wall. The wall is defined by the set of laser range points (r_x^j, r_y^j) for $j = 0$ to nr .

$$E_e = S_w \sum_{i=0}^{ns} \sum_{j=0}^{nr} 1/((s_x^i - r_x^j) - (s_y^i - r_y^j))$$

where, S_w is a snake wall constant. The internal energy, E_i manifests itself as a function of the curvature of the snake, where the bending energy is a function of the minimum turning circle of the vehicle. A potential energy, E_p is added to account for hints supplied from the strategic agent (i.e. positional bias).

There are two advantages with this approach. Firstly, the position of the snake is iterative, i.e. the starting position of the snake is based upon its last position.

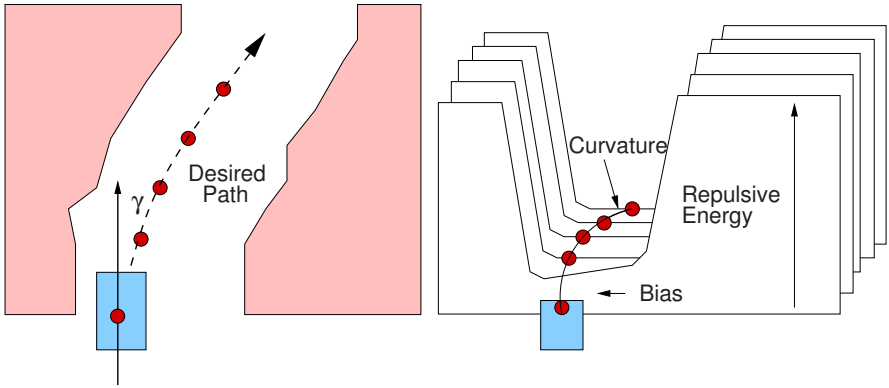


Fig. 1. Wall-following with snakes.

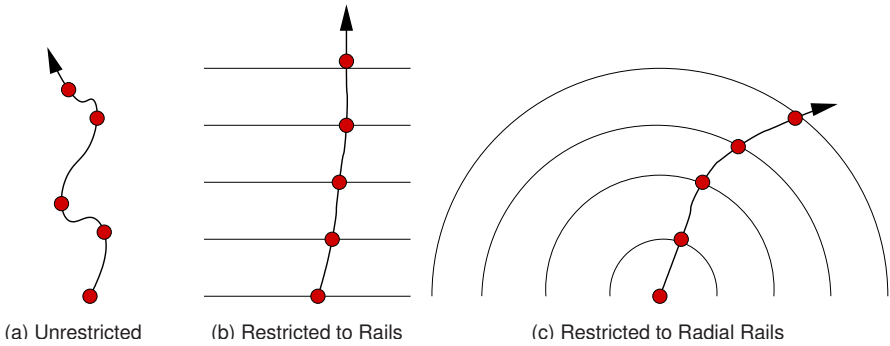


Fig. 2. Different forms of snake.

This historical knowledge is appropriate to situations where the scene in front of the vehicle does not change significantly between scans. Secondly, the tail of the snake is fixed to the front of the vehicle. This means that unlike navigation techniques that rely upon an external map, the desired vehicle path is generated relative to the current position of the vehicle. This technique has an added side effect, in that the energy of the snake is related in some manner to the difficulty of the path (i.e. either high curvature, or high external energy where the walls are very close). In either case, this energy can be used by the software to slow the vehicle down.

2.2 Snakes, Rails and Bands

With conventional snakes [5], each vertebra of the snake is free to move in two dimensions (Figure 2a). To maintain a snake of constant length, it is necessary to correctly balance the binding and bending energy of the snake, otherwise the snake can fold back on itself. To prevent this from happening, a new technique was developed where the vertebra of the snake are fixed to linear rails and are only free to move in one dimension (Figure 2b). Since this halves the degrees-of-freedom of

each vertebra, the solution is significantly faster, but the snake is no longer able to turn corners.

Corners can be addressed by having radial rails, where each vertebra of the snake is free to move at a specified radius from the front of the vehicle (Figure 2c). In this approach, rather than each vertebra being repelled by the wall, it need only be repelled by the wall that intersects the rail. The location of the wall can be established by subdividing each rail into a number of bands. A band is defined as a section of rail that is broken and isolated from the rest of the rail.

An example of a hypothetical Y-intersection is shown in Figure 3 where on the left, the range is plotted against angle. In this example we can see the left and right opening, where the lines that represent the radial rails in polar coordinates at distance of 1 to 4m, have been split into two bands. In the right hand figure, the bands have been redrawn in Cartesian coordinates. Each band represents a space in which the snake is free to move. The edge of each band can be defined by the set of points, (l_x^i, l_y^i) for the left wall, and (u_x^i, u_y^i) for the right, where $i = 0$ to ns . The external energy can be reformulated to be a function of the distance from each vertebra to the edge of the band.

$$E_e = S_w \sum_{i=0}^{ns} \frac{1}{\sqrt{(s_x^i - u_x^j)^2 - (s_y^i - u_y^j)^2}} + \frac{1}{\sqrt{(s_x^i - l_x^j)^2 - (s_y^i - l_y^j)^2}}$$

Since this energy term is dimensionless it is possible to use angular distances, i.e. the angle between the snake vertebra and the wall. In this notation, the position of the snake is defined by an angle s^i . Likewise, for the left and right wall, l^i and u^i . This simplifies the external energy equation.

$$E_e = S_w \sum_{i=0}^{ns} \frac{1}{(u^i - s^i)} + \frac{1}{(s^i - l^i)}$$

In this format it is possible to add the potential energy,

$$E = E_i + S_w \sum_{i=0}^{ns} \frac{S_b}{(u^i - s^i)} + \frac{1 - S_b}{(s^i - l^i)}$$

where, S_b is the snake bias which can range from 0.0 to 1.0. For example, when $S_b = 0.5$, the energy is balanced from left to right. If $S_b = 0.0$ the right hand term is dropped, and the snake is no longer repelled by the right wall. If $S_b = 1.0$, the left hand term is dropped, and the snake is no longer repelled by the left wall. Since, the calculations can now be performed in the coordinate frame of the laser (polar co-ordinates) there is a substantial increase in computational speed (several orders of magnitude) which has made the technique feasible at the output rate of the laser (i.e. 25Hz from Sick PLS).

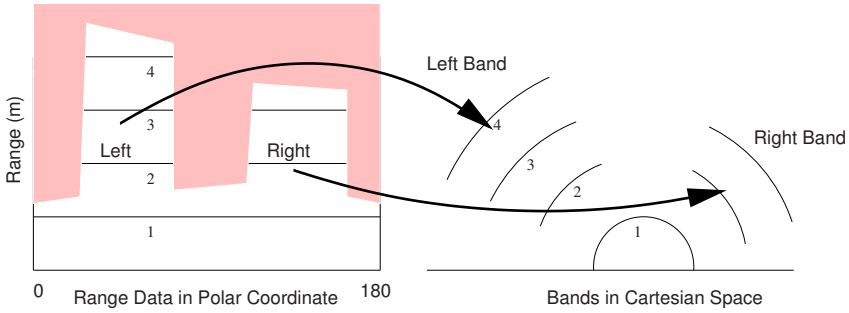


Fig. 3. Conversion from range data to rails.

2.3 Example

The process of generating a snake is demonstrated in a sequence of snapshots in Figure 4, where the vehicle (the LHD articulated truck) is positioned in a straight section of tunnel which widens out to a maximum distance of 25m. Figure 4a, shows a plot of the range data from a laser that is positioned over the wheel arch. This is 3m back from the front of the vehicle. Figure 4b shows the transformation of the range data to the front of the bucket. Figure 4c shows the radial rails that are fitted into the free space at 1m intervals. Figure 4d show the cropping of the radial rails away from the walls. This creates a buffer zone, in which the centre of the vehicle cannot drive. In effect, we consider the vehicle to be a point, and shrink the walls of the tunnel by half the vehicle width. In Figure 4e, the snake vertebra are placed at the midpoint of each rail. In Figure 4f, the vertebra are iteratively moved along the rails to minimize the snake energy, i.e. reduce curvature and repel walls. This repulsion is simply calculated from the inverse square distance to the end of the band.

The position of the snake can be shifted by adding a potential energy bias to the snake. Thus, the snake can be pushed to the left (Figure 4g) or right (Figure 4h). This bias can be used to improve the vehicles driving behaviour. For example, if we wish to make a sharp turn to the right it may be useful to hug the left wall as we approach the corner. It can also be used to account for unusual tunnel configurations or to avoid low lying infrastructure.

2.4 Turning

At various intersection in the mine, it will be necessary to make a decision to turn the vehicle. This can be done with hints supplied from the strategic agent. For example, when the tactical agent receives a hint to turn left, the vehicle does not turn left immediately, it just directs the vehicle to a left opening whenever the opportunity arises. To control turning, the snake can be excluded from regions of space. In the current notation, it is possible to ensure that the vehicle takes the appropriate branch by selecting the appropriate band. This is shown at a intersection in Figures 5a. where there are two potential branches, and thus two bands. The position of the

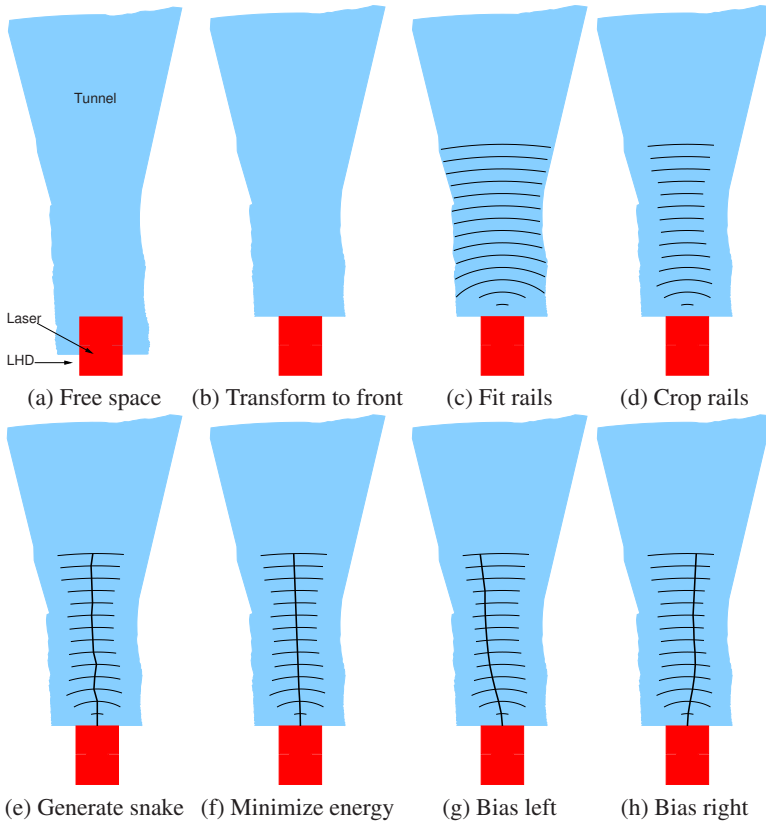


Fig. 4. Steps to generate snake in straight tunnel.

snake and thus the desired path can be controlled by selecting the appropriate band. In this case, the left and right hand bands are selected.

One of the advantages of this representation of free space is that it is very insensitive to small changes in topology, i.e. a branch can only be selected if there is enough room for the vehicle to drive through it. In practical terms, what this means is that it is possible to use a turning hint well in advance of the actual intersection. Thus making it possible to navigate with weak localization such as odometry.

As mentioned in the previous section, the bands are cropped to prevent the vehicle from driving too close to the wall. However, it is also possible to crop the bands down to the width of the vehicle. This can be used to keep the vehicle against a wall. The cropping does not occur in the first four rails, to allow the snake some freedom close to the vehicle. An example of this technique at a Y-junction is shown in Figure 5c, where it is possible to force the snake to hug the left or right wall.

Now that the snake is able to generate a path, it is necessary to estimate the articulation angle that will get the vehicle to follow this path. There are many ways that this can be done. One method is to match the curvature of the vehicle to the

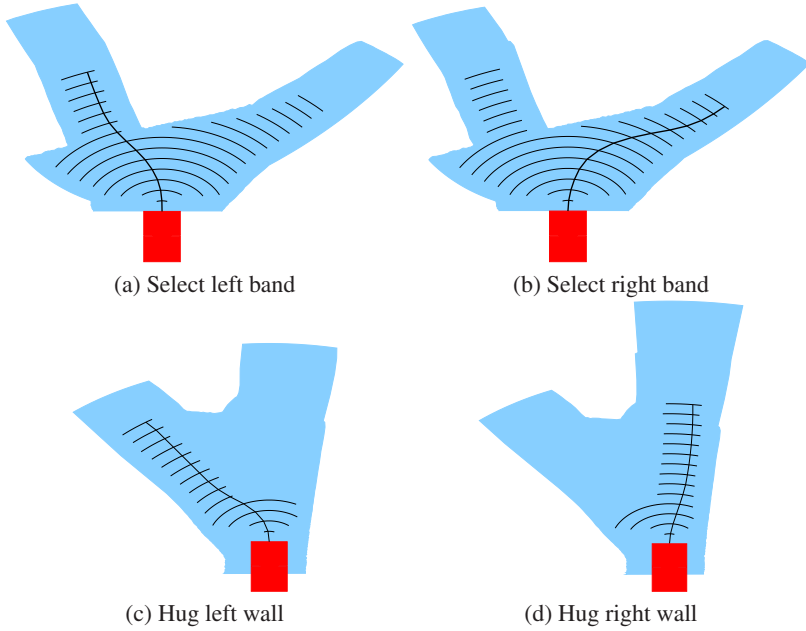


Fig. 5. Influencing snake at intersection.

curvature of the desired path. A kinematic model of the LHD can be used to estimate a change to the articulation angle that will move the LHD onto this circle at some time in the future. Unfortunately, in practice this technique did not work very well. A much simpler approach is to use a heading error, i.e. the angle to one of the snake vertebra. The choice of vertebra will depend upon how far we need to look ahead. This simple approach proved to be very successful.

3 Results

For initial testing the LHD was driven in co-pilot mode, i.e. where turning hints and speed are directed from a remote user interface. In these trials, the various control parameters were adjusted, (bending energy, snake vertebra spacing etc) until stable motion had been achieved, at which point the strategic software and the associated nodal-map were installed.

Under fully-autonomous control the LHD was able to navigate more than 300m of our test tunnel, which included two 90° corners and a sweeping loop with a radius of 8m, at speeds of 18km/h with clearances of less than 1m. The vehicle operated for over an hour at a time without any human intervention. Under most conditions the LHD was driven autonomously at the same speed as a human operator. Subjectively, the LHD under autonomous control takes a better line, and reacts faster than a human operator. The only weakness occurs at sharp intersections where the autonomous LHD must travel slowly to “see” around the corner. A human driver can drive more

aggressively around a blind corner because they can remember the profile of the tunnel from a previous run.

One of the concerns raised about such a simple control strategy was whether it could be ported to a different machine. Fortunately, we were able to test the system on a substantially larger machine (60 tonnes) with different hydraulics. Once again the automation system was able to match the performance of a human operator in most situations.

4 Discussion

One of the questions raised in this research concerned the apparent success of the simple heading error over a more sophisticated model based approach. The answer lies in the definition of the desired vehicle path. Relative navigation systems do not use a global path but a relative path. The issue of global vs relative paths is a question of absolute vs relative navigation. This difference is highlighted in the following examples where there is an intention to turn left:

Absolute Navigation: In Figure 6a the global path lies in the middle of the tunnel. This path was determined *a priori* from a global map. Given that the vehicle knows its location in the mine, then from the map, it knows what to expect around the corner and can plan accordingly.

Relative Navigation: In Figure 6b the relative path heads for the T-junction wall. This path was determined *a posteriori* from the local environment. Since the system does not have any knowledge of what is behind the laser shadow it can only generate a path from the free space in front of the vehicle.

The purpose of a model-based steering demand is to fit the actual path of the vehicle to the desired path. This approach is best suited to absolute navigation system because the global path **is the desired path**. However for relative navigation, the relative path **may not be the desired path**. This is certainly the case in the previous example, where the relative path collides with the far wall !

This situation can be contrasted with the heading-based steering demand (follow-the-carrot [6]) where there is little expectation that the vehicle will ever follow the demanded path. This type of steering demand can be likened to a string that pulls a wooden toy, where the vehicle responds to demands at the tail of the string. In this situation, as the vehicle is towed around, the vehicle does not follow the path of the entire string. One of the reasons that this approach works is that it involves positive feedback. In the example given, when the relative navigation system sees the T-junction and decides to turn left, then the relative path will head slightly towards the left. This will generate a heading error to the left. As the vehicle approaches the T-junction, the vehicle can see more of the left corridor, and the relative path will extend further down it. This will generate a increased heading demand, and so on.

To reiterate, in a reactive navigation system, it is not important that the vehicle follow some path, but that a sequence of heading commands ensures that the vehicle does not hit the wall. This type of behaviour has all the hallmarks of *visual servoing*

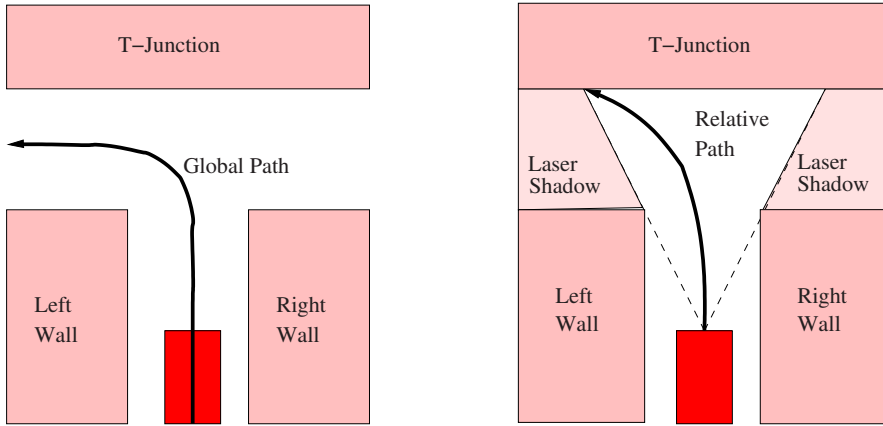


Fig. 6. Global paths (a) and relative path (b).

(i.e. a system that reacts to changes in the visual environment) except the visual environment is the laser range image. In principal, visual servo systems can navigate by simply turning away from obstacles. For example, where there is an expectation that we will collide with the a wall at some time in the future. To avoid this collision we turn away. If at a later time, after the vehicle has turned we are still heading for the wall, we should turn a little more. The success of this feedback depends upon the speed of acquisition and execution. In effect, we are treating navigation as a control problem.

The problem with this technique is knowing how much to steer left or right. This is called the gain. In an attempt to understand the relationship between the snake and reactive navigation, it is possible to select different vertebra along the snake. In practice, if the vertebra chosen is less than 4m away from the vehicle, then at normal driving speeds the vehicle did not turn quickly enough. In terms of reactive navigation, it means that the gain was too low. Although in theory, a reactive navigation system should be able to compensate for this, there is a subtle interaction between the demands of the snake and the operational layer. In this layer, there is a steering predictor that is used to handle steering delays. The effect of this predictor is that the vehicle will actuate faster given a higher steering demand. Small corrections to steering that are within the dead-band of the steering controller are ignored.

5 Summary

To summarize:

- The task of navigation can be split into tactical and strategic agents.
- Reactive navigation is suitable for the the underground environment.
- Active contours (snakes) can be used to generate a desired path.
- Radial rails and bands can be used to constrain the motion of snake vertebra.

- Cropping the bands has the effect of reducing the vehicle to a point.
- Simple heading error is sufficient for vehicle control.
- Only very simple turning hints are required to navigate.

The wall-following algorithm described in this paper has proved to be extremely robust under both experimental and operational conditions. It can handle different articulated vehicles and various tunnel geometries. The parameters that control the snake, once tuned, can be left unchanged. Since most of the calculations are done in radial-space (the coordinate space of the sensor) the algorithm is very efficient. In fact, it is quite possible that it could be ported to small micro-controllers.

Acknowledgments

The authors would like to thank the rest of the project team: Peter Corke, Graeme Winstanley, Stuart Wolfe, Andrew Castleden, Leslie Overs and Reece McCasker who designed and built the hardware for our LHD. Thanks also to Pavan Sikka and Robin Kirkham who wrote key software modules, and to Jock Cunningham who managed the project. The experimental part of this work was conducted and partially funded through the CRC for Mining Technology and Equipment (CMTE). Funding from industry was provided through the Australian Mineral Industries Research Association (AMIRA) under project P517 by MIM Holdings Ltd, Normandy Mining Ltd, WMC Resources, North Limited, Pasminco Ltd, Western Metals, Automotive Industrial Mining Supplies and Caterpillar Elphinstone Pty Ltd. Research results from other CSIRO projects were also applied.

References

1. J. Roberts, E. Duff, P. Corke, P. Sikka, G. Winstanley, and J. Cunningham, "Autonomous control of underground mining vehicles using reactive navigation," in *Proceedings of IEEE Int. Conf. on Robotics and Automation*, (San Francisco, USA), pp. 3790–3795, 2000.
2. R. C. Arkin, "Towards the Unification of Navigational Planning and Reactive Control," Working Notes of the AAAI spring Symposium on Robot Navigation, Stanford University, March 1989.
3. S. Scheding, E. Nebot, M. Stevens, H. Durrant-Whyte, J. Roberts, P. Corke, J. Cunningham, and B. Cook, "Experiments in autonomous underground guidance," in *Proc. IEEE Conf. Robotics and Automation*, (Albuquerque, NM), pp. 1898–1903, 1997.
4. O. Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
5. A. Blake and M. Isard, *Active Contours*. Springer-Verlag, 1998.
6. M. Hebert, C. Thorpe, and A. T. Stentz, eds., *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*. KluwerAcademic Publishers, 1997.

Landmark-Based Nonholonomic Visual Homing

Kane Usher^{1,2}, Peter Corke¹, and Peter Ridley²

¹ CSIRO Manufacturing and Infrastructure Technology
P.O. Box 883, Kenmore, 4069, Queensland, Australia
{firstname}.{surname}@csiro.au
<http://www.cat.csiro.au/cmst/automation>

² School of Mechanical, Manufacturing and Medical Engineering
Queensland University of Technology
Brisbane, 4001, Queensland, Australia

Abstract. In this paper, we present a method which allows pose stabilization of a car-like vehicle to a learnt location based on feature bearing angle and range discrepancies between the vehicle's current view of the environment, and that at the learnt location. We then extend the technique to include obstacle avoidance. Simulations and experimental results using our outdoor mobile platform are presented.

1 Introduction

In order to perform useful tasks, a mobile robot requires the ability to servo to particular poses in the environment. For the nonholonomic, car-like vehicle used in these experiments, Brockett [2], showed that there is no smooth, continuous feedback control law which can locally stabilise such systems.

Insects in general display amazing navigation abilities, traversing distances far surpassing the best of our mobile robots on a relative scale. Evolution has provided insects with many 'shortcuts' enabling the achievement of relatively complex tasks with a minimum of resources in terms of processing power and sensors [12]. The high ground temperatures encountered by the desert ant, *cataglyphis bicolor*, eliminates pheromones as a potential navigation aid, as is used by ants in cooler climates [6]. The desert ant navigates using a combination of path integration and visual homing.

Visual homing is the process of matching an agent's current view of a location in a distinctive locale to a (pre-stored) view at some target position. Discrepancies between the two views are used to generate a command that drives the agent closer to the target position. The process enables the agent to 'find' positions in distinctive locales. When applied to nonholonomic mobile robots, the constraints of Brockett's theorem prevent the insect-based strategies from completely resolving the pose stabilization problem; they enable servoing to a position but cannot guarantee a particular orientation (see e.g. [6,12]).

In the control community, the problem of stabilising a mobile robot to a specific pose has generally been approached from two directions; the open-loop and closed-loop strategies. Open-loop strategies seek to find a bounded sequence of control inputs, driving the vehicle from an initial pose to some arbitrary goal pose, usually working in conjunction with a motion planner (e.g. [7,9]). Finding this sequence of

control inputs is difficult for the nonholonomic case and, in the event of disturbances, a new plan has to be formulated. The closed-loop strategies consist of designing a feedback loop using proprioceptive and exteroceptive sensors to provide estimates of the vehicle's state. Feedback control systems are generally more robust to uncertainty and disturbances when compared to their open-loop counterparts. All real mobile robots and sensors are subject to noise and uncertainty — feedback control would thus seem essential. Feedback pose stabilization for under-actuated nonholonomic systems is addressed with either discontinuous control techniques (see e.g. [1]), time-varying control (see e.g. [10]), or combinations thereof.

Much of the literature does not address what has been found in this study to be a significant limitation of many control algorithms — input saturation. Further to this, many of the algorithms in the literature cannot cope with systems which have significant velocity and steering loop dynamics. In fact, there are few instances of actual implementations of pose control to a car-like vehicle (an exception is Fraichard and Garnier [4] who use fuzzy control techniques applied to a small electric car).

In this paper we develop a discontinuous feedback control strategy, showing that it can be used in combination with a visual sensor in a system which has significant velocity and steering loop dynamics. The remainder of this paper is arranged as follows: Section 2 details the switching control technique and our version of visual homing; Section 3 describes our experimental system and briefly outlines some preliminary results; Section 4 outlines some preliminary results in pose stabilization with obstacle avoidance; and Section 5 concludes the paper.

2 Control Strategy

In this section, the switching control strategy presented by Lee et al. [8] is developed and an instability is corrected. We then show how a derivative of the insect-inspired Average Landmark Vector model of navigation presented by Lambrinos et al. [6] can be used to provide the required quantities to the switching controller.

2.1 Kinematics

Car-like vehicle kinematics are usually represented using the *bicycle* model. Referring to Fig. 1, the kinematics of our experimental vehicle are:

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= v \frac{\tan \phi}{L}\end{aligned}\tag{1}$$

where v is the vehicle's forward velocity (measured at the centre of the rear axle), L is the vehicle's length, ϕ is the steering angle, and the point (x, y) refers to the centre of the rear axle.

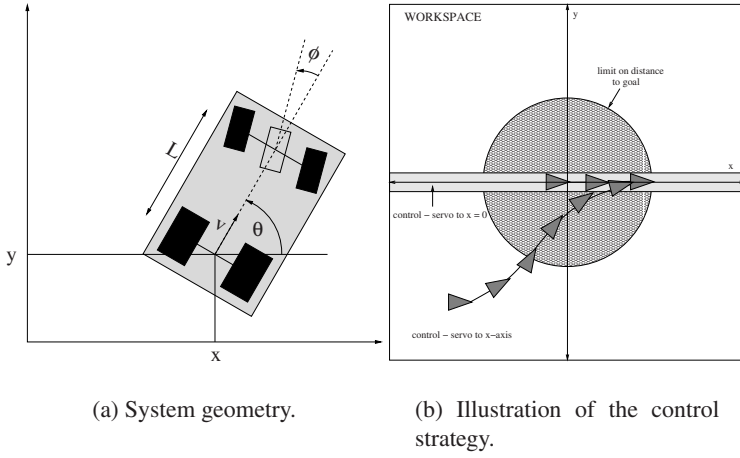


Fig. 1. The car-like vehicle system and the stages of control.

2.2 Control Law Development

Lee et al. [8] use discontinuous control to sidestep Brockett's theorem, breaking the stabilization task into two stages. Without loss of generality, consider the goal pose to be $(x, y, \theta) = (0, 0, 0)$. The initial stage involves minimising y and θ ; i.e. the vehicle converges to the x -axis with an orientation of 0. The second stage then moves the vehicle along the x -axis to the desired point. In practice, we have found the need for an additional stage which helps to minimise the space required to servo to a pose. A discrete event supervisor decides which stage of control is in effect.

Control Law — Converge to x -axis Using a suitably chosen Lyapunov function, ($V = k_1 \frac{1}{2} y^2 + \frac{1}{2} \theta^2$) Lee et al. [8] showed that y and θ converge to zero with the following control demands:

$$\phi = \arctan \left[-\frac{L}{v} \left(k_2 \theta + k_1 v \frac{\sin(\theta)}{\theta} y \right) \right] \quad (2)$$

and

$$v = \begin{cases} k_3 & \text{if } \cos \varphi_{\text{initial}} \geq 0 \\ -k_3 & \text{otherwise} \end{cases} \quad (3)$$

where φ_{initial} is the initial orientation of the goal relative to the vehicle longitudinal axis and k_3 is the velocity magnitude. However, we have found an instability in this controller due to saturation of the steering angle (in our case $\phi_{\text{max}} = \pm 30^\circ$). The velocity, v , is constant in magnitude (due to Equation 3). Hence, it is the angular rate of the vehicle, $\dot{\theta}$, which saturates:

$$\Phi(\phi) = \frac{v \tan \phi}{L} \quad (4)$$

which in turn leads to instability when the vehicle is far from the x-axis. By adapting the gain, k_1 , instability of the controller is avoided:

$$k_1 < \frac{\Phi(\phi_{max})\theta}{y \sin \theta} \quad (5)$$

Control Law — Servo to $x = 0$ On reaching the y-axis and $\theta = 0$, the following control law on the vehicle's velocity is invoked:

$$v = -k_3x \quad (6)$$

bringing the vehicle to the desired pose of $(x, y, \theta) = (0, 0, 0)$. In the original work of Lee et al., the steering angle was set to zero in this stage of control. We have found that in practice, due to the imperfections of 'real' systems, it is necessary to control the steering angle — we again use equation 2. An analysis of the Lyapunov function for this stage of control shows that convergence is still guaranteed.

Control Law — Turn Stage Although the velocity in the 'converge to x-axis' stage of control is chosen according to the vehicle's initial orientation with respect to 'home', this does not always guarantee that in converging to the x-axis, the vehicle will make efficient use of the workspace. If using landmarks to position the vehicle, a wide area of operation may result in landmarks disappearing from the vehicle's perception horizon. Thus, we have added a further stage to the controller, which turns the vehicle towards the goal. Again, the controller is based upon a Lyapunov formulation, this time with the aim of minimising the goal's orientation relative to the vehicle. Once a threshold is reached, the 'converge to x-axis' stage is invoked.

Discussion In combination, these control laws stabilise the vehicle to the desired pose from any initial condition. Again in practice, the imperfections of 'real' systems requires that switching to the 'servo to $x = 0$ ' stage of control occurs when the y and θ errors drop below pre-specified thresholds.

Many of the pose stabilization algorithms with which we experimented failed in the face of input saturation on the steering angle and could not cope with actuator dynamics in the velocity and steering loops. We believe that this method is successful because it does not attempt to resolve pose in one step — in servoing to the x-axis first, the dynamic response of the vehicle becomes less important, as does steering input saturation. On reaching the x-axis with the correct orientation, the dynamics are less important and in general, smaller steering inputs are required and saturation is avoided.

The control technique presented in this section assumes that the pose of the vehicle is known. In the next section, we detail a landmark-based vision technique which yields pose estimates in an extremely simple, yet surprisingly robust, manner using an omnidirectional camera and a compass.

2.3 Ant Navigation

An elegant, correspondence free, homing method developed from hypotheses on ant visual homing is the Average Landmark Vector model [6]. An ALV for any particular position is found by summing unit vectors towards all currently visible landmarks and dividing by the number of landmarks. By matching the current ALV with a pre-stored ALV of the target location, a homing vector can be formed which drives the robot towards the target location [6]. In order to consistently add the vectors in the ALV model, an absolute reference direction is required, and, unless apparent size information is incorporated, a minimum of three landmarks is needed.

We have found that by using *range* vectors to the individual landmarks, rather than *unit* vectors, the distance and angle to the goal are yielded directly. In addition, the minimum required landmarks is reduced to one. An example of the IALV method is shown in Fig. 2. In essence, the IALV method is equivalent to finding a position relative to the centroid of the landmarks in the workspace.

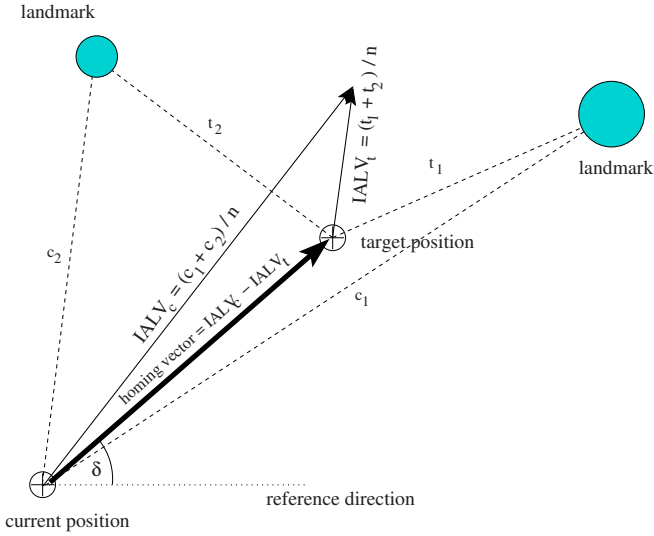


Fig. 2. Illustration of the the IALV method for $n = 2$ landmarks in a workspace.

As with the ALV method, the IALV method is sensor-based. Landmark bearings are readily ascertained with an omnidirectional camera. If a flat-earth assumption is made, range information can be derived from such an image through the geometry of the camera/mirror optics. One of the advantages of the ALV, and hence the IALV method, is that knowledge of a target location is contained within a single quantity. This reduces the need for complex map-like representations of the environment and is well suited for a topological navigation method, (see e.g. [5]). Additionally, landmarks need not be unique, and the need for landmark correspondence is also bypassed. Of course, like all sensor-based techniques, this method has a finite catch-

ment area, limited by the omnidirectional sensor's range and, in addition, has the potential to suffer from perceptual aliasing, or in a similar sense, the local minima problem.

The homing vector provided by the IALV method can be used to drive the agent towards home but does not provide a means of guaranteeing a final orientation. However, the quantities derived from the IALV can easily be converted to the states required by the switching controller presented in Section 2.

2.4 Simulations

The model of the vehicle consists of the kinematic equations (Equation 1) and dynamic models of the steering and velocity loops (identified as first and second-order respectively). Also included in the model are input saturation and rate limiting. A simulation of the omnidirectional sensor (with random noise) provides the IALV, from which the required states are derived and fed to the switching controller. Fig. 3 shows the path generated and control inputs for a starting pose of $(x, y, \theta) = (0, 3, 0)$ and a goal pose of $(0, 0, 0)$. Gains were set to $(k_1, k_2, k_3) = (0.35, 0.1, 0.1)$ for these simulations, based on a root locus analysis of the linearised system. The method works for all starting and goal poses.

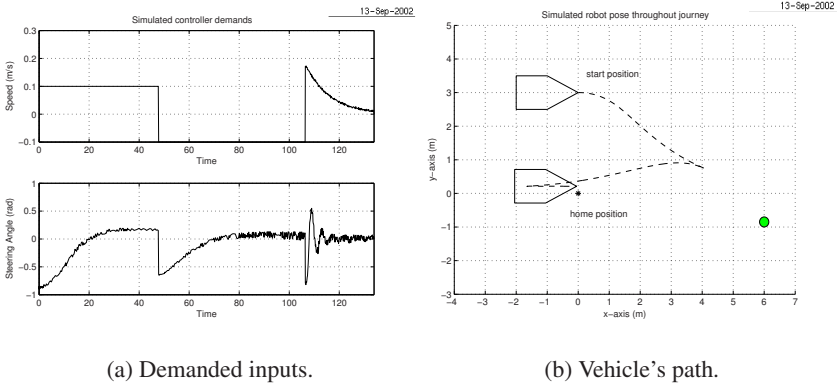


Fig. 3. Results of the simulation for a starting pose of $(x, y, \theta) = (0, 3, 0)$.

The method has been extensively tested in simulation and found to be very robust to input saturation and noise. The next section presents some experimental results which validate our simulations.

3 Experiments

Experimental validation of this visual servoing technique was conducted on our outdoor research testbed. In these experiments, artificial landmarks in the form

of red traffic cones (witches hats) were used. The following sections give a brief overview of the vehicle and an outline of the image processing used to extract the landmarks.

3.1 Robotic Testbed

The experimental platform is a Toro ride-on mower which has been retro-fitted with actuators, sensors and a computer system, enabling control over the vehicle's operations. All computing occurs on-board. The vehicle is fitted with an array of sensors including odometry, GPS, a magnetometer, a laser range-finder (for collision avoidance only) and an omnidirectional camera (see Fig. 4). For these experiments, the primary sensor used is the omnidirectional camera with the magnetometer providing an absolute reference direction.



Fig. 4. The experimental platform.

3.2 Image Processing

In testing our controller, we use artificial landmarks in the form of bright red road cones (also known as witches hats). We use colour segmentation to find the landmarks, estimate landmark range based upon a geometric model of the camera / mirror system and a flat-Earth assumption, track the landmarks using a vehicle/landmark relative motion model, and improve the quality of the measurements using complementary filters which combine vehicle odometry with the vision and compass data. For a complete description of the sensing system see [11].

3.3 Results and Discussion

The testing arena used for these experiments is a large shed. The vehicle was placed in the middle of the shed defining the goal pose $(x, y, \theta) = (0, 0, 0)$. A ‘landmark’ was then placed at $(x, y) = (5.85, -1)$, and the target IALV was found and stored.

The vehicle was then manually driven to $(x, y, \theta) = (0, 3, 0)$, and the control system activated. Fig. 5 depicts the results of the experiment, showing the vehicle servoing to the goal pose based on vision and compass data alone. The similarity to the simulation results plotted in Fig. 3 is quite clear, although the experimental system did take longer to stabilise to the desired pose. This could be due to the quite coarse on-board velocity measurements and the noisy pose estimates.

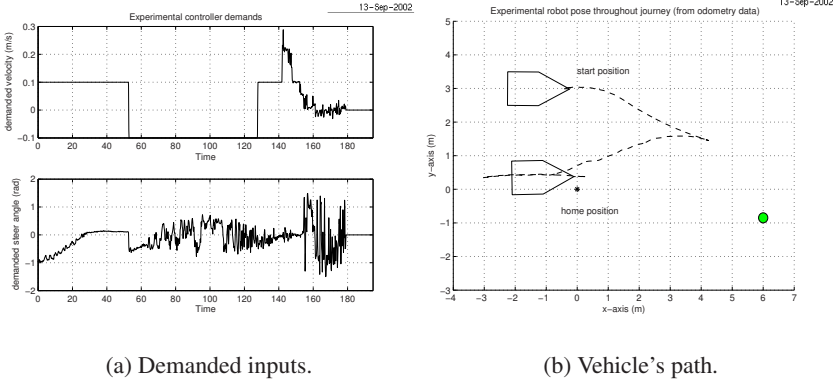


Fig. 5. Experimental results for a starting pose of $(x, y, \theta) = (0, 3, 0)$.

4 Obstacle Avoidance

In stabilising to a pose, it has so far been assumed that the local workspace is obstacle free. However, this will not always be the case and a means of dealing with obstacles is required. Of course, in using a purely feedback type strategy, no guarantee can be provided that the goal will always be reached, as there will always be pathological obstacle configurations which could be devised. Our intention here is to provide a means of stabilising to a pose and avoiding obstacles in *most* obstacle configurations, assuming that in the pathological cases, a higher level of control could be invoked.

Our approach is similar to that used by Fox et al. [3]. At each time step, the current demands to the vehicle are used to predict the trajectory of the vehicle, given knowledge of the kinematics and dynamics of the vehicle. In addition, an obstacle motion model is used to predict the relative position of each of the currently detected obstacles along the trajectory. If the trajectory passes too close to an obstacle, then it is flagged as unsuitable. At this stage, a loop is entered in which trajectories resulting from every available steering angle (quantized at 5° intervals) at the current velocity are evaluated. From the suitable evaluated trajectories, the one with a steering command closest to the original demand is selected. If none is available, then the vehicle's velocity command is iteratively reduced and the trajectories reevaluated. If

suitable commands are still not found, the control level is notified and a temporary, intermediate target is selected in a direction in reverse to the current travel direction. The vehicle then ‘homes’ to this temporary target point, re-targeting if necessary. On reaching the temporary target point, the original goal pose is re-acquired.

Figure 6 illustrates a simulation of pose stabilization with obstacle avoidance. Note that the vehicle initially detects that it cannot move forward and reassigns the target to the rear of the vehicle. Once this target is acquired the original target is re-acquired.

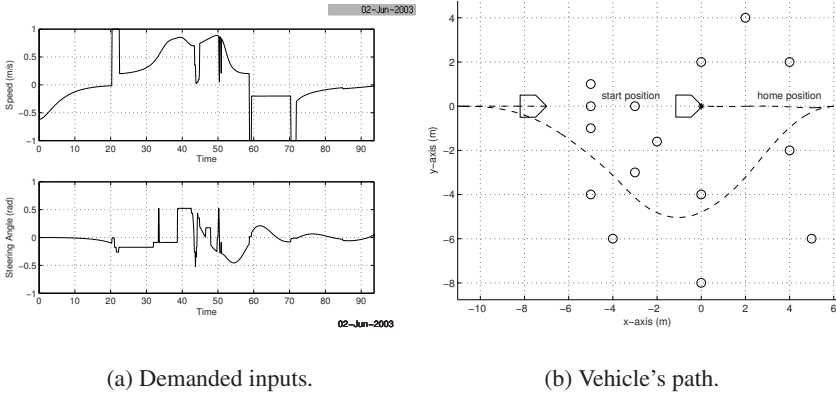


Fig. 6. Simulation results for a starting pose of $(x, y, \theta) = (-7, 0, 0)$ with obstacle avoidance.

5 Conclusion

This paper has described a method of stabilising a car-like vehicle to a target pose based on the discrepancies between a target view of the landmarks in a workspace and the vehicle's current view. The vehicle's view of the workspace is summarised by a single quantity, the Improved Average Landmark Vector. At each instant, the vehicle compares the current IALV with that at the target location which yields a vehicle pose estimate with respect to the target pose. At no stage is there a requirement for landmark correspondence; this represents a key advantage over many other homing methods. Furthermore, the method uses polar representations of the landmarks, and thus, in using an omnidirectional camera, the need for processor intensive image unwarping is circumvented.

We have presented simulations and real experimental results showing the validity of our approach even in the face of input saturation and noise. In addition, we have presented preliminary results in pose stabilization with obstacle avoidance. Future work includes further experimental validation of the method, along with a deeper understanding of the impact of dynamic elements in the control loops. Finally,

this technique is ideally suited to topological navigation, and this too represents a direction for future research.

Acknowledgements

The authors thank the automation team for their invaluable assistance. In particular, the support of the CMIT Tractor Team — Peter Hynes, Stuart Wolfe, Stephen Brosnan, Graeme Winstanley, Pavan Sikka, Elliot Duff, Jonathan Roberts, Les Overs, Craig Worthington and Steven Hogan — is gratefully acknowledged. Jonathon O'Brien at UNSW is gratefully thanked for the loan of the Toro ride-on mower.

References

1. Michele Aicardi, Giuseppe Casalino, Antonio Bicchi, and Aldo Balestrino. Closed loop steering of unicycle-like vehicles via Lyapunov techniques. *IEEE Robotics and Automation Magazine*, pages 27–35, March 1995.
2. R.W. Brockett. Asymptotic stability and feedback stabilization. In R. W. Brockett, R. S. Millman, and H. J. Sussman, editors, *Differential Geometric Control Theory*, pages 181–191. Birkhauser, Boston, USA, 1983.
3. D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, v(1), 1997.
4. Th. Fraichard and Ph Garnier. Fuzzy control to drive car-like vehicles. *Robotics and Autonomous Systems*, 34:1–22, 2001.
5. Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.
6. Dimitrios Lambrinos, Ralf Möller, Thomas Labhart, Rolf Pfeifer, and Rudiger Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30:39–64, 2000.
7. Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic, 1991.
8. Sungon Lee, Manchul Kim, Youngil Youm, and Wankyung Chung. Control of a car-like mobile robot for parking problem. In *International Conference on Robotics and Automation*, pages 1–6, Detroit, Michigan, 1999. IEEE.
9. Richard M. Murray and S. Shankar Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, May 1993.
10. C. Samson and K. Ait-Abderrahim. Feedback control of a non-holonomic wheeled cart in cartesian space. In *International Conference on Robotics and Automation*, pages 1136–1141, Sacramento, California, USA, April 1991. IEEE.
11. Kane Usher, Matthew Dunbabin, Peter Corke, and Peter Ridley. Sensing for visual homing. In *Proceedings of the 2003 Australasian Conference on Robotics and Automation*, Brisbane, Australia, December 2003. published via CDROM.
12. Keven Weber, Svetha Venkatesh, and Mandyam Srinivasan. Insect-inspired robotic homing. *Adaptive Behavior*, 7(1):65–97, 1999.

Recursive Probabilistic Velocity Obstacles for Reflective Navigation

Boris Kluge and Erwin Prassler

Research Institute for Applied Knowledge Processing
Helmholtzstr. 16, 89081 Ulm, Germany
{kluge, prassler}@faw.uni-ulm.de
<http://www.faw.uni-ulm.de>

Abstract. An approach to motion planning among moving obstacles is presented, whereby obstacles are modeled as intelligent decision-making agents. The decision-making processes of the obstacles are assumed to be similar to that of the mobile robot. A probabilistic extension to the velocity obstacle approach is used as a means for navigation as well as modeling uncertainty about the moving obstacles' decisions.

1 Introduction

For the task of navigating a mobile robot among moving obstacles, numerous approaches have been proposed previously. However, moving obstacles are most commonly assumed to be traveling without having any perception or motion goals (i.e. collision avoidance or goal positions) of their own. In the expanding domain of mobile service robots deployed in natural, everyday environments, this assumption does not hold, since humans (which are the moving obstacles in this context) do perceive the robot and its motion and adapt their own motion accordingly. Therefore, reflective navigation approaches which include reasoning about other agents' navigational decision processes become increasingly interesting.

In this paper an approach to reflective navigation is presented which extends the velocity obstacle navigation scheme to incorporate reasoning about other objects' perception and motion goals.

1.1 Related Work

Some recent approaches (see for example [3,5]) use a prediction of the future motion of the obstacles in order to yield more successful motion (with respect to travel time or collision avoidance). However, reflective navigation approaches are an extension of this concept, since they include further reasoning about perception and navigational processes of moving obstacles.

The velocity obstacle paradigm [2] is able to cope with obstacles moving on straight lines and has been extended [6] for the case of obstacles moving on arbitrary (but known) trajectories.

Modeling other agents' decision making similar to the own agent's decision making is used by the recursive agent modeling approach [4], where the own agent

bases its decisions not only on its models of other agents' decision making processes, but also on its models of the other agents' models of its own decision making, and so on (hence the label *recursive*).

1.2 Overview

This paper is organized as follows: The basic velocity obstacle approach is introduced in Section 2, and its probabilistic extension is presented in Section 3. Now being able to cope with uncertain obstacle velocities, Section 4 describes how to recursively apply the velocity obstacle scheme in order to create a reflective navigation behavior. An implementation of the approach and an experiment are given in Section 5. After discussing the presented work in Section 6, Section 7 concludes the paper.

2 Velocity Obstacle Approach

Let B_i and B_j be circular objects with centers c_i and c_j and radii r_i and r_j , moving with constant velocities $v_i = \dot{c}_i$ and $v_j = \dot{c}_j$. To decide if these two objects are on a collision course, it is sufficient to consider their current positions together with their relative velocity $v_{ij} = v_i - v_j$, see Fig. 1. Let

$$\hat{B}_{ij} = \{c_j + r \mid r \in \mathbb{R}^2, |r| \leq r_i + r_j\}, \quad (1)$$

$$\lambda_{ij}(v_{ij}) = \{c_i + \mu v_{ij} \mid \mu \geq 0\}. \quad (2)$$

Then B_i and B_j are on a collision course, if and only if $\hat{B}_{ij} \cap \lambda_{ij} \neq \emptyset$.

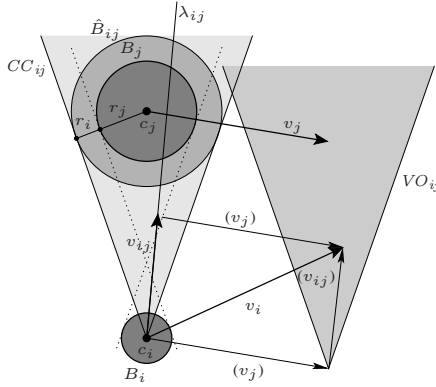


Fig. 1. Collision cone and velocity obstacle

Therefore we can define a set of colliding relative velocities, which is called the *collision cone* CC_{ij} , as

$$CC_{ij} = \{v_{ij} \mid \hat{B}_{ij} \cap \lambda_{ij}(v_{ij}) \neq \emptyset\}. \quad (3)$$

In order to be able to decide if an absolute velocity v_i of B_i leads to a collision with B_j , we define the *velocity obstacle* of B_j for B_i to be the set

$$VO_{ij} = \{v_i \mid (v_i - v_j) \in CC_{ij}\}, \quad (4)$$

which is, in other words,

$$VO_{ij} = v_j + CC_{ij}. \quad (5)$$

Now for B_i , any velocity $v_i \in VO_{ij}$ will lead to a collision with B_j , and any velocity $v_i \notin VO_{ij}$ for B_i will avoid any collision with B_j .

In general, object B_i is confronted with more than one other moving object. Let $\mathcal{B} = \{B_1, \dots, B_n\}$ the set of moving objects under consideration. The velocity obstacle of \mathcal{B} for B_i is defined as the set

$$VO_i = \cup_{j \neq i} VO_{ij}. \quad (6)$$

For any velocity $v_i \notin VO_i$, object B_i will not collide with any other object.

Finally, a simple navigation scheme based on velocity obstacles (VO) can be constructed as following. The moving and non-moving obstacles in the environment are continuously tracked, and the corresponding velocity obstacles are repeatedly computed. In each cycle, a velocity is chosen which avoids collisions and approaches a motion goal, for example a maximum velocity towards a goal position.

3 Probabilistic Velocity Obstacles

The velocity obstacle approach as presented in the preceeding section can be extended to deal with uncertainty in shape and velocity of the objects. This allows to reflect the limitations of real sensors and object tracking techniques.

3.1 Representing Uncertainty

Uncertainty in the exact shape of an object is reflected by uncertainty in the corresponding collision cone. Therefore, we define the *probabilistic collision cone* of object B_j relative to object B_i to be a function

$$PCC_{ij} : \mathbb{R}^2 \rightarrow [0, 1] \quad (7)$$

where $PCC_{ij}(v_{ij})$ is the probability of B_i to collide with B_j if B_i moves with velocity v_{ij} relative to B_j .

Similarly, we represent the *uncertain velocity* of object B_j by a probability density function

$$V_j : \mathbb{R}^2 \rightarrow \mathbb{R}_0^+. \quad (8)$$

With these two definition, we get the *probabilistic velocity obstacle* of object B_j relative to object B_i as a function

$$PVO_{ij} : \mathbb{R}^2 \rightarrow [0, 1] \quad (9)$$

which maps absolute velocities v_i of B_i to the according probability of colliding with B_j . It is

$$\begin{aligned} PVO_{ij}(v_i) &= \int_{\mathbb{R}^2} V_j(v) PCC_{ij}(v_i - v) d^2v \\ &= (V_j * PCC_{ij})(v_i) \end{aligned} \quad (10)$$

where $*$ denotes the convolution of two function.

3.2 Probabilistic Velocity Obstacle

The probability of B_i colliding with any other obstacle when traveling with velocity v_i is the probability of not avoiding collisions with each other moving obstacle. Therefore we may define the *probabilistic velocity obstacle* for B_i as the function

$$PVO_i = 1 - \prod_{j \neq i} (1 - PVO_{ij}). \quad (11)$$

3.3 Navigating with Probabilistic VO

In the deterministic case, navigating is rather easy since we consider only collision free velocities and can choose a velocity which is optimal for reaching the goal. But now, we are confronted with two objectives: reaching a goal and minimizing the probability of a collision.

Let $U_i : \mathbb{R}^2 \rightarrow [0, 1]$ a function representing the utility of velocities v_i for the motion goal of B_i . However, the full utility of a velocity v_i is only attained if (a) v_i is dynamically reachable, and (b) v_i is collision free. Therefore we define the relative utility function

$$RU_i = U_i \cdot D_i \cdot (1 - PVO_i), \quad (12)$$

where $D_i : \mathbb{R}^2 \rightarrow [0, 1]$ describes the reachability of a new velocity.

Now a simple navigation scheme for object B_i based on probabilistic velocity obstacles (PVO) is obtained by repeatedly choosing a velocity v_i which maximizes the relative utility RU_i .

4 Recursive Probabilistic VO

In contrast to traditional approaches, recursive modeling for mobile robot navigation as presented in this paper presumes moving obstacles to deploy navigational decision

making processes similar to the approach used by the robot. This means any object B_j is assumed to take actions maximizing its relative utility function RU_j . Therefore, in order to predict the action of obstacle B_j , we need to know its current utility function U_j , dynamic capabilities D_j , and velocity obstacle PVO_j .

The utility of velocities can be inferred by recognition of the current motion goal of the moving obstacle. For example, Bennewitz et al. [1] learn and recognize typical motion patterns of humans. If no global motion goal is available through recognition, one can still assume that there exists such a goal which the obstacle strives to approach, expecting it to be willing to keep its current speed and heading. By continuous observation of a moving obstacle it is also possible to deduce a model of its dynamics, which describes feasible accelerations depending on its current speed and heading. Further details about acquiring models of velocity utilities and dynamic capabilities of other objects are beyond the scope of this paper.

Finally, the velocity obstacle PVO_j for object B_j is computed in a recursive manner, where predicted velocities are used in all but the terminal recursion step.

4.1 Formal Representation

Let $d \in \mathbb{N}$ the current recursive depth. Then the following equation

$$RU_j^d = \begin{cases} U_j D_j (1 - PVO_j^{d-1}) & \text{if } d > 0, \\ U_j D_j & \text{else,} \end{cases} \quad (13)$$

expresses that each object is assumed to derive its relative utility from recursive PVO considerations, and equation

$$V_j^d = \begin{cases} 1/w RU_j^d & \text{if } d > 0, \\ & w = \int_{\mathbb{R}^2} RU_j^d(v) d^2 v \text{ exists,} \\ & \text{and } w > 0, \\ V_j & \text{else.} \end{cases} \quad (14)$$

expresses the assumption that objects will move according to their relative utility function. Probabilistic velocity obstacles PVO_j^d of depth $d \geq 0$ are computed in the obvious way from depth- d models of other objects' velocities.

Computational demands will increase with the depth of the recursion, but, intuitively, one does not expect recursion depths of more than two or three to be of broad practical value, since such deeper modeling is not observed when we are walking as human beings among other humans.

4.2 Navigation with Recursive PVO

To navigate a mobile robot B_i using depth- d recursive probabilistic velocity obstacles, we repeatedly choose a velocity v_i maximizing RU_i^d . For $d = 0$, we get a behavior that only obeys the robot's utility function U_i and its dynamic capabilities D_i , but completely ignores other obstacles. For $d = 1$, we get the plain probabilistic velocity obstacle behavior as described in Section 3. Finally, for $d > 1$, the robot starts modeling the obstacles as perceptive and decision making.

5 Implementation

The implementation is given by Algorithm 1. Recursive function calls are not used, the models are computed starting from depth zero up to a predefined maximum depth.

Algorithm 1 RPVO(depth r , n objects)

```

1: Input: for  $i, j = 1, \dots, n, j \neq i$ 
    • object descriptions for  $PCC_{ij}$ 
    • velocities  $V_i$ 
    • dynamic capabilities  $D_i$ 
    • utilities  $U_i$ 
2: for  $i = 1, \dots, n$  do
3:    $V_i^0 \leftarrow V_i$ 
4:    $RU_i^0 \leftarrow D_i U_i$ 
5: end for
6: for  $d = 1, \dots, r$  do
7:   for  $i = 1, \dots, n$  do
8:      $RU_i^d \leftarrow D_i U_i \prod_{j \neq i} (1 - V_j^{d-1} * PCC_{ij})$ 
9:      $w \leftarrow \int_{\mathbb{R}^2} RU_i^d(v) d^2 v$ 
10:    if  $w > 0$  then
11:       $V_i^d \leftarrow (1/w) RU_i^d$ 
12:    else
13:       $V_i^d \leftarrow V_i^0$ 
14:    end if
15:  end for
16: end for
17: Output: recursive models  $V_i^d$  and  $RU_i^d$ 
    for  $i = 1, \dots, n$  and  $0 \leq d \leq r$ 

```

For implementation, objects like uncertain velocities, probabilistic collision cones and velocity obstacles have to be discretized. A function f is called discrete with respect to a partition $\Pi = \{\pi_1, \pi_2, \dots\}$ of \mathbb{R}^2 if the restriction of f to any $\pi_i \in \Pi$ is constant. We assume a unique partition for all functions in the algorithm. The supporting set $\sigma(f) \subseteq \Pi$ of a discrete function f is the set of cells $\pi_i \in \Pi$ where f is non-zero.

5.1 Complexity

We begin the complexity assessment by measuring the sizes of the supporting sets of the discretized functions used in Algorithm 1. Line 4 implies

$$\sigma(RU_i^0) \subseteq \sigma(D_i), \quad (15)$$

and from line 8 follows

$$\sigma(RU_i^d) \subseteq \sigma(D_i) \quad (16)$$

for $d > 0$. Line 3 implies

$$\sigma(V_i^0) = \sigma(V_i), \quad (17)$$

and from lines 11 and 13 follows

$$\sigma(V_i^d) \subseteq \sigma(D_i) \cup \sigma(V_i) \quad (18)$$

for $d > 0$, using the three preceding Equations.

Now we count the numbers of operations used in the algorithm, which we write down using $N_i = |\sigma(D_i) \cup \sigma(V_i)|$. Line 8 can be implemented to use $\mathcal{O}(N_i \cdot \sum_{j \neq i} N_j)$ operations. Lines 9, 11, and 13 each require $\mathcal{O}(N_i)$ operations, and are thus dominated by line 8. Therefore the loop starting in line 7 requires

$$\mathcal{O}\left(\sum_{i=1}^n (N_i \sum_{j \neq i} N_j)\right) \quad (19)$$

operations, and the loop starting in line 6 requires

$$\mathcal{O}\left(r \sum_{i=1}^n (N_i \sum_{j \neq i} N_j)\right) \quad (20)$$

operations. The complexity of the loop starting in line 6 clearly dominates the complexity of the initialization loop starting in line 2. Therefore Equation 20 gives an upper bound of the overall time complexity of our implementation. That is to say the dependence on the recursion depth is linear, and the dependence on the number of objects is $\mathcal{O}(n^2)$.

5.2 Experiments

A simulation of a dynamic environment has been used as a testbed for the presented approach. Results for some example situations are given below.

For two objects initially on an exact collision course, Fig. 2 shows the resulting motion for two pairs of depths. In Fig. 2(a) object A (depth 2) correctly models object B (depth 1) to be able to avoid collisions. As a result, object A stays on its course, forcing object B to deviate. Now if object A's assumption failed, i.e. object B won't avoid collisions, Fig. 2(b) shows that object A is still able to prevent a crash in this situation.

Another example is the situation where a fast object approaches a slower one from behind, i.e. where overtaking is imminent, which is depicted by Fig. 3. If the slow object B (depth 2) expects the fast object A (depth 1) from behind to avoid collisions, it mostly stays in its lane (Fig. 3(a)). On the other hand, if object B (now depth 3) assumes other objects to expect it to avoid collisions, a rather defensive driving style is realized (Fig. 3(b)).

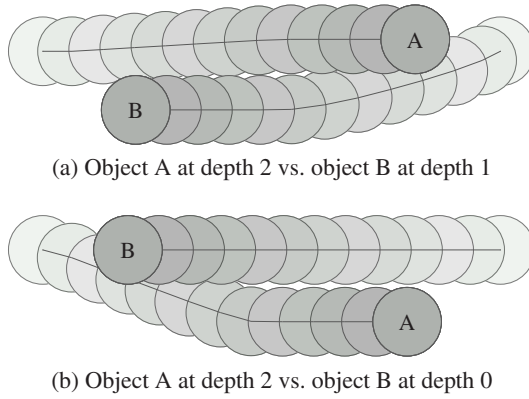


Fig. 2. Collision Course Examples

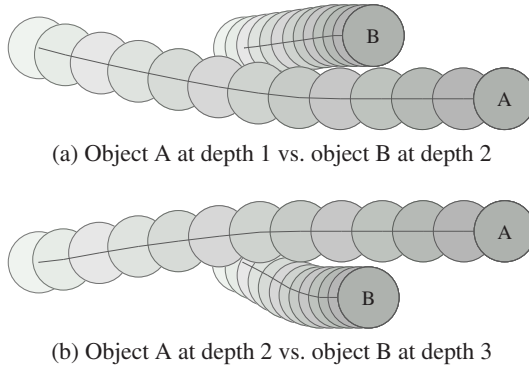


Fig. 3. Overtaking Examples

6 Discussion

Considering the experiments from the previous section, the nature of a robot's motion behavior appears to be adjustable by changing the evaluation depth. Depth 1 corresponds to a plain collision avoidance behavior. A robot using depth 2 will reflect on its environment and is able to exploit obstacle avoiding capabilities of other moving agents. This sometimes results in more aggressive navigation, which nevertheless may be desirable in certain situations: a robot which is navigating too defensively will surely get stuck in dense pedestrian traffic. In general, depth 3 seems to be more appealing, since a robot using that level of reflection assumes that the other agents expect it to avoid collisions, which results in a very defensive behavior with anticipating collision avoidance.

A rather different aspect of the presented recursive modeling scheme is that it can serve as a basis for an approach to reasoning about the objects in the environment.

That is to say, one could compare the observed motion of the objects to the motion that was predicted by recursive modeling, possibly discovering relationships among the objects. An example for such a relationship is deliberate obstruction, when one object obtrusively refrains from collision avoidance.

7 Conclusion

An approach to motion coordination in dynamic environments has been presented, which reflects the peculiarities of natural, populated environments: obstacles are not only moving, but also perceiving and making decisions based on their perception.

The presented approach can be seen as a twofold extension of the velocity obstacle framework. Firstly, object velocities and shapes may be known and processed with respect to some uncertainty. Secondly, the perception and decision making of other objects is modeled and included in the own decision making process.

Due to its reflective capabilities, the proposed navigation scheme may represent an interesting option for mobile robots sharing the environment with humans.

Acknowledgment

This work was supported by the German Department for Education and Research (BMB+F) under grant no. 01 IL 902 F6 as part of the project MORPHA.

References

1. M. Bennewitz, W. Burgard, and S. Thrun. Learning motion patterns of persons for mobile service robots. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2002.
2. P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17(7):760–772, July 1998.
3. A. F. Foka and P. E. Trahanias. Predictive autonomous robot navigation. In *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pages 490–495, EPFL, Lausanne, Switzerland, October 2002.
4. P. J. Gmytrasiewicz. *A Decision-Theoretic Model of Coordination and Communication in Autonomous Systems (Reasoning Systems)*. PhD thesis, University of Michigan, 1992.
5. J. Miura and Y. Shirai. Modeling motion uncertainty of moving obstacles for robot motion planning. In *Proc. of Int. Conf. on Robotics and Automation (ICRA)*, 2000.
6. Z. Shiller, F. Large, and S. Sekhavat. Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 3716–3721, Seoul, Korea, May 2001.

Learning Predictions of the Load-Bearing Surface for Autonomous Rough-Terrain Navigation in Vegetation

Carl Wellington¹ and Anthony Stentz²

¹ Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15201 USA
cwellin@ri.cmu.edu
http://www.ri.cmu.edu/people/wellington_carl.html

² Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15201 USA
axs@ri.cmu.edu
http://www.ri.cmu.edu/people/stentz_anthony.html

Abstract. Current methods for off-road navigation using vehicle and terrain models to predict future vehicle response are limited by the accuracy of the models they use and can suffer if the world is unknown or if conditions change and the models become inaccurate. In this paper, an adaptive approach is presented that closes the loop around the vehicle predictions. This approach is applied to an autonomous vehicle driving through unknown terrain with varied vegetation. Features are extracted from range points from forward looking sensors. These features are used by a locally weighted learning module to predict the load-bearing surface, which is often hidden by vegetation. The true surface is then found when the vehicle drives over that area, and this feedback is used to improve the model. Results using real data show improved predictions of the load-bearing surface and successful adaptation to changing conditions.

1 Introduction and Related Work

Automated vehicles that can safely operate in rough terrain hold the promise of higher productivity and efficiency by reducing the need for skilled operators, increased safety by removing people from dangerous environments, and an improved ability to explore difficult domains on earth and other planets. Even if a vehicle is not fully autonomous, there are benefits from having a vehicle that can reason about its environment to keep itself safe. Such systems can be used in safeguarded teleoperation or as an additional safety system for human operated vehicles.

To safely perform tasks in unstructured environments, an automated vehicle must be able to recognize terrain interactions that could cause damage to the vehicle. This is a difficult problem because there are complex dynamic interactions between the vehicle and the terrain that are often unknown and can change over time, vegetation is compressible which prevents a purely geometric interpretation of the world, there are catastrophic states such as rollover that must be avoided, and there is uncertainty in everything. In agricultural applications, much about the environment is known, but unexpected changes can occur due to weather, and the vehicle is often required to drive through vegetation that changes during the year. In more general off-road

exploration tasks, driving through vegetated areas may save time or provide the only possible route to a goal destination, and the terrain is often unknown to the vehicle.

Many researchers have approached the rough terrain navigation problem by creating terrain representations from sensor information and then using a vehicle model to make predictions of the future vehicle trajectory to determine safe control actions [1–4]. These techniques have been successful on rolling terrain with discrete obstacles and have shown promise in more cluttered environments, but handling vegetation remains a challenge.

Navigation in vegetation is difficult because the range points from stereo cameras or a laser range-finder do not generally give the load-bearing surface. Classification of vegetation and solid substances can be useful for this task, but it is not sufficient. A grassy area on a steep slope may be dangerous to drive on whereas the same grass on a flat area could be easily traversable. Researchers have modeled the statistics of laser data in grass to find hard objects [5], assigned spring models to different terrain classes to determine traversability using a simple dynamic analysis [4], and kept track of the ratio of laser hits to laser pass-throughs to determine the ground surface in vegetation [3].

The above methods all rely on various forms of vehicle and terrain models. These models are difficult to construct, hard to tune, and if the terrain is unknown or changing, the models can become inaccurate and the predictions will be wrong. Incorrect predictions may lead to poor decisions and unsafe vehicle behavior. In this work, we investigate model learning methods to mitigate this problem.

Other researchers have investigated the use of parameter identification techniques with soil models to estimate soil parameters on-line from sensor data [6,7], but these methods only determine the terrain that the vehicle is currently traversing. We are interested in taking this a step further and closing the loop around the vehicle predictions themselves by learning a better mapping from forward looking sensor data to future vehicle state. This allows the vehicle to use its experience from interacting with the terrain to adapt to changing conditions and improve its performance autonomously.

Our vehicle test platform is described in section 2 and our model-based approach to safeguarding in rough terrain is given in section 3. Section 4 explains the general approach of learning vehicle predictions and then describes how this is used to find the load-bearing surface in vegetation. Experimental results are given in section 5 and conclusions and future work are given in section 6.

2 Vehicle Platform and Terrain Mapping

Our project team [8] has automated a John Deere 6410 tractor (see figure 1). This vehicle has a rich set of sensors, including a differential GPS unit, a 3-axis fiber optic vertical gyro, a doppler radar ground speed sensor, a steering angle encoder, four custom wheel encoders, a high-resolution stereo pair of digital cameras, and two SICK laser range-finders (ladar) mounted on custom actively controlled scanning mounts. The first ladar on the roof of the vehicle is mounted horizontally and is



Fig. 1. Automated tractor test platform.

scanned to cover the area in front of the tractor. The ladar on the front bumper is mounted vertically and is actively scanned in the direction the tractor is steering. We are currently experimenting with a near-infrared camera and a millimeter-wave radar unit as well.

The approach described in this work builds maps using range points from multiple lasers that are actively scanned while the vehicle moves over rough terrain. The true ground surface is then found when the tractor drives over that area a number of seconds later. To make this process work, it is important that the scanned ladars are precisely calibrated and registered with each other in the tractor frame, the timing of all the various pieces of sensor data is carefully synchronized, and the vehicle has a precise pose estimate. Our system has a 13 state extended Kalman filter with bias compensation and outlier rejection that integrates the vehicle sensors described above into an accurate estimate of the pose of the vehicle at 75Hz. This pose information is used to tightly register the data from the ladars into high quality terrain maps.

The information from the forward looking sensors represents a massive amount of data in its raw form, so some form of data reduction is needed. One simple approach is to create a grid in the world frame and then combine the raw data into summary information such as average height for each grid cell. This approach makes it easy to combine range information from the two ladars on our vehicle and to combine sensor information over time as the vehicle drives. Figure 3 shows the type of terrain we tested on and a grid representation of this area using the average height of each cell.

3 Rough Terrain Navigation

The goal of our system is to follow a predefined path through rough terrain while keeping the vehicle safe. Path tracking is performed using a modified form of pure pursuit [8]. The decision to continue is based on safety thresholds on the model

predictions for roll, pitch, clearance, and suspension limits. These quantities are found by building a map of the upcoming terrain and using a vehicle model to forward simulate the expected trajectory on that terrain [2].

If the vehicle is moving relatively slowly and the load-bearing surface of the surrounding terrain can be measured, these quantities can be computed using a simple kinematic analysis. The trajectory of the vehicle is simulated forward in time using its current velocity and steering angle. A kinematic model of the vehicle is then placed on the terrain map at regular intervals along the predicted trajectory, and the heights of the four wheels are found in order to make predictions of vehicle roll and pitch. The clearance under the vehicle is important for finding body collisions and high centering hazards. It is found by measuring the distance from the height of the ground in each cell under the vehicle to the plane of the bottom of the vehicle. Our vehicle has a simple front rocker suspension, so checking the suspension limits involves calculating the roll of the front axle and comparing it to the roll of the rear axle. For smooth terrain with solid obstacles, this approach works well because accurate predictions of the load bearing surface can be found by simply averaging the height of the range points in the terrain map.

If there is vegetation, finding the load-bearing surface can be difficult because many laser range points hit various places on the vegetation instead of the ground. Simply averaging the points in a grid cell performs poorly in this case. One possible solution is to use the lowest point in each grid cell instead. This correctly ignores the range points that hit vegetation, but because there is inevitable noise in the range points (especially at long distances), this results in the lowest outlier in the noise distribution being chosen, thus underestimating the true ground height.

4 Learning Vehicle Predictions

To overcome the difficulties associated with creating vehicle and terrain models for a complex environment that may be unknown or changing, a learning method is proposed. At the highest level, this approach is about closing the loop around vehicle predictions, as shown in figure 2. A vehicle prediction is a mapping from environmental sensor information and current vehicle state to future vehicle motion. This mapping is learned by observing actual vehicle motion after driving over a given terrain. During training and execution, the vehicle makes predictions about the future state of the vehicle by reasoning about its current state and the terrain in front of the vehicle. Then, when the vehicle drives over that terrain, it compares its predictions to what actually happened. This feedback is used for continual learning and adaptation to the current conditions.

By closing the loop around vehicle predictions and improving the system models on-line, tuning a system to a given application is easier, the system can handle changing or unknown terrain, and the system is able to improve its performance over time.

The learning vehicle predictions approach has been applied to the problem of finding the load-bearing surface in vegetation. The system makes predictions of the

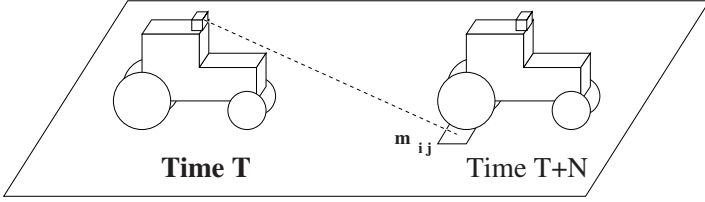


Fig. 2. Learning vehicle predictions. Features from map cell m_{ij} extracted at time T are used to make a prediction. Then, at time $T + N$ the vehicle traverses the area and determines if its prediction is correct. This feedback is used to improve the model.

load-bearing surface from features extracted from the laser range points. Then it drives over the terrain and measures the true surface height with the rear wheels. These input-output pairs are used as training examples to a locally weighted learner that learns the mapping from terrain features to load-bearing surface height. Once the load-bearing surface is known, parameters of interest such as roll, pitch, clearance, and suspension limits can easily be computed using a kinematic vehicle model as described in section 3.

This combination of kinematic equations with machine learning techniques offers several advantages. Known kinematic relationships do not need to be learned, so the learner can focus on the difficult unknown relationships. Also, the learned function can be trained on flat safe areas, but is valid on steep dangerous areas. If we learned the roll and pitch directly, we would need to provide training examples in dangerous areas to get valid predictions there.

4.1 Feature Extraction

As described in section 2, the range points from the ladars are collected over time in a world frame grid. In addition to maintaining the average and lowest height of points in each cell, we use an approach similar to [3] to take advantage of the added information about free space that a laser ray provides. We maintain a scrolling map of 3D voxels around the vehicle that records the locations of any hits in a voxel, as well as the number of laser rays that pass through the voxel. Each voxel is 50cm square by 10cm tall. We use a cell size of 50cm because that is the width of the rear tires on our tractor, which are used for finding the true ground height.

Four different features are extracted from each column of voxels in the terrain map. The average height of range points works well for hard surfaces such as roads and rocks. The lowest point may provide more information about the ground height if there is sparse vegetation. Voxels that have a high ratio of hits to pass-throughs are likely to represent solid objects, so the average of the points in these voxels may help determine the load-bearing surface. As shown in figure 4, the standard deviation from a plane fit provides a good measure of how “smooth” an area is, and works well as a discriminator between hard things like road and compressible things like weeds. We are currently working on other features that use color and texture information in addition to laser range points.

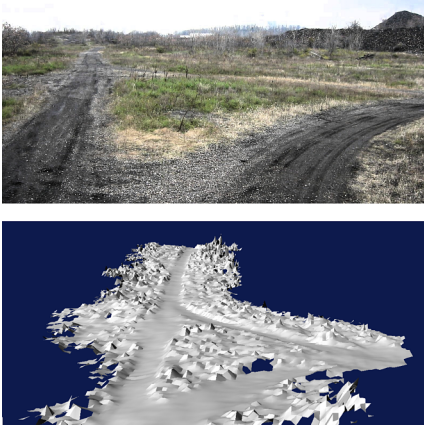


Fig. 3. Top: Test area showing dirt roads and vegetation. Bottom: Map of test area using average height.

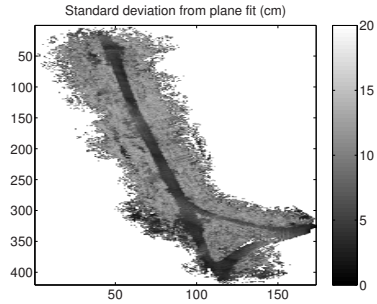


Fig. 4. The standard deviation from a plane fit is found for each cell in the terrain map to discriminate between hard things like road and compressible things like weeds.

4.2 Learning

By closing the loop around vehicle predictions, this approach produces a large amount of input-output pairs of training data. The system extracts features from the sensor data when making predictions and then records the true value when it drives over that area. This happens continuously, so the more the vehicle interacts with the environment, the more training data the learning system has to work with.

The mapping between the laser point features and the true ground height is unknown and potentially complex, so we use a general purpose function approximator for this task. Among the many possibilities, locally weighted learning [9] was chosen because it can accurately fit complex functions, it produces confidence estimates on its predictions, and there are online versions available.

A common form of locally weighted learning is locally weighted regression (LWR). Training with this algorithm simply involves inserting input-output pairs into memory. Then, when a new prediction is requested, the points in memory are weighted by a kernel function of the distance to the new query point, and a local multi-dimensional linear regression is performed on these weighted data points to produce a prediction. For good results, the kernel width must be chosen properly so that the resulting function does not over-fit or over-smooth the data. We use global leave-one-out cross validation to find the kernel width.

Standard statistical techniques for computing prediction bounds have been adapted to be used with this algorithm [10]. The size of the prediction bound depends both on the density of data points in the area, and on the noise in the outputs of nearby data points that cannot be explained by the model. The prediction bounds assume that the linear model structure is correct and the noise is zero-mean Gaussian. These assumptions can rarely be verified, but because they only have to be satisfied locally,

the prediction bounds still give useful information in practice about the confidence in the prediction.

Locally weighted learning stores all of its training data, so predictions take longer to compute as more training data is collected. This is not practical for systems such as ours that receive a continuous stream of data. Schaal [11] has described an on-line incremental version of LWR called receptive field weighted regression (RFWR). Instead of postponing all computations until a new prediction is requested, RFWR incrementally builds a set of receptive fields, each of which has a local regression model that is incrementally updated. The data points are then discarded, and predictions are made from a combination of nearby receptive fields. A forgetting factor is used to slowly discount old experience as it is replaced with new data.

5 Experimental Results

A set of experiments were performed to test the capabilities of this system. The first experiment shows the improved performance of the learned predictions and the usefulness of prediction bounds. The second experiment shows the system's ability to adapt to a change in the environment.

Although the analysis for the following experiments was performed off-line, the data used was collected in realistic conditions at the test site shown in figure 3. We collected approximately 25 minutes of data traveling on dirt roads and driving through varied vegetation often over a meter tall. Traveling at a speed of 1 m/s, we logged input-output pairs of laser features and corresponding rear wheel height at a rate of 4 examples per second, giving a total of 5700 data points. The data from the section shown in figure 3 was used as the test set, and the remaining 75% of the data was used for training.

5.1 Performance Improvement

The locally weighted regression technique described in section 4 was used to find a model for the training data described above. Leave-one-out cross validation was used to choose the kernel bandwidth. Figure 5 shows height prediction results for a section of the test set. The top graph shows the errors in predicted heights using the average height, the lowest point, and the learned result using all the features described in section 4.1. It can be seen that using the average point does quite poorly in vegetation and using the lowest point can result in outlier points being chosen. This especially occurs on roads where many laser points are recorded at long range (10m), resulting in higher uncertainty. The learned result has smaller errors than either of these because it is able to combine the different features in an appropriate way. The lower graph shows the 95% prediction intervals produced by LWR. The three times that the vehicle drives on the road are clearly shown by the tighter prediction bounds, and the learner has very little confidence in its predictions for some of the points in deep vegetation. These bounds are important because they could be used to modify the behavior of the vehicle. On simple terrain such as roads where the learner is

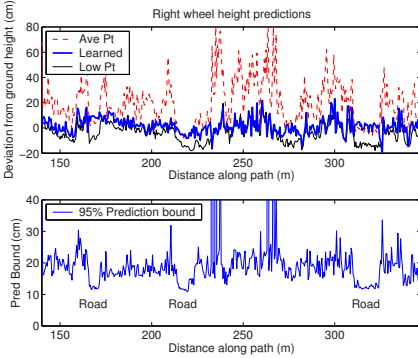


Fig. 5. Top: Height prediction deviations for right wheel (should be zero). Average point has trouble in grass, lowest point underestimates, learned predictions do better. Bottom: Learner produces prediction bounds on its estimate that are lower for the smooth road areas and high for some of the tall vegetation.

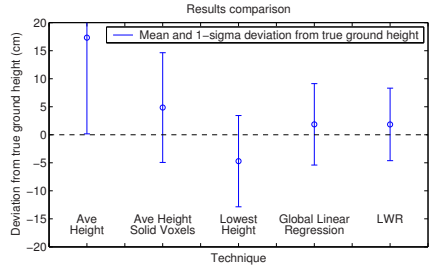


Fig. 6. Comparison of the performance of the different features along with global and local regression on the test set.

confident, the vehicle could drive faster or attempt more extreme angles. In difficult terrain with tall vegetation where the learner has low confidence, the vehicle could approach the area with caution or avoid it completely.

Figure 6 shows that combining all the features using global or local regression performs better than using any of the features individually. LWR can represent the small non-linearities in this problem and performs slightly better than global linear regression. More importantly, LWR produces local prediction bounds that reflect the confidence in the prediction at that particular location in the feature space, allowing the system to be cautious in areas it hasn't experienced before.

The height predictions in figures 5 and 6 use all the laser points that are collected for a given patch of terrain. In practice, this would only happen if the system made predictions of future vehicle motion just in front of its current location. This is not useful if the vehicle cannot react in this distance. Figure 7 shows the effect of making predictions further ahead of the vehicle. The plot shows that the spread of the prediction errors increases when the predictions are made farther into the future. The figure also shows the mean 95% prediction bound the learner produces for different prediction distances. This curve is similar to the 2σ (95%) level on the error, again showing the usefulness of the prediction bounds.

5.2 Adapting to Environmental Change

The above tests were performed in batch mode using locally weighted linear regression. In this experiment, the on-line receptive field weighted regression algorithm was used to test the system's ability to adapt to a changing environment. We collected more data in another test site that had dense vegetation approximately 0.75m high.

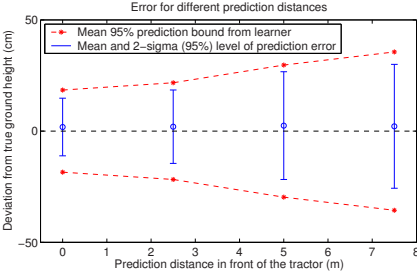


Fig. 7. Error of predictions on the test set at different distances using LWR. The prediction bounds from the learner are similar to the actual prediction errors.

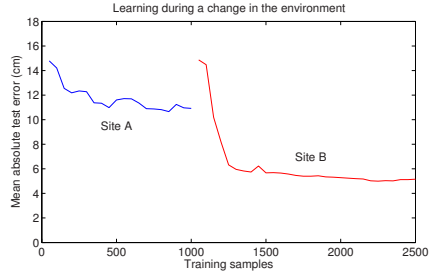


Fig. 8. After learning a model for Site A using RFWR, the system is transported to Site B. The error rises initially, but then the system adapts to the new environment.

This data was split into a training and test set. The training data was presented to the RFWR learning algorithm, and the prediction error on the test set was periodically computed.

Figure 8 shows the algorithm learning the characteristics of this site and reducing prediction error on the test set. Then, starting with sample 1000, the system was presented data used in the previous experiments. After being trained in the first site with dense vegetation, the algorithm did poorly in the new site with roads and more varied vegetation. However, the learner quickly adapted to the new environment and ended up with a prediction error similar to the batch mode version of LWR on the test set. This experiment shows the importance of using an adaptive approach to be able to handle a changing environment.

6 Conclusions and Future Work

We have described the general approach of learning vehicle predictions for local navigation, and we have applied the technique to the problem of finding the load-bearing surface in vegetation. The locally weighted learning solution to this problem performed better than using a particular feature or performing global linear regression. Another benefit of this technique is that it produces prediction bounds on its estimate, and these were shown to be fairly accurate. Finally, the ability of the system to adapt to changing environmental conditions was shown.

Future work in this area will include an investigation into other features such as color and texture from camera data, and the use of the prediction bounds for better vehicle control. We will also investigate whether the technique of learning vehicle predictions can be applied to other rough-terrain navigation problems such as finding terrain friction characteristics or dynamic vehicle response.

Acknowledgements

This work has been supported by John Deere under contract number 476169. The authors would like to thank Jeff Schneider for the use of his Vizier code for locally weighted learning, and Stefan Schaal for the use of his code for incremental locally weighted learning.

References

1. M. Daily, J. Harris, D. Keirsey, D. Olin, D. Payton, K. Reiser, J. Rosenblatt, D. Tseng, and V. Wong, "Autonomous cross-country navigation with the ALV," in *IEEE Int. Conf. on Robotics and Automation (ICRA 88)*, vol. 2, pp. 718–726, April 1988.
2. A. Kelly and A. Stentz, "Rough terrain autonomous mobility - part 2: An active vision, predictive control approach," *Autonomous Robots*, vol. 5, pp. 163–198, May 1998.
3. A. Lacaze, K. Murphy, and M. DelGiorno, "Autonomous mobility for the Demo III experimental unmanned vehicles," in *Assoc. for Unmanned Vehicle Systems Int. Conf. on Unmanned Vehicles (AUVSI 02)*, July 2002.
4. A. Talukder, R. Manduchi, R. Castano, K. Owens, L. Matthies, A. Castano, and R. Hogg, "Autonomous terrain characterisation and modelling for dynamic control of unmanned vehicles," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 02)*, pp. 708–713, October 2002.
5. J. Macedo, R. Manduchi, and L. Matthies, "Ladar-based discrimination of grass from obstacles for autonomous navigation," in *Int. Symposium on Experimental Robotics (ISER 00)*, December 2000.
6. A. T. Le, D. Rye, and H. Durrant-Whyte, "Estimation of track-soil interactions for autonomous tracked vehicles," in *IEEE Int. Conf. on Robotics and Automation (ICRA 97)*, vol. 2, pp. 1388–1393, April 1997.
7. K. Iagnemma, H. Shibly, and S. Dubowsky, "On-line terrain parameter estimation for planetary rovers," in *IEEE Int. Conf. on Robotics and Automation (ICRA 02)*, May 2002.
8. A. Stentz, C. Dima, C. Wellington, H. Herman, and D. Stager, "A system for semi-autonomous tractor operations," *Autonomous Robots*, vol. 13, pp. 87–104, July 2002.
9. C. Atkeson, A. Moore, and S. Schaal, "Locally weighted learning," *AI Review*, vol. 11, pp. 11–73, April 1997.
10. S. Schaal and C. G. Atkeson, "Assessing the quality of learned local models," in *Advances in Neural Information Processing Systems (NIPS 94)*, pp. 160–167, 1994.
11. S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," in *Neural Computation*, vol. 10, pp. 147–184, 1998.

A Terrain-Aided Tracking Algorithm for Marine Systems

Stefan Williams and Ian Mahon

ARC Centre of Excellence for Autonomous Systems
School of Aerospace, Mechanical and Mechatronic Engineering
University of Sydney, Sydney, NSW, 2006, Australia

Abstract. This paper presents a novel method for incorporating unstructured, natural terrain information into the process of tracking of underwater vehicles. Terrain-aided navigation promises to revolutionise the ability of marine systems to track underwater bodies in deep-water applications. This work represents a crucial step in the development of underwater technologies capable of long-term, reliable deployment. A particle based estimator is used to incorporate observations of altitude into the estimation process using *a priori* map information. Results of the application of this technique to the tracking of a towed body and a ship operating in Sydney Harbour are shown.

1 Introduction

This paper presents a novel method for incorporating unstructured, natural terrain information into the process of tracking of underwater vehicles. Terrain-aided navigation promises to revolutionise the ability of marine systems to track underwater bodies in deepwater applications. The development of techniques to extract terrain aiding information from complex natural features, such as coral reefs and the natural variations on the sea floor, is an area of active research to be developed as part of this work [5][13]. The ability to use natural features will allow a submersible body to be deployed in a large range of environments without the need to introduce artificial beacons or rely on acoustic tracking technology - a very significant innovation in this area of work. This work represents a crucial step in the development of underwater technologies capable of long-term, reliable deployment.

This paper is organized as follows. Section 2 introduces some of the difficulties inherent with positioning of underwater systems. Section 3 makes reference to bathymetric maps available to many scientific expeditions and presents a particle based estimator used to incorporate observations of altitude into the estimation process. Section 4 presents results of the application of these techniques to a simulated deployment of a towed body. Finally, Section 5 summarizes the findings and presents directions for future work.

2 Underwater Positioning

While many land-based tracking systems use GPS to provide accurate position updates for navigation, a body operating underwater does not typically have access to

this type of information. Many underwater systems rely on fixed acoustic transponders that are surveyed into the system's work area [11]. These transponders are then interrogated to triangulate the position of the underwater body. The surveying of these transponders can be a costly and time consuming affair - especially at the depths at which these bodies often operate and their performance can vary with conditions within the water column in which the vehicle is operating.

Other acoustic positioning systems are able to track a body without the introduction of fixed transponders through the use of ultra-short baseline technology. By comparing the phase lag of a ping received from a transponder affixed to the submerged body, the range, elevation and azimuth to a submerged body can be computed. These systems are also very susceptible to conditions in the water column and noise and alignment of the ship board hydrophone. All of these factors suggest that the use of terrain aiding information will prove invaluable to the deployment of underwater bodies in situations where acoustic positioning system performance is affected.

To date very little work has been done on the use of terrain information for the purposes of localisation in unmanned underwater applications. Some work has investigated underwater positioning using information from sonar in swimming pool environments [7] and other work has looked at methods for constructing seafloor terrain elevation maps [3] but these techniques have not seen widespread deployment on systems operating in real, unstructured environments.

3 Bathymetric Terrain Maps

In underwater scientific missions, *a priori* maps are often available. Bathymetric data collected by ships can be compiled to generate sea floor elevation maps [4][6] and Geospatial Information Systems (GIS) exist that allow this information to be efficiently accessed. The work undertaken as part of this project will allow bathymetric data to be used to aid in the process of localising an underwater body. Methods considered here examine the use of terrain-elevation data from a known map to aid the localisation process. Observations of altitude, when combined with models of the motion of the vehicle, can serve to bound its likely position. Recent work in the field has shown that this type of information can be used to bound the estimation accuracy of a filter [1,10].

The first step towards using terrain information for the purposes of navigation is to incorporate information provided by *a priori* maps into the estimation process, as shown in Figure 1. Seafloor elevation maps represent a strong source of information that can be used to improve the localisation process. Given sufficient information in the terrain structure, observations of terrain elevation provide an external observation of the likely pose of the vehicle. Although individual observations of altitude may not be sufficient to reliably identify the location of the body, the trend in the terrain elevation can serve to reduce the feasible locations of the submerged body. This is especially true in areas of the sea floor with unique terrain signatures, such as crevices and hills.

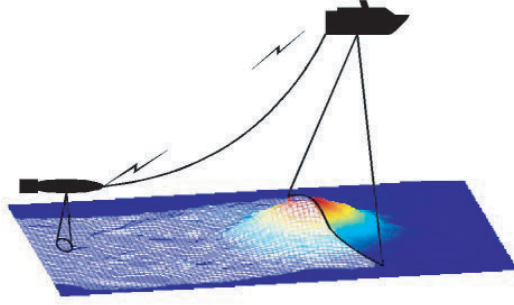


Fig. 1. Terrain-aided navigation using information available from a surface vessel, on-board sensors and an *a priori* map. The surface vessel has access to GPS signals allowing the observations of terrain elevation to be transformed into the earth's inertial frame with relative ease within an area bounded by the deployed tether.

By using Bayesian methods, it is possible to incorporate this information into the navigation loop by computing the probability distribution of the pose of the vehicle given an *a priori* map and observations taken by the body

$$p(x_k|M, U^k, Z^k) = \frac{p(z_k|M, u_k, x_k)p(x_{k-1}|M, U^{k-1}, Z^{k-1})}{p(z_k|M, U^k, Z^k)} \quad (1)$$

where x_k is the state at time k , M is the map, z_k is the current observation, u_k is the vehicle control input and U^k and Z^k are the sequences of control inputs and observations to time k .

Recent work has examined the possibility of using Monte Carlo methods for position tracking given a map of the environment, as shown in related work in indoor office environments [9], in road following [2], in aerial navigation [1] and for underwater applications [8]. This approach is often implemented as a particle filter. The particle filter models an unknown probability distribution by a large number of samples. Given sufficient particles, the distribution will provide an accurate model of the underlying process it is attempting to describe. The particle filter is able to efficiently track multi-modal and non-gaussian probability densities. This makes it an ideal choice for map-based localisation.

In the case considered here, the states of interest are the position and the velocities of the submerged body in the inertial frame of reference. Under the assumption that the UUV roll and pitch angles are small relative to the beam width of the altimeter (an assumption that will be verified in the true system), the observation of altitude is dependent primarily on the position of the submerged body. Maintaining an estimate of the position of the submerged body should therefore be sufficient to allow the submerged body to be tracked during deployment.

Each sample of the state vector, referred to as particle \mathbf{X}_i , is therefore represented by six states

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{x}_i^e \\ \dot{\mathbf{x}}_i^e \end{bmatrix} \quad (2)$$

with

$$\mathbf{x}_i^e = [x_i^e \ y_i^e \ z_i^e]^T \quad (3)$$

$$\dot{\mathbf{x}}_i^e = [\dot{x}_i^e \ \dot{y}_i^e \ \dot{z}_i^e]^T \quad (4)$$

Given that there are limits to the rate at which the submerged body will move when submerged, it is possible to use a constant velocity model to predict the movements of the submerged body.

$$\begin{bmatrix} \mathbf{x}_i^e(k) \\ \dot{\mathbf{x}}_i^e(k) \end{bmatrix} = \begin{bmatrix} I & \Delta T \cdot I \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{x}_i^e(k-1) \\ \dot{\mathbf{x}}_i^e(k-1) \end{bmatrix} + w(k) \quad (5)$$

where $w(k)$ represents the growth in uncertainty of position and velocity over the interval ΔT . Each particle is predicted forward using this simple motion model at each timestep.

The resulting estimates can then be bounded using the observations of range to the ship, depth and altitude to eliminate unlikely particles from the estimation process. The update stage consists of resampling the particle distribution according to the likelihood that the current observation is received given a particular sample of the state space. The likelihood function computes the probability that a given observation, $z(k)$, is received given the state $\mathbf{X}_i(k)$

$$p^i = p(z(k)|\mathbf{X}_i(k)) \quad (6)$$

The probability density is then resampled according to this likelihood. This step involves selecting N samples from the set $\{\mathbf{X}_i(k)\}_{i=1}^N$ where the probability to take sample i is p^i .

A particular advantage of this approach to the filtering process is the ability to represent and use non-Gaussian estimates, observations and constraints. Given sufficient particles, the filter is able to track multiple hypotheses until they are resolved, rather than committing to a single estimate, which may turn out to be erroneous. This makes it an ideal tool for use with map-based observations where a single observation of terrain elevation is often not sufficient to resolve the position of the body. Additional information, such as the fact that the submerged body is not likely to be outside of a particular arc length behind the vehicle or the relative water speed in the vicinity of the ship as shown in Figure 2, can easily be accommodated using this approach.

4 Results

This section examines results of the application of these techniques to the tracking problem for a towed body. In this simulation, the towed body is assumed to be

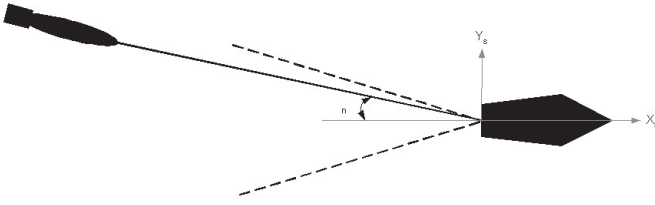


Fig. 2. The constraint applied to the particle distribution. Given the towing configuration of the target system, the towed body will be located within some arc-length behind the ship.

equipped with an altimeter with a maximum range of 750m and a depth sensor. The ship is equipped with a GPS receiver for observing its position as well as an acoustic ranging device capable of providing noisy range measurements between the towed body and the ship. The towed body is deployed from the ship during the course of the simulation and descends to a maximum depth of 1200m, which is in close proximity to the sea floor. Table 1 shows the parameters used in this simulation. This scenario has been motivated by recent work undertaken in the area of tracking for deep sea fishing and benthic surveying applications. It clearly lends itself well to the deployment of Unmanned Underwater Vehicles.

Table 1. Simulation Parameters

item	description	value
N	number of particles	2500
v_s	ship speed	3.5m/s
d	mean sea floor depth	1350m
L	cable length	2000m
t_d	deployment time	640s
t_T	total time	960s
R_a	altimeter range	750m
σ_a	altitude accuracy	5m
σ_d	depth accuracy	5m
σ_r	range accuracy	5% of range
σ_p	GPS position accuracy	1m
ψ_{max}	maximum bearing	+/- 15deg

Figure 3 (a) shows a view of the particle filter operating in 3D. The particle clouds represent the resampled distribution of the particles after every 40th observation. As can be seen, the particles spread out along the arc described by the distance between the ship and the submerged body until the altimeter is within range. Subsequently, the distributions converge to compact distributions around the true towed body position. This phenomenon is clearly evident in the top view shown in Figure 3 (b).

Since the distributions of particles are fairly uni-modal in this application, a measure of the mean and standard deviation of the particle density can be used

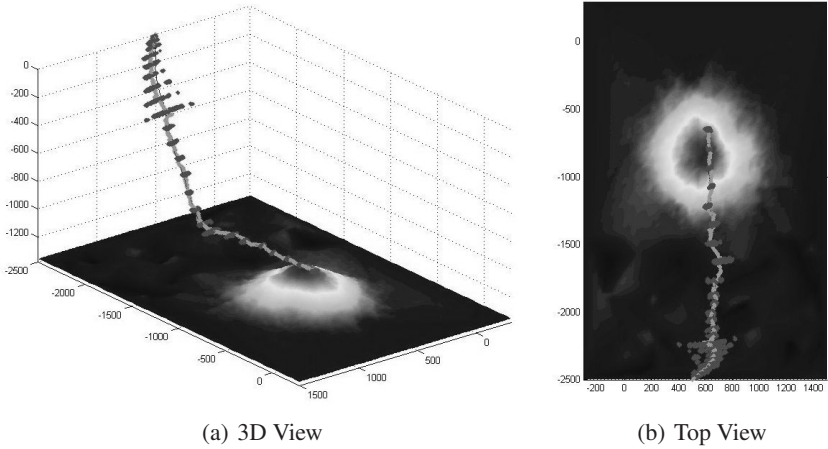


Fig. 3. The means of the particle distributions and a subset of the particle clouds in 3D shown relative to a rendered view of the terrain map. Notice how the particle cloud is spread out along the arc defined by the range measurement until the towed body is close enough to the sea floor to begin receiving observations of altitude.

as a measure of the probability distribution. This information might then be used to display the estimated position of the towed body and to give a measure of the uncertainty in the estimate. Figure 4 shows the mean state estimates and the errors between the estimate and the true location together with the 2σ bounds of the particle distribution. The filter is clearly maintaining consistent bounds on the estimation error. It is also interesting to note how the uncertainty in position drops dramatically once the altimeter is within range of the sea floor at $t = 375s$.

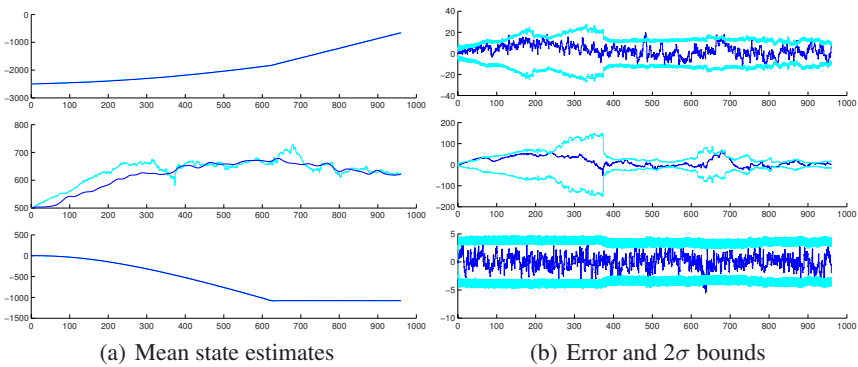


Fig. 4. (a) The mean of particle state estimates for x , y and z . (b) The errors between the mean estimate and true position. The variances drop dramatically once the system begins receiving altitude readings at approximately 375 seconds.

The use of terrain information will be the key source of accuracy with this method. For areas in which the map information is sparse the observation of altitude will not provide much information with which to bound the uncertainty in position. The positioning accuracy achievable will also depend on the terrain over which the vehicle is travelling. A flat, uniform bottom yields no information to bound the estimate of the filter. In this case, the uncertainty in the estimate will grow along the arc subtended by the range observation. If, on the other hand, some unique terrain features are present, such as hills and crevices, the probability distribution will converge to a compact estimate. As shown in the example presented here, the 2σ uncertainty bounds converge to approximately 10m in the X and Y position estimates. The depth accuracy remains constant and is a function of the accuracy of the depth sensor.

The uncertainty in the position estimate will grow while the body is not receiving altimeter readings. As shown in Figure 4, the error in the lateral position of the towed body relative to the ship grows large since there is no information available to the filter. Once the altimeter readings are received, the uncertainty in both the X and Y positions are reduced. So long as the trend in the terrain elevation remains fairly unique, the uncertainty will remain small.

4.1 Sydney Harbour Demonstration

In order to facilitate the demonstration of these techniques, data sets taken in Sydney Harbour have been acquired. This data includes a detailed bathymetric map of the harbour, shown in Figure 5 (a), and ship transect data, including GPS and depth soundings, shown in Figure 5 (b). This data has kindly been donated by the Australian Defence Science and Technology Organization (DSTO) in relation to their hosting of the 3rd Shallow Water Survey Conference held in Sydney in November, 2003.

The particle filter based techniques described in this paper have been applied to these data sets. Figure 6 shows results of these tests. The ship location is initially assumed to be unknown and particles are distributed randomly across the extent of the Harbour. The GPS fixes were used to generate noisy velocity and heading control inputs to drive the filter predictions. Observations of altitude using the ship's depth sounder were then used to validate the estimated particle locations using a simple Gaussian height model relative to the bathymetry in the map. As can be seen in the figure, the filter is able to localise the ship and successfully track its motion throughout the deployment. The particle clouds converge to the true ship position within the first 45 observations and successfully track the remainder of the ship track. Figure 7 shows the errors between the ship position and heading estimates generated by the filter and the GPS positions.

The assumption that the initial ship location is unknown is somewhat unrealistic for the target application of these techniques as submersibles will generally be deployed from a known initial location with good GPS fixes. This represents a worst case scenario, however, and it is encouraging to see that the technique is able to localise the ship even in the absence of an initial estimate of its position.

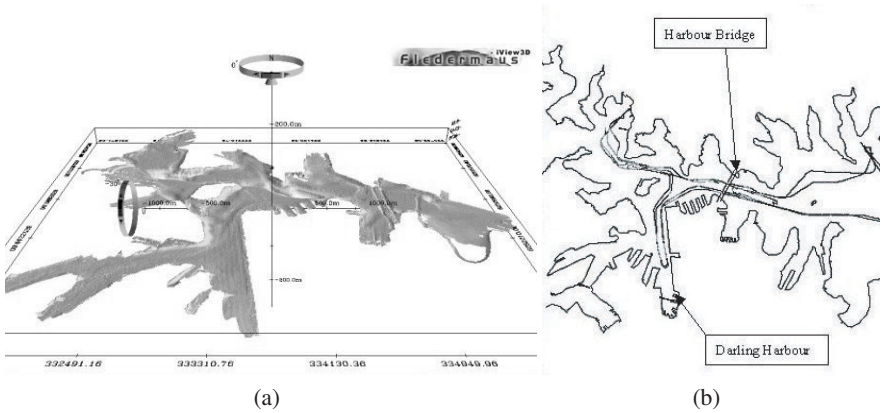


Fig. 5. Sydney Harbour bathymetric data. (a) The Harbour contains a number of interesting features, including the Harbour tunnel on the right hand side and a number of large holes which will present unique terrain signatures to the navigation filter. (b) The ship path for the Sydney Harbour transect. Shown are the contours of the harbour together with the path of the vehicle. Included in this data set are the GPS position and depth sounder observations at 5s intervals.

5 Conclusions

The proposed terrain-aided navigation scheme has been shown to reliably track a ship position in a harbour situation given depth soundings and a bathymetric map of the harbour. This technique is currently being augmented to support observations using a multi-beam or scanning sonar in preparation for deployment using the Unmanned Underwater Vehicle Oberon available at the University of Sydney's Australian Centre for Field Robotics.

Following successful demonstration of the map based navigation approach, the techniques developed will be applied to building terrain maps from the information available solely from the vehicle's on-board sensors. There is considerable information contained in strong energy sonar returns received from the sea floor as well as in the images supplied by on-board vision systems. This information can be combined to aid in the identification and classification of natural features present in the environment, allowing detailed maps of the sea floor to be constructed. These maps can then be used for the purposes of navigation in a similar manner to that of the more traditional, parametric feature based SLAM algorithm [13,12].

Acknowledgements

The authors wish to acknowledge the support provided under the ARC Centre of Excellence Program by the Australian Research Council and the New South Wales government. Thanks must also go to the staff of the Defence Science and Technology Organization (DSTO) for making the Sydney Harbour bathymetry and transect data available.

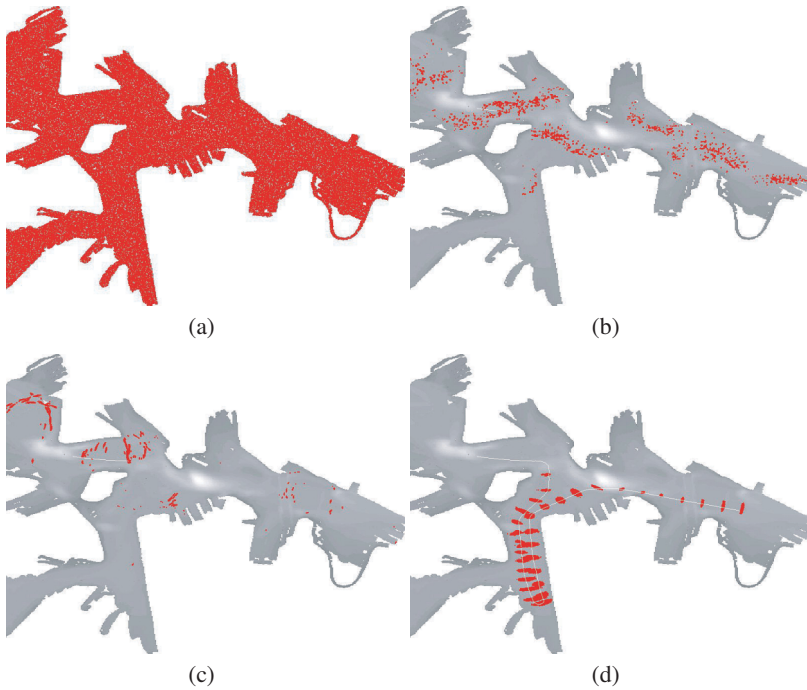


Fig. 6. Monte Carlo localisation example using the Sydney Harbour bathymetric map. The line represents the ship track in this deployment and the particles are shown overlaid on the figure. (a) The particles are initially drawn from the uniform distribution across the extent of the harbour. (b) The potential location of the ship is reduced to areas of the harbour with a common depth to the start of the trial and (c) begin to converge on the true ship location. (d) Once the particles have converged to the actual position of the ship, its motion is tracked as additional observations are taken. As can be seen, the particle clouds track the true ship path over the extent of the run in spite of there being no absolute observations of the ship position.

References

1. N. Bergman, L. Ljung, and F. Gustafsson. Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine*, 19(3):33–40, 1999.
2. F. Gustafsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P-J Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Trans. on Signal Processing*, 1999.
3. A.E. Johnson and M. Hebert. Seafloor map generation for autonomous underwater vehicle navigation. *Autonomous Robots*, 3(2-3):145–68, 1996.
4. D. Langer and M. Hebert. Building qualitative elevation maps from underwater sonar data for autonomous underwater navigation. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, volume 3, pages 2478–2483, 1991.

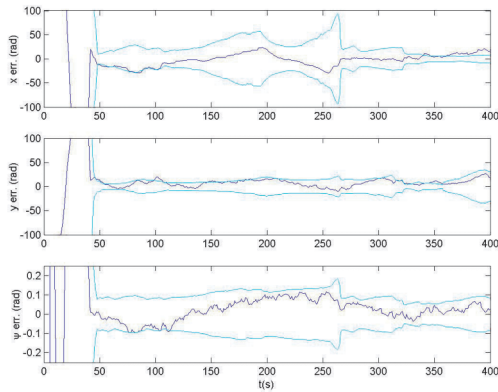


Fig. 7. The error between the mean of the particle densities and the GPS positions. The errors are bounded by the 2σ error bounds for the distributions.

5. S. Majumder, S. Scheduling, and H.F. Durrant-Whyte. Sensor fusion and map building for underwater navigation. In *Proc. Australian Conf. on Robotics and Automation*, pages 25–30. Australian Robotics Association, 2000.
6. C. De Moustier and H. Matsumoto. Seafloor acoustic remote sensing with multibeam echo-sounders and bathymetric sidescan sonar systems. *Marine Geophysical Researches*, 15(1):27–42, 1993.
7. V. Rigaud and L. Marc. Absolute location of underwater robotic vehicles by acoustic data fusion. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, volume 2, pages 1310–1315, 1990.
8. R. Karlsson, F. Gustafsson, and T. Karlsson. Particle filtering and cramer-rao lower bound for underwater navigation. In *Internal Report LiTH-ISY-R-2474*, 2002.
9. S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, 1998.
10. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 2000.
11. L. Whitcomb, D. Yoerger, H. Singh, and J. Howland. Advances in underwater robot vehicles for deep ocean exploration: Navigation, control and survey operations. *The Ninth International Symposium on Robotics Research*, pages 346–353, 1999.
12. S.B. Williams, G. Dissanayake, and H.F. Durrant-Whyte. Constrained initialisation of the simultaneous localisation and mapping algorithm. In *Proc. Intl. Conference on Field and Service Robotics*, pages 315–320, 2001.
13. S.B. Williams, G. Dissanayake, and H.F. Durrant-Whyte. Towards terrain-aided navigation for underwater robotics. *Advanced Robotics*, 15(5):533–550, 2001.

Experimental Results in Using Aerial LADAR Data for Mobile Robot Navigation

Nicolas Vandapel, Raghavendra Donamukkala, and Martial Hebert

The Robotics Institute
Carnegie Mellon University
vandapel@ri.cmu.edu

Abstract. In this paper, we investigate the use of high resolution aerial LADAR (LAsER De-tectiOn And Ranging) data for autonomous mobile robot navigation in natural environments. The use of prior maps from aerial LADAR survey is considered for enhancing system performance in two areas. First, the prior maps are used for registration with the data from the robot in order to compute accurate localization in the map. Second, the prior maps are used for computing detailed traversability maps that are used for planning over long distances. Our objective is to assess the key issues in using such data and to report on a first batch of experiments in combining high-resolution aerial data and on-board sensing.

1 Introduction

Autonomous mobility in unstructured, natural environments is still a daunting challenge due to the difficulty in analyzing the data from mobility sensors, such as stereo cameras or laser range finders. Recent developments make it possible and economical to acquire high-resolution aerial data of an area prior a ground robot traverses it. The resolution of former digital elevation maps is too limited to be used effectively for local robot navigation. New high-resolution aerial mapping systems opens the door to preprocessing a terrain model at a resolution level comparable to the one produced by on-board sensors.

In this paper, we investigate the use of high resolution aerial LADAR data for autonomous mobile robot navigation in natural environments. The use of prior maps from aerial survey is considered for enhancing system performance in two areas. First, the prior maps are used for registration with the data from the robot in order to compute accurate localization in the map. Second, the prior maps are used for computing detailed traversability maps that are used for planning over long distances. Our objective is to assess the key issues in using such data and to report on a first set of experiments in combining high-resolution aerial data and on-board 3-D sensing.

In the application considered here, the typical mission scenario aims at performing waypoint navigation over hundreds of meters in a rough terrain cluttered with various types of vegetation. The ground vehicle system, built by General Robotics Dynamic Systems (GDRS), is equipped with a three dimensional (3-D) laser sensor, an inertial navigation unit and a Global Positioning System (GPS). Overhead LADAR data is provided prior to the mission.

Using aerial LADAR data poses two main challenges: the volume of data and the nature of the terrain. The aerial LADAR data contains typically 44 millions of 3-D

points each associated with an identifier and a reflectance value. The area mapped covers $2.5 \text{ km} \times 3.0 \text{ km}$. Such a large data set is enormous amount of data to use effectively on-board a robot. In addition, the data includes two components of the terrain - the vegetation cover and the terrain surface, which need to be discriminated from one to another. The work reported here is a first step toward using this type of high-resolution aerial data in conjunction with data from a robot's mobility sensor.

The paper is organized as follows: The second section presents the field test sites and the sensors we used. The third section deals with the details of vegetation filtering both with ground and aerial LADAR data. In Sections 4 and 5 our work on vegetation filtering is in the context of two problems: robot position estimation using 3-D terrain registration and path planning.

2 Data Sets

This section introduces the various data sets used to produce the results presented in this article as well as the sensors that collected them.

2.1 Data Collection

The data used for this work was collected on two different sites, characteristic of very different types of terrain: the APHill site is a wooded areas with large concentrations of tree canopies; the Yuma site is a desert area with scattered bush. The data used in this article was collected during these two field tests; unless explicitly noted, these results were obtained after the field tests.

2.2 Sensors

We used the GDRS mobility laser scanner [18], mounted on a ground mobile robot, and a Saab TopEye mapping system mounted on a manned helicopter [1], to collect data.

The ground LADAR is mounted on a turret sitting at the front of the ground vehicle. The vehicle is built using the chassis of a 4 wheels drive All Terrain Vehicle (ATV). The laser has a maximum range of 80 meters and a 7.5 cm range resolution. In the configuration used in these experiments, its field of view is $90^\circ \times 15^\circ$. In addition, the sensor can pan and tilt by $\pm 90^\circ \times \pm 15^\circ$ for increasing terrain coverage.

The aerial mapping system is operated at 400 meter above the ground. The laser beam is deflected by an oscillating mirror which produces a Z-shaped laser track along the flight path. The range resolution is 1 cm and the point position accuracy varies between 10 and 30 cm, depending on the altitude. The laser records two echoes per pulse (first and last), but only objects taller than 1.8 meter will produce two echoes. For each demonstration, the helicopter flew over the test area along several directions to produce higher point density, 4 to 30 points per square meter.

3 Vegetation Filtering

In this section we justify the need for removing the vegetation from the raw data, we review the issues of perceiving vegetation with laser sensors, we present a state of the art of the techniques commonly used, and we explain the two methods implemented for the aerial and the ground LADAR data.

3.1 Motivation

Vegetation constitutes a major challenge for robot navigation for the following reasons: 1) Vegetated areas are unstable features, susceptible to natural changes and human activities, 2) It is difficult to recover and model the shape of trees and bushes, 3) Vegetation can prevent the detection of hazards such as trenches or rocks, 4) Vegetation such as trees and bushes might constitute a hazard, whereas grass might not and overhanging branches are sometimes difficult to detect, 5) Vegetation prevents the detection of the terrain surface, which is used in terrain-based localization, and is required when the canopies produce GPS drop-off.

3.2 Vegetation and Active Range Sensors

Issues related to the interaction of an aerial LADAR sensor with ground cover can be summarized into three broad categories: sensing, data interpretation, and ground versus aerial sensing.

The foliage penetration rate will depend on: the canopy density (winter preferable); the scanning angle (nadir preferable); the laser footprint on the ground (small aperture, low altitude preferable). The signal returned might contain several echoes, one for the canopy and one for the ground.

A change in elevation can be interpreted as a change in the ground cover (terrain versus trees) or as a change in the terrain elevation. Because of the range dispersion when the laser beam hits multiple objects at different depths, the range measurements will be erroneous, depending on the flight path and the scanning angle.

Ground scans are less sensitive to the effects described above because the laser range is shorter (dozens versus hundreds of meters) and so the beam footprint on the target is smaller. Another difference is the geometry of the scene. Aerial LADAR will most likely map the ground and the top of the canopy. Trunks and small obstacles will not be perceived. With a ground LADAR, the bottom of the canopy, small obstacles and trunks will be seen. Trunks will produce range shadows, occluding a large part of the terrain. The last difference will be the density of points, with a difference of two orders of magnitude.

3.3 State of the Art

Filtering LADAR data has been mainly studied in the remote sensing community with three objectives: producing surface terrain models [13] (in urban or natural

environment), studying forest biomass [14], and inventoring forest resources [8]. To filter LADAR data authors used linear prediction [13], mathematical morphology (grey opening) [5], dual rank filter [15], texture [6], and adaptive window filtering [17]. All these methods are sensitive to the terrain slope. In the computer vision community, Mumford [10] pioneered the work for ground range images. Macedo [16] and Matthies [3] focused on obstacle detection among grass. In [7], Hebert proposed to use 3-D points local shape distribution to segment natural scenes into linear, scatter and solid surface features.

3.4 Methods Implemented

We implemented two methods to filter the vegetation. The first one takes advantage of the aerial LADAR capability to detect multiple echoes per laser pulse emitted. The ground LADAR does not have this capability, so a second filtering method had to be implemented.

Multi-echoes based filtering The LADAR scans from multiple flights are gathered and the terrain is divided into a grid with $1\text{ m} \times 1\text{ m}$ cells. Each point falling within a given cell is classified as ground or vegetation by k-mean clustering on the elevation. Laser pulses with multiple echoes (first and last) are used to seed the two clusters (vegetation and ground respectively). Single echo pulses are assigned initially to the ground cluster. After convergence, if the difference between the mean value of the two clusters is less than a threshold, both clusters are merged into the ground cluster. The clustering is performed in groups of 5×5 cells centered at every cell in the grid. As we sweep the space, each point is classified 25 times and a majority vote defines the cluster to which the point is assigned. This method has been used to produce the results presented in section 5.

Cone based filtering We had to implement a new method for filtering the vegetation from ground data. Our approach is inspired by [19] and is based on a simple fact: the volume below a ground point will be free of any LADAR return. For each LADAR point, we estimate the density of data points falling into a cone oriented downward and centered at the point of interest.

While the robot traverses a part of the terrain, LADAR frames are registered using the Inertial Navigation System (INS). The INS is sensitive to shocks (e.g., a high velocity wheel hitting a rock), which causes misalignment of consecutive scans. In order to deal with slightly misaligned frames, we introduce a blind area defined by the parameter ρ (typically 15 cm). The opening of the cone (typically $10\text{-}20^\circ$) depends on the expected maximum slope in the terrain and the distribution of the points.

This approach has been used to produce the results presented section 4. Our current implementation filters 67,000 points spread over $100\text{ m} \times 100\text{ m}$, in 25 s on a Pentium III, 1.2 GHz.

4 Terrain Registration

In this section we present briefly our method for 3-D terrain registration and we show registration results. These results are obtained in an area in which registration would not be possible without the vegetation filtering and ground recovery algorithms described above.

4.1 Terrain Registration Method

The objective of terrain registration is to recover the vehicle position in the map by matching a local map from 3-D data from on-board sensors with 3-D data from the prior map. The core of our registration approach involves the computation of pose-invariant surface signatures in the neighborhood of feature points in both the robot and the prior map. Correspondences between the features are established by comparing the signatures, and the most consistent set of correspondences is retained for computing the registration transformation. An initial description of this class of approaches can be found in [12]. Initial extensions to terrain matching are described in [9]. Details of the current version of the map registration approaches are described in [20]. Importantly, this approach does not require accurate prior knowledge of vehicle pose. In fact, we have performed registration with 20 m initial error in robot position and $\pm 10^\circ$ error in orientation.

Key to the current approach is the automatic selection of interest points or feature points in the terrain maps. This is challenging because the points must be selected in a consistent manner between the aerial and ground data. The approach can be summarized as follows. Points are selected using three criteria computed from the configuration of the 3-D surface in the neighborhood of each point. The first criterion uses the configuration of range shadows in the vicinity of the point; essentially, points with low density of range shadows in their neighborhood are selected in priority. The second criterion evaluates the amount of variation of the terrain surface in the vicinity of each candidate point, so that only points with sufficiently curvy terrain are selected. Finally, the signatures are analyzed so that only those points whose signatures contain enough information are retained. Given a 3-D map constructed from the on-board sensors, the algorithm that combines the three criteria extracts a small set of points that are used as the feature points for matching. Andrew Johnson introduced a more elaborated but more expensive landmark point selection strategy in [11]. By contrast, our implementation runs on-board the mobile robot.

We present the performance evaluation below for the registration using the data described above. The key conclusion of the experiments is that vegetation filtering and ground recovery is critical to reliable terrain registration for two reasons. First, the vegetation generates very different data sets when viewed from the vantage point of the robot as opposed to the vantage point of the aerial sensor. This is true in obvious cases such as tree canopies, but this is also true of more mundane structures such as bushes. Second, this problem is compounded by the fact that selecting features for registration is based on the amount of variation in 3-D data around each point, which causes feature points to be selected near vegetation areas, the least reliable

to be for matching. For example, bushes and trees tend to be the most prominent 3-D structures in the scene, occluding other terrain surface features, such as rock formation and ledge. Furthermore, the last stage in registration involves minimizing the registration error between the two 3-D data sets, after transformation by the transformation computed from correspondences between feature points. Because of the stochastic nature of the data in vegetation areas, this minimization is, in practice, unreliable.

4.2 Example with the Yuma Data Set

Edge course Figure 1-(a) shows an example of air-ground registration. The ground area to be registered does not contain any vegetation, but the aerial data does. Without filtering the vegetation the points retained for performing the registration are mostly located in the wash, visible in the rear of the scene. The green surface represents the aerial data, it is $100\text{ m} \times 100\text{ m}$. The hill that is visible is roughly 10 m high and the ledge cliff is at most 3 m high. The data was collected during the demonstration performed at Yuma, Arizona, in May 2002.

Wash course In this second example the ground data contains bushes and trees. Aerial and ground data are represented as points clouds Figure 1-(b) and (c) respectively. The robot drove 9 m. The filtering method works correctly even in the presence of a steep slope, as shown in the side of the ledge Figure 1-(c). The vegetation points are plotted in color while the ground points are in white. The ground (aerial) data contains 50,000 (541,136) 3-D points, 9,583 (387,393) of them are classified as vegetation and 40,417 (153,721) as ground. The processing time is respectively 15 seconds and 117 seconds.

One added difficulty in this example is that, after filtering the vegetation in the robot data, the terrain surface contains a number of empty areas with no data, termed “range shadows”. This is because in many cases there is not enough data to recover the ground surface after filtering. This makes the feature selection more difficult because we cannot use the range shadows as a criterion to reject potential feature points. In the aerial data, on the other hand, the ground surface can be recovered more reliably.

Figure 1-(d) contains the aerial data without vegetation (in grey) registered with the ground data without vegetation (in red). The ground (aerial) mesh has a resolution of 0.27 m (1.2 m), it covers $40\text{ m} \times 25\text{ m}$ ($100\text{ m} \times 100\text{ m}$), and it contains 1,252 (9,000) vertices and 400 (637) of them are used to produce spin-images. The registration procedure lasts 2.3 s.

5 Path Planning

In this section, we describe the use of prior maps from high-resolution aerial data for path planning. We consider the problem of planning from a start position to a distant goal point through a set of intermediate waypoints. We evaluate different options

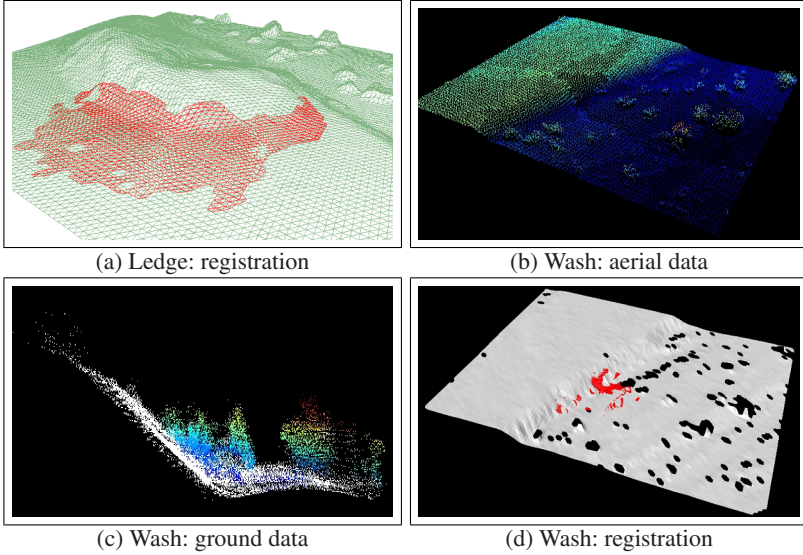


Fig. 1. Terrain registration with vegetation filtering, Yuma data set. (a) Registration example from the Ledge course; (b),(c),(d) registration example from the Wash course. In (b) the elevation is color coded from blue to red. In (c) the white area corresponds to the terrain surface, the colored points to the vegetation segmented. In (d), the grey represents the aerial data and the red mesh the ground data. Both in (c) and (d) the vegetation has been filtered.

for processing the prior map data and show the effect of vegetation filtering on the performance.

5.1 Traversability Maps

Our task is to provide a path between waypoints based on aerial LADAR data. Starting with the raw data we first divide the space into $1 \text{ m} \times 1 \text{ m}$ cells. Points are then segmented into two clusters: vegetation and ground. For each cell the ratio between the number of vegetation points and the total number of points defines a confidence criterion on the terrain reconstructed below the tree canopy. We called it the *vegetationness*. This new criterion allows the path planner to consider new trajectories, usually prohibited because vegetation is not treated a priori as an obstacle.

The traversability at each location in the map is evaluated using the standard approach of convolving a vehicle model with the elevation map. More precisely, using the ground points we compute height traversability maps (one every 45° in heading) as follows: 1) we interpolate the position of each tire on top of the ground surface, 2) we fit a plane and extract the current roll, pitch, and remaining ground clearance of the vehicle, 3) these values are remapped between 0 (non-traversable) and 1 (traversable) and then thresholded, using a sigmoid function, and the static performances of the vehicle (maximal frontal and side slopes, and the ground clearance). The final traversability value assigned to the cell is the value of

the least favorable of the three criterion. If one of the three criterion exceeds the robot's limits, the cell is marked as non-traversable.

5.2 Planner

To test our approach we implement a grid-based path planner to determine the path of least cost in our gridded map. The cost at each node is computed as follow :

$$C_{comb.}(\theta) = \frac{1}{(1 - C_{trav.}(\theta))^2} + \frac{1}{(1 - C_{veg.})^2} \quad (1)$$

With $C_{trav.}(\theta)$ the directional traversability of the cell, with θ the heading of the robot when it entered that cell ; $C_{veg.}$ is the *vegetationness* value of the cell; finally $C_{comb.}(\theta)$ is the cell cost used to determine the path in this specific cell along that direction.

We tested the relative influence of the traversability and *vegetationness* map on the results produced by the path planner with the Yuma data set. We performed 3 different tests using 47 pairs of starting/ending points, selected randomly in the scene. We computed a path for each of them using three different sets of maps: 1) the 8 directional traversability maps and the *vegetationness* map, 2) one directional map and the *vegetationness* map, 3) the 8 directional traversability maps only. Each path produced, 141 in total, has been evaluated visually using a high resolution aerial image (17 cm) as ground truth. The corresponding failure rates are 4.2 %, 19.2% and 4.2% for an average path lenght of 529 m, 535 m and 522 m respectively. In all cases, a valid path is known to exist and a failure is recorded whenever a path cannot be generated from the cost map.

The importance of using the directionality map is clear from the experiments. The role of the vegetation map is not critical on the Yuma data set because of the nature of the terrain: desert with tall bushes and trees with sparse vegetation. There is no continuous tree canopy as in the example presented in the next section.

5.3 Example with the APHill Data Set

Figure 2 presents an example of path obtained with our planner using the APHill data set. The terrain is made of tall trees producing a continuous canopy cover of the ground. The density of foliage is such that sufficient ground points were sensed to reconstruct the terrain below the canopy. Figure 2-(a) represents the *vegetationness* map. The path computed by the path planner, between 15 waypoints, is overlaid in black. The total cumulative distance is 152 m. Figure 2-(b) shows the traversability map computed after filtering. The green areas are traversable regions and the dark red terrain is non-traversable. Blue points are part of a water pond. Without filtering the vegetation, only the dirt roads appear traversable. The interest of the method is explicit on this example. During the demonstration a different path planner, developed by NIST, was used to produce a similar path passing through the same waypoints [2]. The robot actually navigated autonomously along this path, avoiding local obstacle, such as small obstacle and overhanging branches, not perceived in the aerial LADAR data.

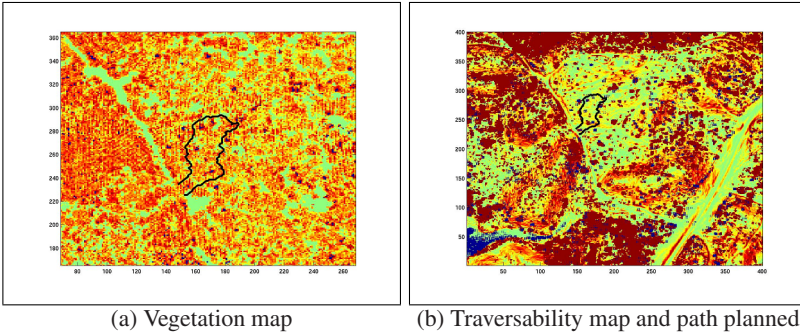


Fig. 2. Path planning and vegetation filtering, AP Hill data set. (a) Vegetation map with a color scale from green to red for no vegetation to high density vegetation, in black the path planned. (b) Traversability map with from green to red traversable to non-traversable terrains

6 Conclusion

In this article, we presented results on the use of aerial LADAR data for 3-D terrain registration and path planning obtained during two different field demonstrations, using a ground mobile robot. We showed that because vegetation hides potential obstacles, masks the ground terrain features, and introduces artifacts which mislead point selection strategies, it is a major problem for mobile robot navigation in natural environment. We presented the methods used to filter vegetation both for ground and aerial LADAR data. Once the ground terrain is recovered, we have been able to produce ground-ground / air-ground 3-D terrain registration for terrains in which registration would not be possible without vegetation filtering. We proposed to use ground terrain recovery below the canopy for path planning. We demonstrated with an autonomous ground robot that this new approach is viable.

Although encouraging, these results are still limited, and work still needs to be done to further evaluate performance and to extend the approach to reach operational performance. Key directions of future work include: systematic evaluation of our approach with correlation to terrain ground truth, making the vegetation filtering method invariant to the terrain slope, and improvement of path planning by taking into account the clearance between the terrain and the bottom of the canopy.

Acknowledgments

This project was funded by DARPA under the PerceptOR program, under subcontract to General Dynamics Robotic Systems. This work would not have been possible without the help of William Klarquist from PerceptTEK.

References

1. E.P. Baltsavias, "Airborne laser scanning: existing systems and firms and other resources", *ISPR Journal of Photogrammetry & Remote Sensing*, 1999, vol 54

2. S. Balakirsky and A. Lacaze, "World modeling and behavior generation for autonomous ground vehicle" *IEEE International Conference on Robotics and Automation*, 2000
3. A. Castano and L. Matthies, "Foliage Discrimination using a Rotating Ladar", *International Conference on Robotics and Automation*, 2003
4. D. Coombs, K. Murphy, A. Lacaze and S. Legowik, "Driving autonomously offroad up to 36 km/hr", *IEEE Intelligent Vehicles Symposium*, 2000
5. W. Eckstein and O. Munkelt, "Extrating objects from digital terrain models", *Remote Sensing and Reconstruction for Three-Dimensional Objects and scenes*, SPIE Proceedings vol 2572, 1995
6. S.O. Elberink and H.G. Mass, "The use of anisotropic height texture measures for the segmentation of airborne laser scanner data", *International Archives of Photogrammetry and Remote Sensing*, vol XXXIII
7. M. Hebert and N. Vandapel, "Terrain Classification Techniques from LADAR Data for Autonomous Navigation", *Collaborative Technology Alliance Conference*, 2003
8. J. Hyypä, O. Kelle, M. Lehtikoinen and M. Inkinen, "A Segmentation-based method to retrieve stem volume estimates from 3-D tree height models produce by laser scanners", *IEEE Transaction on Geoscience and Remote Sensing*, vol 39, no 5, May 2000
9. D. Huber and M. Hebert, "A new approach to 3-D terrain mapping", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999
10. J. Huang, A.B. Lee and D. Mumford, "Statistics of range images", *IEEE International Conference on Computer Vision and Pattern Recognition*, 2000
11. A. Johnson, "Surface landmark selection and matching in natural terrain", *IEEE International Conference on Computer Vision and Pattern Recognition*, 2000
12. A. Johnson, "Spin-Images: a representation for 3-D surface matching", *Ph.D Thesis, Carnegie Mellon University*, 1997
13. K. Krauss and N. Pfeifer, "Determination of terrain models in wooded areas with airborne laser scanner data" *ISPRS Journal of Photogrammetry & Remote Sensing*, vol 53
14. M.A. Lefsky et al., "Lidar remote sensing of the canopy structure and biophysical properties of Douglas-FirWestern Hemlock forests" *Remote Sensing Environment*, vol 70, 1999
15. P. Lohmann, A. Koch and M. Shaeffer, "Approaches to the filtering of laser scanner data", *International Archives of Photogrammetry and Remote Sensing* vol XXXIII, 2000
16. J. Macedo, R. Manduchi and L. Matthies, "Laser-based discrimination of grass from obstacles for autonomous navigation", *International Symposium on Experimental Robotics*, 2000
17. B. Petzold, P. Reiss and W. Stossel, "Laser scanning surveying and mapping agencies are using a new technique for the derivation of the digital terrain models", *ISPRS Journal of Photogrammetry and Remote Sensing*, 2000
18. M. Shneier et al., "A Repository of sensor data for autonomous driving research", *SPIE Aerosense Conference*, 2003
19. G. Sithole, "Filtering of laser altimetry data using a slope adaptive filter", *ISPRS workshop on Land Surface Mapping and Characterization using laser altimetry*, 2001
20. N. Vandapel and M. hebert, "3-D rover localization in airborne LADAR data", *International Symposium on Experimental Robotics*, 2002

Autonomous Detection of Untraversability of the Path on Rough Terrain for the Remote Controlled Mobile Robots

Kazuma Hashimoto¹ and Shin'ichi Yuta²

¹ Denso Corp. *

KAZUMA.HASHIMOTO@denso.co.jp

² University of Tsukuba

yuta@roboken.esys.tsukuba.ac.jp

<http://www.roboken.esys.tsukuba.ac.jp>

Abstract. The mobile robot which traverses on a rough terrain, should have an ability to recognize the shape of the ground surface and to examine the traversability by itself. Even if the robot is remotely controlled by operator, such an ability is still very important to save the operator's load and to keep safety. For this purpose, the robot measures the front ground, and generate the local elevation map which represents the area where the robot is going to pass, and examines that the wheels can pass through and the bottom surface of robot body dose not contact with ground. This paper reports a simple method of traversability test for the wheeled mobile robot and show an experimental system with some results.

1 Introduction

We are investigating on the mobile robots, which works based on the combination of remote operation and autonomous control on the rough terrain. The lunar rover is one of the examples. For such kind of vehicles, the operator remotely gives a path or a sequence of sub-goals, which the robot should track to. However, since the environment is irregular and not known well, the operator may not always give a safe path. Therefore, the robot has to keep its safety by itself, even while it moves according to the operator's commands.

For this purpose, the robot should continuously observe its front ground surface, and check its traversability. When it judges not safe to traverse, it should stop and wait a new command from the operator.

This paper reports a simple method of traversability test for the wheeled mobile robot by itself, and show an experimental system with some results.

2 Traversability Test

First, we propose a basic framework of the autonomous traversability test of the remote controlled mobile robots.

* This research was conducted, when the first author was at graduate school in Science and Engineering, University of Tsukuba.

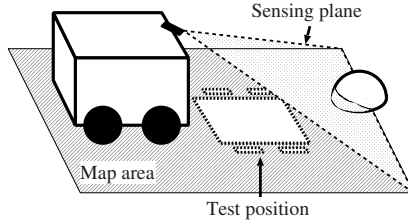


Fig. 1. Observing front to test traversability by a mobile robot

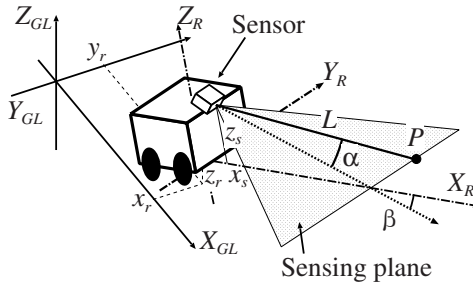


Fig. 2. Sensing front

The traversability of the front ground is tested by the following steps, when the robot moves on the rough terrain:

1. While moving forwards, the robot always observe a single lateral plane with a certain inclination in front of the robot, and measure the range to the ground surface on this plane from the robot (fig 1).
2. The robot continuously generates a local elevation map of the ground surface in the area of the front of the robot, by integrating measured data.
3. The robot examines the possibility of traveling over of the given path at a certain distant front of the robot.

2.1 Sensing Front

The robot uses a line distance sensor, which is attached on the front top of the robot body, and heading towards the front ground. It measures the range to the ground surface on the single lateral line ahead of the robot. They are converted to the position data of ground surface on the global coordinates frame, by calculating with the robot position and posture information, and stored to generate a two-dimensional elevation map.

Let denote the global coordinate frame X_{GL}, Y_{GL}, Z_{GL} , and the robot coordinate frame X_R, Y_R, Z_R , of which, the origin is center of the robot bottom surface and

X axis is towards the robot front (figure 2). The line range sensor is attached on the robot at $(x_s, 0, z_s)$ of robot coordinate, towards the front ground with the slant angle β rotated around Y axis. The sensor measures the distance L to the ground point, which is shown P in figure 2, for each direction α .

When the robot position in global coordinate is given as (x_r, y_r, z_r) and robot posture is given as (ϕ, θ, ψ) by Z-Y-X Euler angle, the position of point P is calculated by

$$\begin{bmatrix} P_{X_{GL}} \\ P_{Y_{GL}} \\ P_{Z_{GL}} \end{bmatrix} = \begin{bmatrix} T_{ij} \end{bmatrix} \begin{bmatrix} L \cos \alpha \\ L \sin \alpha \\ 1 \end{bmatrix} \quad (1)$$

where,

$$T_{11} = \cos \psi \cos \theta \cos \beta - \sin \beta (\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) \quad (2)$$

$$T_{12} = -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi \quad (3)$$

$$T_{13} = x_r + x_s \cos \psi \cos \theta + z_s (\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) \quad (4)$$

$$T_{21} = \sin \psi \cos \theta \cos \beta \sin \beta (-\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi) \quad (5)$$

$$T_{22} = \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi \quad (6)$$

$$T_{23} = y_r + x_s \sin \psi \cos \theta + z_s (-\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi) \quad (7)$$

$$T_{31} = -\sin \theta \cos \beta - \cos \theta \cos \phi \sin \beta \quad (8)$$

$$T_{32} = \cos \theta \sin \phi \quad (9)$$

$$T_{33} = z_r - x_s \sin \theta + z_s \cos \theta \cos \phi \quad (10)$$

As a sensor for such range measurement, the stereo camera, mechanically scanning laser range sensor, or, light-plane intersecting method can be considered.

The robot posture (ϕ, θ, ψ) can be measured by the accumulation of angular velocities obtained by the gyro sensor, and/or using the direction to the reference landmarks. The robot position is estimated by the accumulation of the motion data with its posture information.

2.2 Making Elevation Map

The robot always collects data on the front ground while it travels, and integrates them to generate a local elevation map. The elevation map can be represented by height of each ground surface on the global two dimensional plane [1]. To generate the map, a two dimensional array of cells are defined at first, and when the robot gets three dimensional position information of the points on the ground surface, it registers them as a height information of the corresponding cell to generate the elevation map. So, by continuous measurement while the robot moves forward, the each cell of the map will have an elevation information.

In case a grid cell has multiple height data, the resultant height of the cell is decided as;

1. When the difference of multiple height information on a cell is small, the average is adopted as the height of the grid.
2. When it has the large difference among the height data, the maximum value is adopted.

And, in case the grid cell has no height data, it should be treated as;

1. When all neighboring grids have height data, the their average is used for this grid.
2. When there is a grid without height data around it, this grid point remains as unknown.

2.3 Test of Traversability

The robot should test the traversability of its front ground using the generated elevation map and the robot path given by operator or command. For the wheel type robot, the traversability can be tested by;

1. Wheel traversability : Whether the wheels can pass over, and
2. Body traversability : Whether the bottom surface of the robot body can keep no contact with the ground, while it moves on the the planned path.

Wheel traversability tests the driving ability of the robot and its wheels on the step or the slope on the ground. For test of this condition, the sequence of ground height information along the path of each wheel is extracted, and the position with a certain distance ahead from the current wheel position is examined at each time instant. Figure 3 shows an example of the side view of height sequence and an examining point. The points before the examining one are traversable, since the robot has already tested them. Let assume that the maximum height of the step which can be passed over by this wheel is \mathcal{H} . In this case, the wheel traversability at the examining point is tested by the following steps.

- Draw a level line, which is \mathcal{H} lower than the examine point in elevation on the side view diagram of height sequence.
- Draw two virtual wheels which contacts the line and examining point.
- If there is any ground point inside in the first virtual wheel, it means that the robot can arrive at this point. And if there are no, this means that the wheel can not pass this point over.

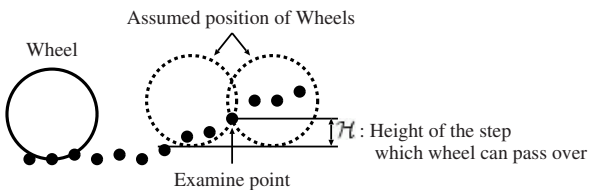


Fig. 3. Examine whether wheels can pass over

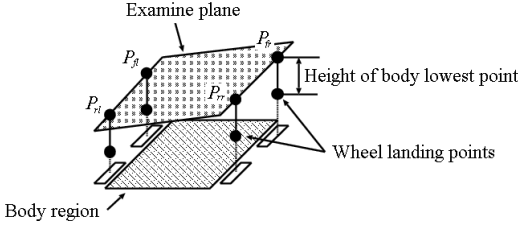


Fig. 4. Examine the contact of body with the ground

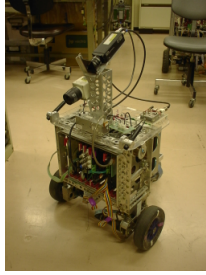


Fig. 5. "Yamabico-Navi"

- In case there are no ground points inside in the other virtual wheel, it means that the robot can not go back to this point from the opposite direction, and it is also concluded not safe to go over.

The body traversability tests the collision of the lower body surface with ground. The examination is done for the position where is a certain distant ahead of current robot position. Figure 4 shows projective figure of the robot at examine position. As the examination, calculate the ground contact points of the robot wheels on the expected robot path at this distant ahead at first. And, calculate a linear plane which approximates the lower body surface of the robot at this position, using the elevation of all wheel points. Then, the elevation of the cells on the map in the region of the robot are compared with this plane. As the result, the test of the contact of the robot body with the ground surface is performed.

3 Implementation of Experimental System

3.1 Mobile Robot Platform "Yamabico-Navi"

We implemented an experimental system on our mobile robot platform Yamabico-Navi, which is shown in figure 5.

Yamabico-Navi is equipped with two independent driving wheels with a diameter of $150mm$ and a castor wheel with a diameter of $110mm$. The body size is about $450mm(W) \times 450mm(D) \times 700mm(H)$, and the weight is about $12kgs$.

Yamabico-Navi has a functionally distributed controller with three control modules. First control module is for the vehicle control and driving wheels, and second module is for vision processing which is used as the range sensor. The third module is the master controller, on which the elevation map is generated and traversability is tested, as well as the remote control of the robot motion is also executed. Each module is equipped with an independent CPU (T805, 20MHz) [2].

3.2 Sensing Front

Range measurement In this system, line range sensor is realized by light plane intersecting method with laser slit projector and CCD image sensor[3]. Figure 6 shows a geometrical arrangement of the sensor. We defined a sensor coordinates on the laser slit projector as X axis parallel to its heading, and X-Y plane is corresponded to the laser plane. And we equipped a CCD image sensor on (x_c, y_c, z_c) in the sensor coordinate, heading towards the X axis, with slant angle θ_c rotated on Y axis. When P is a one of the laser reflected point, the distance L and the direction α to point P from an origin of the sensor coordinate is calculated by

$$L = \sqrt{x^2 + y^2} \quad (11)$$

$$\alpha = \tan^{-1} \frac{y}{x} \quad (12)$$

$$x = \frac{z_c}{\tan(\theta_c - \tan^{-1} \frac{v}{F})} + x_c \quad (13)$$

$$y = h \sqrt{\frac{x^2 + z_c^2}{F^2 + v^2}} + y_c \quad (14)$$

Where h and v is the coordinate of the reflect point in imaging plane, and F is the focal distance between the pinhole and the imaging plane.

The parameters of the implemented system were : (figure 7)

$$x_s = 74mm$$

$$z_s = 420mm$$

$$\beta = 24.5degree$$

$$x_c = -104mm$$

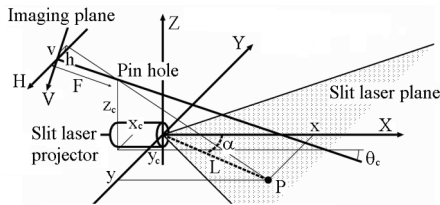


Fig. 6. A range sensor by light plane intersecting method

data. And the height of the grid which it has no height data is decided when the robot using the grid data.

In the experiment, the robot moves in the speed of $3\text{cm}/\text{second}$, so that the robot can measure the front ground every 1cm by the simple robot controller.

3.4 Test of Traversability

In experimental system, the examine position is defined as 100mm front from the current robot position, both for the examine of wheels passing over and the contact of body with the ground. Yamabico-Navi is a tricycle, so examine plane for the body traversability is calculated using the elevation of three wheels points. The height of the step, which the wheel can cross over, is 17mm , and the height of the body lowest point is 40mm .

3.5 Remote Control of the Robot

The robot is controlled remotely based on the transmitted images [4]. The camera observing front direction is mounted on Yamabico-Navi and images are transmitted to the operator station and shown on the display of PC. The operator gives the robot path on the displayed image using the pointing device, so that the robot knows the requested motion, which is given relatively to the robot position and posture when the image is taken.

4 Experimental Results and Discussion

We made several experiments. As, an example of the experiments, Yamabico moved on the environment shown as figure 8, where, A through E is safe steps, and F is un-safe step.

Yamabico was expected to move and pass over the steps A through E , and to stop before the F step autonomously. As the result, it succeeded in about 80% cases. In the cases of 20%, Yamabico had judged not safe at the steps A through E and stopped.

Figure 9 shows an example of the elevation map which is made by the robot. On this elevation map, height data were larger as X positions become larger. This is caused by the error of posture estimation. Figure 10 shows the recorded robot pitch angles in this experiment.

One of the reasons of the failure was the error on the elevation map caused by the multi reflected laser from the floor surface. The reliability of sensor data was the most critical problem on our experiments.

As a conclusion, we could check the validity of an autonomous traversability test method proposed in this paper. In an experimental system, the calculated posture has a drifted errors. So, when the robot works for a long term, the suppress a drifted error will be necessary. And for improvement of reliability of traversability test, the accuracy of front sensing will be the most important issue.

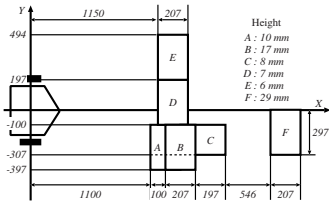


Fig. 8. An environment of experiment

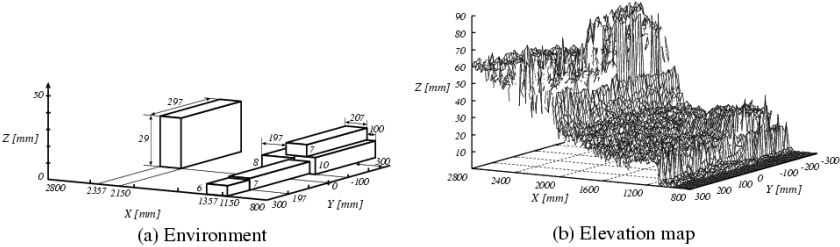


Fig. 9. An elevation map made by robot

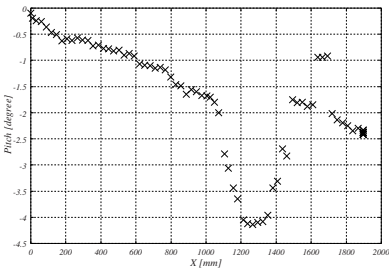


Fig. 10. Pitch angle of the robot measured by gyro sensor while the experiment

We also conducted the similar experiments in the outdoor environment at evening (figure 11). Yamabico could also detect and stop before the large step.

5 Conclusion

In this paper, we report a simple method of the traversability test of remote controlled mobile robot which moves on the rough terrain. Our method is very simple, so that only a T805 processor could perform to make an elevation map and judge the traversability in our experimental system. We implemented an experimental system, and tested the proposed method by experiments. Even though the implemented experimental system was very simple and executed on a small computer, and the range sensor was not accurate enough, the experimental system showed the effectiveness of the proposed autonomous test method.



Fig. 11. An experimental robot in tested environment

References

1. T.Yoshimitsu and T.Kubota and I.Nakatani, "Path Planning for Exploration Rovers over Natural Terrain Described by Digital Elevation Map," *Journal of the Robotics Society of Japan*, vol. 18, no. 7, pp. 1019–1025, 2000. in Japanese.
2. T.Miyai and S.Yuta, "Design and Implementation of Distributed Controller and its Operating System for Autonomous Mobile Robot Platform," *International Conference on Field and Service Robotics (FSR'97)*, Canberra, Australia, pp. 342–347, 1997.
3. S.Yuta and S.Suzuki and Y.Saito and S.Iida, "Implementation of an Active Optical Range Sensor Using Laser Slit for In-Door Intelligent Mobile Robot," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS) '91*, Osaka, Japan, pp. 415–420, 1991.
4. T.Sekimoto and T.Tsubouchi and S.Yuta, "A Simple Driving Device for a Vehicle - Implementation and Evaluation," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS) '97*, Grenoble, France, pp. 147–154, 1997.
5. K.Hashimoto and S.Yuta, "An Automatic Judgement of a Traversability for a Teleoperated Mobile Robot," *Proceeding of the 20th annual conference of RSJ*, Osaka, Japan, 1J34, 2002. in Japanese.

Mobile Robot Navigation Based on DGPS and Odometry in Campus Environment [★]

Kazunori Ohno¹, Takashi Tsubouchi¹, Bunji Shigematsu², Shoichi Maeyama³ and Shin'ichi Yuta¹

¹ Intelligent Robot Lab., Univ. of Tsukuba, 1-1-1 Tennoudai Tsukuba 305-8573 JAPAN.
<http://www.roboken.esys.tsukuba.ac.jp/>

² Penta-Ocean Construction CO. LTD.

³ Osaka Electro-Communication Univ.

Abstract. This is a technical report on an outdoor navigation for autonomous mobile robot based on DGPS and odometry positioning data. The robot position is estimated by fusion of DGPS and odometry. However, DGPS measurement data suffer from multi-path or other effects near high building and high trees. Thus, it is necessary to pick up only reliable and accurate DGPS measurement data when the robot position is corrected by data fusion of DGPS and odometry. In this paper, the authors propose a selection method of erroneous DGPS measurement data and a rule of data fusion for robot position correction. Finally, the authors equip a mobile robot with the proposed correction method and the robot autonomously navigates in a campus walkway.

1 Introduction

Outdoor navigation is an important issue for autonomous mobile robotics. The authors aim at map based outdoor navigation of a mobile robot in campus walkway using GPS (Global Positioning System) and odometry. Figure 1 illustrates an autonomous mobile robot developed in the laboratory of the authors. Figure 2 illustrates a campus plan of University of Tsukuba and experimental environment of the authors. When a mobile robot navigates in the environment, its position is fundamentally obtained by odometry. However, error of the position increases as the robot moves because the error is cumulative for odometry. On the other hand, GPS does not have such cumulative error and is expected to cancel the cumulative error of odometry. There are many reports [1–4] on map based outdoor navigation using RTK-GPS (Real Time Kinematics GPS), because it is well known that RTK-GPS yields only several centimeters order position errors. Following these issues, the authors examined whether RTK-GPS is actually useful to measure the robot position even when the robot is moving in the walkway environment among buildings or trees. Trimble RTK-GPS receiver (4000SSI) was used in the examination. However, it was found by the authors that RTK-GPS was not suitable for moving vehicles because positioning initialization was often occurred as a result of “cycle-slip”. Once RTK-GPS begins initialization, the receiver must stand still until the initialization is finished.

[★] This work was partly supported by JSPS Grant-in-Aid for Scientific Research 13560130.

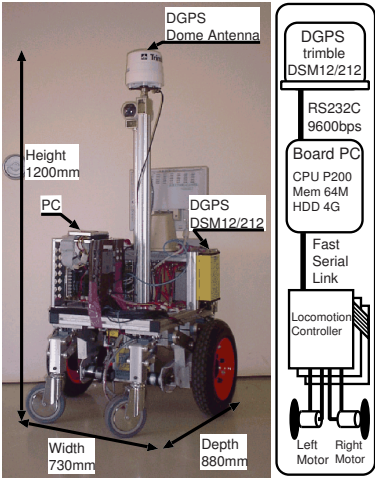


Fig. 1. Yamabico YM2000 and its system configuration

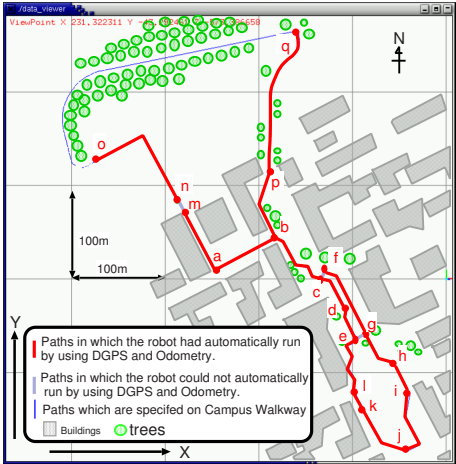


Fig. 2. Campus plan and paths in which the authors examined outdoor navigation

In addition, the RTK-GPS receiver could not sometimes measure its position even when the receiver could watch enough number of GPS satellite, because the receiver could not get necessary informations for position adjustment to maintain high accuracy in RTK method via MCA (Multi Channel Access radio system)¹. Therefore, the authors prefer to use DGPS (Differential GPS) because it can go on to measure its position as far as radio waves from the enough number of the satellites are received and it does not suffer from “cycle-slip” phenomenon, even though its measurement accuracy is lower than RTK-GPS. The authors use DGPS

¹ MCA has access limitation because several user shares one channel. So that, a lot of users must wait for the MCA channel allocated.

(Trimble DSM12/212) receiver which gets adjustment informations via the LF band broadcast channel. It can reduce the measurement error within one or at most several meters from original GPS data.

The mobile robot is equipped with DGPS receiver and odometry. Proposed navigation method is as follows:

1. The robot is placed at the start point, and its location and orientation are given to the robot.
2. The robot is given a path from start to goal which is a list of passing-through-points on the X-Y plane. The authors do not care about a height. The target path for the robot to follow is a sequence of the line segments which connect adjoining points.
3. The robot controls its driving wheels moving along with the target path.
4. While the robot moves, DGPS measurement data $P_{gps}[t] = (x_{gps}, y_{gps}, \theta_{gps})^T$ is obtained in real time. Its reliability is judged by using estimated robot position $P[t] = (x_o, y_o, \theta_o)^T$ based on odometry and its error covariance matrix $\Sigma_P[t]$. If the GPS measurement data is regarded as reliable, the framework of EKF (Extended Karman Filter) is used for fusion of GPS measurement data and odometry data.
5. Iterate 3 and 4 until the robot reaches a goal point.

The authors measured the location of passing-through-point on a desired path for the robot in campus walkway by using a RTK-GPS receiver and made a path map (item 2) [6]. A control of path following based on odometry (item 3) was accomplished by using location control system “Spur” which is developed in the laboratory of the authors [7].

In this paper, the authors propose a selection method of reliable DGPS measurement data and rules for fusion of DGPS measurement data and odometry. The framework of EKF is used for the fusion. The DGPS measurement data sometimes have large error because of multi-path near buildings [5]. Thus, it is necessary to inhibit the fusion if DGPS measurement is considered to be unreliable or to have large error. In this paper, the authors propose a method which detects erroneous DGPS measurement data on a basis of odometry and selects a suitable position correction method (Section 3). Lastly, this paper reports an implementation of proposed correction method on a mobile robot and the experiment of outdoor navigation in our campus walkway (Section 4).

2 Positioning Sensor

2.1 DGPS

In Fig. 1, DGPS (Trimble DSM12/212) receiver which is used by the present authors appears. The Differential GPS position measurement principle suppressed its measurement error within one or at most several meters, which is improved from the elementary GPS measurement.

The robot system is illustrated on the right part of Fig. 1. The data from DGPS receiver is sent to a computer via RS232C with 9600[bps]. The output data format

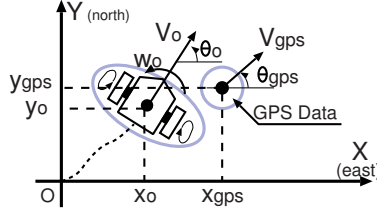


Fig. 3. Robot position estimated with odometry and GPS data

is so-called NMEA-0183 which offers a series of characters through a RS232C communication channel. Table 1 shows the part of NMEA sentences the authors are using. These are obtained from DGPS receiver at every second. By analyzing the GGA sentence, we can obtain latitude, longitude and height with respect to the WGS-84 geographic coordinate system. These data are transformed into the position $(x_{gps}[t][m], y_{gps}[t][m], z_{gps}[t][m])$ which is based on an ellipsoidal surface. The origin of this coordinate is one of Kanto 9 system whose origin is placed at $\{E\ 139^{\circ}\ 50'\ 0''.000, N\ 36^{\circ}\ 0'\ 0''.000\}$. Y-axis of the system directs north. X-axis directs east. The Z-axis is perpendicular to the X-Y plane. On the other hand, by analyzing the VTG sentence, we can obtain heading direction $(\theta_{gps}[t][degree])$ over X-Y plane from DGPS receiver. $\theta_{gps}[t]$ is an orientation which is represented counterclockwise within $\pm 180[degree]$ from X axis. The authors define $P_{gps}[t] = (x_{gps}[t], y_{gps}[t], \theta_{gps}[t])^T$ as a robot position which is measured by DGPS. GPS position $(x_{gps}[t], y_{gps}[t])$ is measured with TOF (Time Of Flight) of GPS radio waves, GPS heading direction $\theta_{gps}[t]$ is calculated with Doppler shift of the GPS radio waves in principle.

The authors recorded the DGPS measurement data $P_{gps}[t]$ in several walkway environment of the university campus of the authors and observed the tendency of the measurement data [5]. Figure 4 illustrates typical DGPS measurement data in the campus. In the Figure, each black point represents DGPS measured position $(x_{gps}[t], y_{gps}[t])$ and a bar from the point represents heading direction $\theta_{gps}[t]$. From Fig. 4, we can find that DGPS measurement data tend to include large error near high building. Only reliable data must be used for fusion or cumulative error compensation for odometry. Discrimination of obtained data from DGPS is necessary. In addition, it is found there are cases that error of DGPS heading direction data are small even when error of the DGPS position data are large. These phenomena may arise from the difference of measurement principle between position and heading direction measurements. Therefore, the authors assume that the DGPS position data and its heading direction data are independent. The reliability of DGPS position data and its heading direction data will be discriminated separately.

2.2 Odometry

Odometry is a sensor which estimates both position and orientation of the robot by integrating the number of left and right driving wheel rotations. Coordinate system of the odometry is defined with the same one to the DGPS (Fig. 3). The robot position estimated by the odometry is defined as $P[t] = (x_o[t][m], y_o[t][m], \theta_o[t][degree])^T$.

Table 1. NMEA sentences

Sentence	Contents
GGA	Latitude, Longitude, Height, Quality (Differential or not), Number of satellites in use
GSA	Mode (2D, 3D), Satellite's ID used in position fix, PDOP, HDOP, VDOP
VTG	Speed over ground, Heading direction

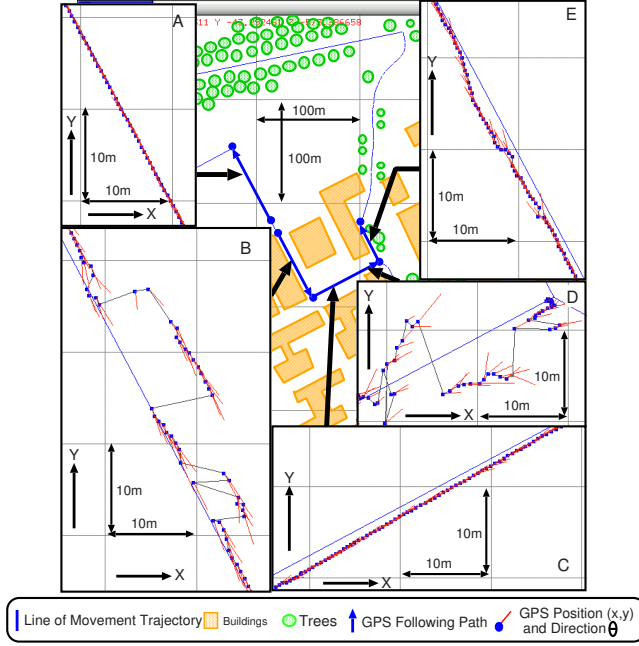


Fig. 4. Typical DGPS data near by structures: A illustrates DGPS measurement data recorded on a walkway in the open space. B, C, D and E illustrate DGPS measurement data on a walkway among high buildings.

$\Sigma_P[t]$ represents the error covariance matrix. $P[t]$ and $\Sigma_P[t]$ are formulated into (1) and (2) in the framework of EKF [8].

$$P[t + \tau] = P[t] + \tau \begin{pmatrix} V[t] \cos(\theta[t]) \\ V[t] \sin(\theta[t]) \\ \omega[t] \end{pmatrix}, \quad (1)$$

$$\Sigma_P[t + \tau] = J \Sigma_P[t] J^T + K[t] \Sigma_V K[t]^T + \Sigma_N, \quad (2)$$

$$\Sigma_P[t] = \begin{pmatrix} x_o[t]^2 & xy_o[t] & x\theta_o[t] \\ xy_o[t] & y_o[t]^2 & y\theta_o[t] \\ x\theta_o[t] & y\theta_o[t] & \theta_o[t]^2 \end{pmatrix},$$

where τ is a sampling interval. $V[t]$, $\theta[t]$ and $\omega[t]$ represent robot velocity, orientation and angular velocity respectively. $J[T]$ is Jacobian of $P[t]$ with respect to x , y and θ . $K[T]$ is that of $P[t]$ with respect to V and θ . Σ_V is measurement error of odometry. Σ_N is truncation error. At every sampling time, estimated robot position and its error matrix is updated by using (1) and (2).

3 Elimination of Unreliable DGPS Data and Fusion

The authors propose that reliability of DGPS measurement data can be judged based on odometry because cumulative error of odometry is small in short running interval. Likelihood of DGPS measurement data l is calculated by using the estimated robot position $P[t]$ and its error covariance matrix $\Sigma_P[t]$. The DGPS measurement data is considered as reliable if l is small. The robot position which is estimated by odometry is modified by using only the reliable DGPS measurement data.

DGPS receiver outputs position $X_{gps} = (x_{gps}, y_{gps})^T$ and heading direction θ_{gps} . The authors assume that there is no correlation between (x_{gps}, y_{gps}) and θ_{gps} because these measurement principles are different each other. Let us denote likelihood of DGPS position as l_{xy} and likelihood of DGPS orientation as l_θ . l_{xy} and l_θ are calculated independently using Mahalanobis distance (Eq. (3) and (4)) with respect to $P[t]$ and its estimated error covariance matrix $\Sigma_P[t]$.

$$2l_{xy}[t]^2 = (X_{gps}[t] - X_o[t])^T (\Sigma_o[t] + \Sigma_{gps})^{-1} (X_{gps}[t] - X_o[t]) \quad (3)$$

$$X_o[t] = (x_o[t], y_o[t])^T, \quad \Sigma_o[t] = \begin{bmatrix} x_o[t]^2 & x_y o[t] \\ x_y o[t] & y_o[t]^2 \end{bmatrix}$$

$$\Sigma_{gps}[t] = \begin{bmatrix} 2_{x_{gps}} & 0 \\ 0 & 2_{y_{gps}} \end{bmatrix}$$

$$2l_\theta[t]^2 = (\theta_{gps}[t] - \theta_o[t]) (\frac{2}{\theta_o[t]} + \frac{2}{\theta_{gps}})^{-1} (\theta_{gps}[t] - \theta_o[t]) \quad (4)$$

$X_o[t], \Sigma_o[t], \theta_o[t], \sigma_{\theta_o}[t]$ are subset of $P[t]$ and $\Sigma_P[t]$ which is obtained from the odometry. $\Sigma_{gps}, \sigma_{\theta_{gps}}$ are constant values which are given in advance based on the observation of DGPS measurement error. In Eq. (3) and (4), likelihoods $l_{xy}[t]$ and $l_\theta[t]$ are calculated in consideration of DGPS measurement error because its error is large and can't be ignored for accurate judgment [9]. Each measurement error of DGPS position and orientation data is regarded as small when each likelihood $l_{xy}[t], l_\theta[t]$ is lower than predefined threshold $l_{xy_thresh}, l_\theta_thresh$. When the DGPS measurement error is small, the robot position which is estimated with odometry is corrected by using an equation of maximum likelihood estimation (Eq. (5)) in the framework of EKF [8].

$$\hat{P}_f[t] = P[t] + \Sigma_f[t] W_{gps}^{-1} (P_{gps}[t] - P[t]) \quad (5)$$

$$\Sigma_f[t] = (\Sigma_P[t]^{-1} + W_{gps}^{-1})^{-1}$$

$\hat{P}_f[t]$ and $\Sigma_f[t]$ represent corrected robot position and its error covariance matrix after the fusion. $P[t]$ and $\Sigma_P[t]$ are the robot position and its error matrix which are estimated with odometry. $P_{gps}[t]$ and W_{gps} are GPS measurement data and its measurement error. For correcting the robot position which is estimated by odometry, each $P[t]$ and $\Sigma_P[t]$ is innovated with $\hat{P}_f[t]$ and $\Sigma_f[t]$. In the proposed method for discriminating DGPS measurement, there are four combinations for these likelihoods $l_{xy}[t]$ and $l_\theta[t]$ in all. According to every combination, $P_{gps}[t]$ and W_{gps}^{-1} are defined for every case. The robot position is correctly modified by calculating Eq. (5) with the $P_{gps}[t]$ and W_{gps}^{-1} as follows.

1. case of $l_{xy}[t] < l_{xy \text{ thresh}}$ and $l_\theta[t] < l_\theta \text{ thresh}$

$P_{gps}[t]$ and W_{gps}^{-1} are defined based on DGPS position and orientation data.

$$P_{gps}[t] = (x_{gps}[t], y_{gps}[t], \theta_{gps}[t])^T, \quad W_{gps}^{-1} = \begin{pmatrix} \frac{2}{x_{gps}}^{-1} & 0 & 0 \\ 0 & \frac{2}{y_{gps}}^{-1} & 0 \\ 0 & 0 & \frac{2}{\theta_{gps}}^{-1} \end{pmatrix}.$$

2. case of $l_{xy}[t] < l_{xy \text{ thresh}}$ and $l_\theta[t] > l_\theta \text{ thresh}$

$P_{gps}[t]$ and W_{gps}^{-1} are defined based only on DGPS position data.

$$P_{gps}[t] = (x_{gps}[t], y_{gps}[t], 0)^T, \quad W_{gps}^{-1} = \begin{pmatrix} \frac{2}{x_{gps}}^{-1} & 0 & 0 \\ 0 & \frac{2}{y_{gps}}^{-1} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

3. case of $l_{xy}[t] > l_{xy \text{ thresh}}$ and $l_\theta[t] < l_\theta \text{ thresh}$

$P_{gps}[t]$ and W_{gps}^{-1} are defined based only on DGPS orientation data.

$$P_{gps}[t] = (0, 0, \theta_{gps}[t])^T, \quad W_{gps}^{-1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{2}{\theta_{gps}}^{-1} \end{pmatrix}.$$

4. case of $l_{xy}[t] > l_{xy \text{ thresh}}$ and $l_\theta[t] > l_\theta \text{ thresh}$

The estimated robot position is not corrected using DGPS measurement data.

Following this proposed method, reliable DGPS measurement data is judged from odometry and cumulative error of odometry is corrected using only reliable DGPS measurement data.

4 Experiment

Two different kind of experiments are presented in this Section. For the first experiment, position of robot which was pushed by human along a path was estimated and modified with the proposed correction method. For the second one, the robot autonomously navigated in campus walkways by using a navigation method based on the proposed correction method.

4.1 Evaluation of Proposed Correction Method

To verify proposed correction method, the robot is moved on the path accurately by hand and the robot position is maintained by the proposed method. The mobile robot was pushed by human accurately along the path $[a \rightarrow b \rightarrow c \rightarrow d]$ in Fig. 2. During the motion, even though the robot was pushed by human, the odometry function was activated. Figure 5 A illustrates the robot position and its error ellipse which were estimated by only odometry without position correction. The error ellipse in Fig. 5 A represents a contour of probability and was drawn by X which satisfies (6).

$$(X - X_o[t])^T \Sigma_o^{-1} (X - X_o[t]) = 1 \quad (6)$$

Actual position of the mobile robot was at the point d in Fig. 5 A. Estimated robot position by only odometry was over ten meters away from the point d . Figure 5

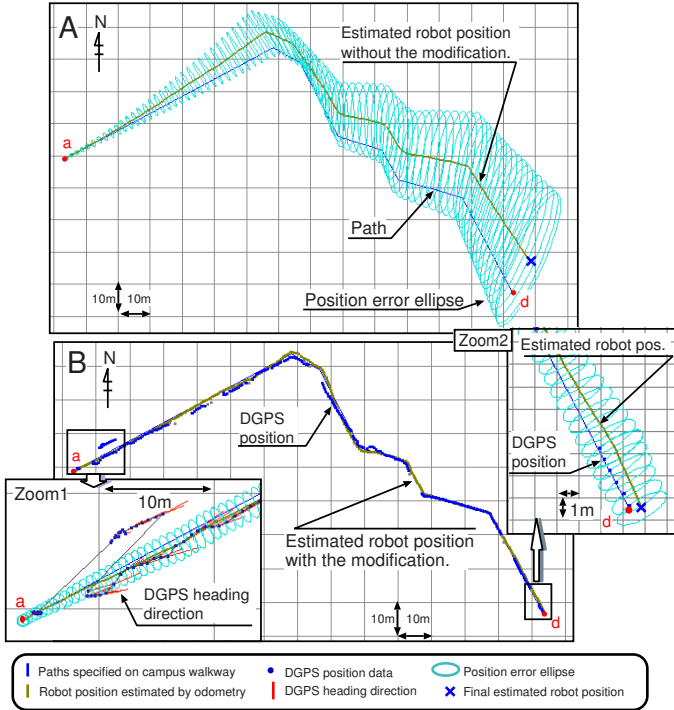


Fig. 5. Comparison of trajectory of the estimated robot position: A illustrates estimated robot position without correction. B illustrates estimated robot position modified by proposed correction method. (each parameters: $l_{xy\ threshold} = 2(87\%)$, $\sigma_{x\ gps} = \sigma_{y\ gps} = 3.5[m]$, $l_{\ threshold} = 1.5(86.6\%)$, $\sigma_{\ gps} = 25[degree]$)

B illustrates a result of position data which were modified by using the proposed correction method which is mentioned in Section 3. Image of Zoom 1 in Fig. 5 B shows a situation which the estimated robot position was corrected by only DGPS heading direction data. Finally, the robot position which was estimated by using proposed correction method was approximately 50 [cm] far from the real position of the point *d* when the robot arrived at the point *d* (Zoom 2 in Fig. 5). It is accurate enough for the mobile robot to run automatically in almost all campus walkways. Discrimination of the DGPS position and its heading direction works effectively and the robot position could be corrected by using only reliable data.

4.2 Autonomous Outdoor Navigation

The authors equipped the proposed method of position correction on a mobile robot (Figure 1) and performed experiments of autonomous navigation in some campus walkways among buildings. Figure 2 illustrates a set of the paths: $[a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow l \rightarrow k \rightarrow j]$, $[a \rightarrow m]$, $[n \rightarrow o]$, $[b \rightarrow p]$, $[f \rightarrow g \rightarrow h \rightarrow i \rightarrow j]$ and $[q \rightarrow p]$ which the robot could move along. In these experiments, the displacement of the real robot position from followed path was in 50 [cm] in usual, and was approximately 150 [cm] in maximum. Figure 6 illustrates snapshots of a navigation experiment on a path $[a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow l \rightarrow k \rightarrow j]$ in Fig. 2. The numbers

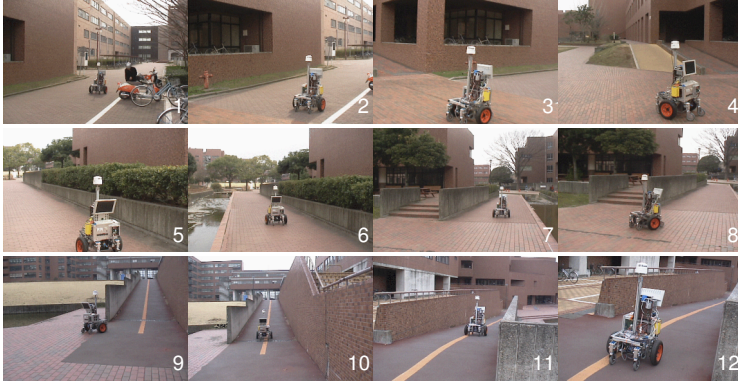


Fig. 6. Snapshots of autonomous outdoor navigation using DGPS and odometry

in Fig. 6 denotes time order. In this experiment, the robot automatically ran 360 [m] from start to goal. The difference between desired goal point and the point which the robot reached was approximately 30 [cm]. In Fig. 6, the robot moved along the center of campus walkway because the authors specified navigation paths at the center of campus walkways. Figure 7 shows the estimated robot position and DGPS measurement data which are recorded in the experiment. In zoom image of Fig. 7, unreliable DGPS position data were eliminated and the robot position was correctly estimated and modified. Moreover, between $[l \rightarrow k]$ (Image 9 – 12 in Fig.6), there is a large slope². The authors confirmed that the robot could also pass through the slopes using proposed navigation method which uses two dimensional map without height information.

5 Conclusion and Feature Work

Based on DGPS and odometry data fusion in the proposed framework by the authors, a mobile robot autonomously navigated in campus walkway among buildings where DGPS measurement tends to suffer from multi-path of GPS radio wave. In this paper, the authors proposed a novel method which checks reliability of DGPS measurement data based on odometry and corrects robot position using only accurate and reliable DGPS measurement data. The authors equipped the proposed correction method on the mobile robot, and also had experiments of autonomous navigation in the campus walkway. However, the proposed correction method fundamentally holds its reliability by canceling the odometry's cumulative error with accurate DGPS measurement data. Thus, its reliability decreases if its error is not canceled for a long time. Once such situation occurred, the correction of robot position failed and the navigation will not be completed. As future work, it is necessary to solve this problem. In addition, the authors once tested a RTK-GPS receiver to measure the robot position and could not get good result. However, GPS technology grows day by day. The authors will pay attention to technical movement of RTK-GPS receiver.

² $[l \rightarrow k]$: an inclination of 5.3 [degree], 2.95 [m] as height and 31.8 [m] as length of the slope. $[h \rightarrow i]$: an inclination of 3.9 [degree], 2.5 [m] as height and 36.9 [m] as length of the slope.

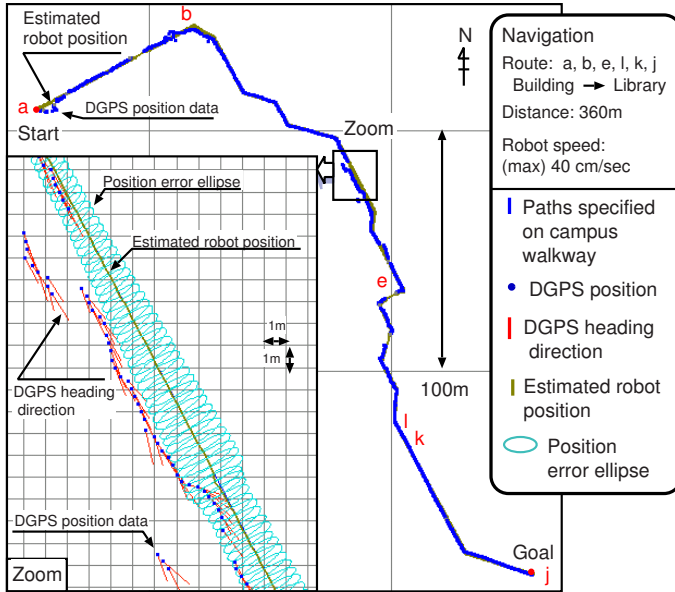


Fig. 7. Estimated robot position and DGPS measurement data of autonomous navigation

References

1. S. Sukkarieh, E. M. Nebot and H. F. Durrant-Whyte, "A High Integrity IMU/GPS Navigation Loop for Autonomous Land Vehicle Applications," *IEEE Trans. on Robotics and Automation*, Vol. 15, No. 3, pp. 572–578, 1999.
2. T. Aono, K. Fujii, S. Hatsumoto and T. Kamiya, "Positioning of vehicle on undulating ground using GPS and dead reckoning," *Proc. of the 1998 IEEE Int'l Conf. on Robotics and Automation*, pp. 3443–3448, 1998.
3. R. Thrapp, C. Westbrook and D. Subramanian, "Robust localization algorithms for an autonomous campus tour guide," *Proc. of the 2001 IEEE Int'l Conf. on Robotics and Automation*, pp. 2065–2071, 2001.
4. M. Kise, N. Noguchi, K. Ishii and H. Terao, "The Sensor Fusion Algorithm for Control of Agricultural Autonomous Vehicle," *Proc. of Robotics Mechatronics 01 Japan*, No. 2P2-J9, 2001, (In Japanese).
5. K. Ohno, T. Tsubouchi, B. Shigematsu, S. Maeyama and S. Yuta, "Outdoor Navigation of a Mobile Robot between Buildings based on DGPS and Odometry Data Fusion," *Proc. of IEEE Int'l Conf. on Robotics and Automation*, Taiwan, pp. 1978–1984, 2003.
6. K. Ohno, T. Tsubouchi, B. Shigematsu and S. Yuta, "Proper use of GPS for Outdoor Navigation by an Autonomous Mobile Robot," *Proc. of The 8th Conf. on Intelligent Autonomous Systems*, Netherlands, 2004 (Accepted).
7. S. Iida and S. Yuta, "Vehicle Command System and Trajectory Control for Autonomous Mobile Robots," *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and System*, Vol. 1, Japan, pp. 212–217, 1991.
8. Y. Watanabe and S. Yuta, "Estimation of Position and its Uncertainty in the Dead Reckoning System for the Wheeled Mobile Robot," *20th ISIR*, pp. 205–210, 1989.
9. H. F. Durrant-Whyte, *Integration Coordination and Control of Multi-Sensor Robot Systems*, Kluwer Academic Publishers, USA, pp. 113, 1988.

Vehicle Localization Using Inertial Sensors and GPS

Libor Přeučil and Roman Mázl

Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Technick~E2, 166 27, Prague 6, Czech Republic
{preucil,mazl}@labe.felk.cvut.cz
<http://gerstner.felk.cvut.cz>

Abstract. The presented contribution describes an approach to preprocessing and fusion of additional vehicle onboard sensors - the odometer and accelerometer, all targeted to serve as optional and temporary substitute for GPS-like navigation. The suggested solution explores a rule-based system for mutual substitutions and calibrations of the used sensors depending on actual conditions. The only usage of the GPS here stands in providing regular position calibrations and serves as a reference method for evaluation of the presented results. The presented solutions have been experimentally tested with real-world data as shown in the experimental part of the paper. Keywords: data fusion, vehicle locator, odometer, accelerometer, odometry corrections, inertial navigation, satellite navigation

1 Introduction

Many today's applications of mobile systems desire more sophisticated solutions to their automation. In particular, this can be achieved via creating reliable system for physical localization of the mobile systems of any kind in general. As the task does in its' full complexity for any kind of entity (in general case these might also be even humans or other living entities) to be localized seems to be of diverse approach, our solution is targeted mainly on wheeled type of vehicles. Although this assumption sets constraints for the final solution, provides also sufficient space for design and testing of specific methods of inertial sensor fusion for navigation purposes.

Standard vehicle position control systems usually rely on track-aside equipments (landmark beacon systems, etc.) supporting the solution of the localization task. On the other hand these might also bring many disadvantages. Track-aside equipments are hard protected against vandalism, doesn't allow dynamic changes and do not guarantee reliable, or even any, performance at accident situations. Besides that, maintenance and operation costs are high. Therefore, recent trends are turning attention towards vehicle-born systems through, so called, distributed intelligent control systems.

The fundament requirement for usage of intelligent control system is availability of continual information about actual configuration of the environment and vehicle position. A satellite navigation system (GPS) is always considered for a backbone for outdoor localization, it still has to be supported or backed-up by another system to improve its' reliability. Although performance of the GPS system was improved

significantly within the last decade, it can still fail from its' principle and due to inaccessibility of the RF signal from satellites in particular situations. These situations are considered for critical and should be avoided from safety reasons. The GPS signal is not available in many locations due to signal shielding e.g. in urban areas, underground spaces, inside buildings, tunnels, deep and narrow valleys, etc.

While the GPS is not able to meet the basic requirements in terms of integrity and availability in a general sense, therefore the localization control system can't rely exclusively on the GPS. The vehicle can be equipped not only with a satellite receiver of navigation data, but also it can employ other on-board sensors e.g. odometry or inertial navigation sensor. Each of these sensors have their own specific features and disadvantages and compared to the GPS they mainly employ the dead-reckoning principle to obtain some suitable navigation information.

Application of sensors for inertial navigation (gyros, accelerometers, tilt sensors) does not depend on operational condition. Their usage has to take into account many fluctuating parameters (e.g. sensor drift, bias, non-linearity). Precise estimation of the sensor parameters has direct impacts on desirable accuracy and reliability of the localization system in this case.

Dead reckoning by the means of odometry fails if insufficient adhesion between the vehicle wheel and ground occurs, due to adhesion or type of the surface (e.g. rain, ice or leaves, soft-type of surface). Quality of adhesion impairs particularly when the vehicle accelerates or brakes. Authors in [1] evaluate the vehicle speed based on a fuzzy inference system and neural networks using differences and rapid changes between odometry sensors joined with multiple wheels.

There are many references in robotics field on data fusion from gyroscopes, accelerometers and odometry. The most of them solve data fusion problems employing the Kalman filter [2], or PDAF techniques [7]. Our approach doesn't use classical state vector to determine current measured values and their errors. Instead of this we recognize occurrence of error situation directly and repair these situations by interpolation methods subsequently.

Aiming to achieve better final performance of the navigation system, we introduce an application-oriented approach to fusion of data measured by the odometry and onboard accelerometer in the following. The presented approach is driven by a belief, that the most typical errors of these sensors are uncorrelated. Significant odometry errors occur during acceleration or braking intervals, which can be successfully discovered by accelerometers. On the other hand, long-lasting and more or less constant motion speeds are very good preconditions for error-free odometry measurement. Then, accelerometers are typically useless in these cases. Therefore, combination of both the sensor types has been assumed to improve the quality of the localization solution.

2 Problem Setup

Navigation of the vehicle consists of determination of forward position on its' path and in precise detection of motion direction, in particular in curves. If the GPS signal

is permanently available, accuracy of the GPS is sufficient even for determination of changes between two tracks after passing a curve of crossing. Whenever the signal of the GPS is lost, the correct tracking of the vehicle position has to be maintained. One of the worst-case situations comes about when the vehicle drives through a shielded region doing some maneuvers. The GPS needs relatively long time for retrieval of actual position again.

This problem situation can be partially overcome by application of inertial sensors, principally gyros and accelerometers with active axis oriented perpendicularly to the motion direction. The gyros can be used for preserving information about heading. For precise solution of heading in a long-term period correction mechanisms have to be employed. One of the powerful approaches is a map-matching algorithm [3], which desires at least an estimate of the traveled distance to determine the position and heading. The odometry can be used for this purpose, but it suffers from randomly occurred, unpredictable and almost unbound errors due to insufficient wheel adhesion.

On the other hand, the main constraints of the inertial system navigation system performance to estimate the traveled distance are set by the finite (and limited) resolution of the sensors themselves. Even a small but permanent offset error in acceleration will be integrated and results in a remarkable error in speed. After double integration it raises in a large error in distance. Therefore, very precise and low offset sensors and error correction mechanisms (feedback algorithms) are necessary to obtain an acceptable inertial navigation platform. Therefore our contribution deals

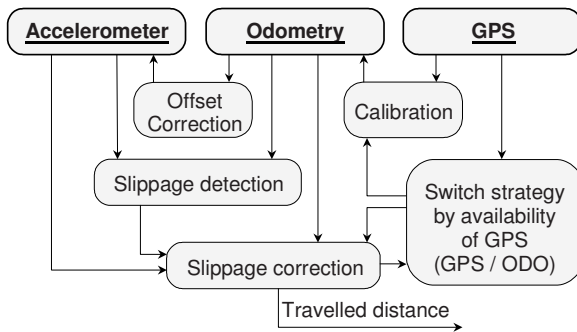


Fig. 1. Principal overview of the proposed method.

with a design of a robust feedback algorithm to perform the double integration of acceleration from accelerometer with acceptable final errors enabling to use the method output as a temporal-substitute dead-reckoning system. For simplification, in the first steps, the method has been designed for the case of accelerometer in question with its' active axis having mounted collinear with the vehicle driving direction. Global overview of the proposed method and mutual interconnections of particular sensors in the approach to distance measurement are illustrated in the Fig. 1.

2.1 Data Analysis and Preprocessing

The approach to estimation of the traveled distance is based on processing of signal from odometer and accelerometers. As the odometer measurements may be processed directly, the acceleration data are corrupted by remarkable noise (caused by vehicle vibrations while driving and/or noise of the sensing system itself).

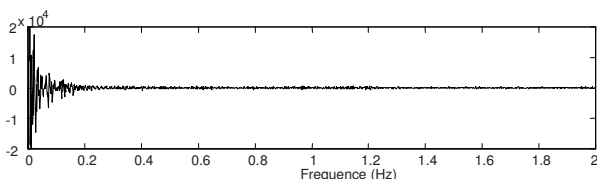


Fig. 2. Frequency spectrum of the accelerometer signal.

Even though data from accelerometer do not need to be filtered before further processing (integration itself provides a strong low-pass filtration effect), we have been interested in filtration of the signal for experiment evaluation purposes.

The intention was to find suitable structure and parameters of a filter providing relatively smooth shape of the signal without damaging the integral (mean) value of the origin. The parameter has to be determined as a trade-off between the smoothness (but also the level of degradation) of the signal and the level of noise in the output signal. To optimize the achieved results, two filtering techniques have been applied; the former uses sliding-window averaging as the latter employs standard 4th-order Butterworth low-pass filter.

The Fig. 3 introduces not only the influence of particular parameters in the process of filtration, but it also illustrates some other issues related to the used sensor offset. As the vehicle speed can be obtained by simple integration of the acceleration along time:

$$v(t) = \int_0^t (a(t) - \text{offset}(t)) dt \quad (1)$$

where a stands for measured acceleration and offset for actual offset of acceleration sensor, being basically an unknown but constrained function of time.

The Fig. 3 compares direct (reference) speed measurements obtained from the GPS and integration of the acceleration after filtration. The precision of the preceding process result strongly depends on the exact estimation of the sensor offset. Unfortunately, the sensor offset is highly variable with time and it is not possible to estimate its' exact model. Moreover, the sensor offset depends on the past behaviour of the vehicle, where its' evolution is driven by unknown transfer characteristic with substantial hysteresis as can be seen in the Fig. 4.

The integration of data plotted in the Fig. 3 assumes that an accelerated body performs a bounded motion with final return to the original position (a forth-and-back motion). This additional information allows us to determine an average offset

along the measured data sequence. However, some deviations of integrated data from the reference GPS shape are clearly visible (e.g. the amplitude of the integrated data is lower than the reference).

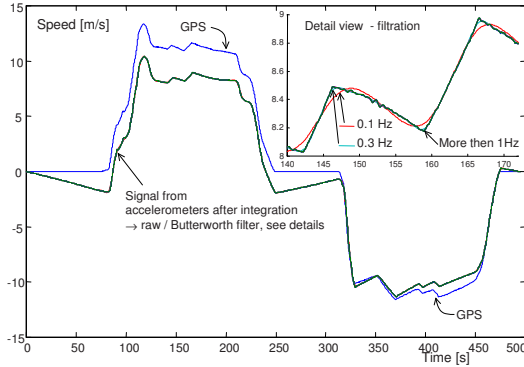


Fig. 3. An example of signal degradation after a filtering.

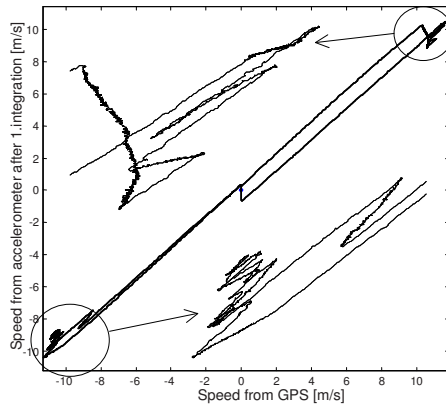


Fig. 4. Comparison of the vehicle speed from GPS vs. speed via acceleration. The shown behavior gives good reasons for the exact determination of a sensor offset to be the central topic in the following sections.

3 Data Fusion

To obtain the travelled distance value, data fusion based on GPS, odometry and accelerometer has to be introduced. The fusion method has to respect the possibility of temporal dropouts in performance of each sensor. Suppose, the GPS signal is mostly available and therefore our effort is mainly targeted to bridge the GPS-dark areas.

The recent research has shown that one of the good ways for fusion of the odometry and accelerometer sensor data is a rule-based mechanism. The desired

behavior is to prefer odometry data to accelerometers as soon as GPS measurement is not available. As the GPS is lost the navigation system has to ensure immediate and smooth switching to inertial sensors and odometry. The basic strategy for fusion and data processing is sketched in the previous Fig. 1.

The odometry data are generally reliable as we suppose precise vehicle wheel calibration and no slippage between the wheel and the ground. Calibration of the odometry seems to be a straightforward task to be performed whenever the GPS is in operation and even together with an existing map of the environment.

As long as the odometer has been properly calibrated (or continuously recalibrated during movements), it is possible to switch the navigation system from GPS to odometer. Unfortunately, the odometer itself can still cause large and cumulative errors due to wheel slippage. Our method for data fusion solves in particular some issues associated with wheel slippage. The core idea of the approach is based on different essence of errors, which both the odometry and the accelerometer give. The odometer typically fails in relatively short time intervals mainly during acceleration or braking periods. These situations can be successfully handled by accelerometer as long as current sensor parameters for integration are known at a time (in particular its' last value of the offset). The odometer and accelerometer can serve as supplementary sensing pair substituting each other, if desired.

During a short time period, whenever neither GPS, nor reliable odometry data are accessible, the measurement of the traveled distance relies only on integration of acceleration. Precondition to achieve reliable results of such integration stands in estimation of the current offset, as mentioned above. Besides that, the estimated offset can also be used for recognition of the odometry slippage.

The final algorithm works in separated time frames (the time length of one basic frame is typically in order of seconds), while in scope of which offsets of the accelerometer are estimated. This is done by the means of LSQ method, which evaluates offset of accelerometer in order to reach a minimal difference between speeds obtained from odometer and integrated acceleration with subtracted offset.

$$offs_{acc} = \arg \min_{offs_{acc}} \sum_{t_k=t_1}^{t_2} \left[v_{odo}(t_k) - \left(\int_{t_1}^{t_k} (a_{acc}(t) - offs_{acc}) dt + v_{t_1} \right) \right]^2 \quad (2)$$

where t_1 and t_2 stands for time-boundary of processed time frame, v_{odo} denotes the odometric speed, a_{acc} means the current sensor value from accelerometer, $offs_{acc}$ stands for estimated current sensor offset for the time frame and v_{t_1} is the initial velocity (at the beginning of a time frame) computed from acceleration.

The minimization process returns estimation of the apparent accelerometer offset. Provided that estimated offset has a sharp slope, it is an indication of the odometry slippage (e.g the estimated accelerometer offset is significantly greater or lesser than a standard value which varies slowly). This situation is illustrated in Fig. 5

In case, that progress of the apparent accelerometer offset is smooth, the offset is treated as a real accelerometer offset over the whole time frame and the final traveled distance is computed directly by double integration of accelerometer data

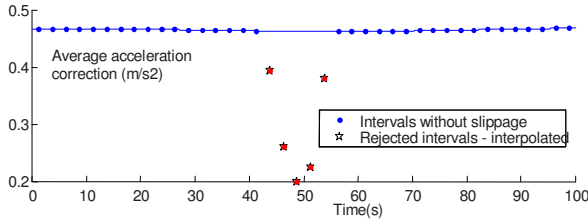


Fig. 5. Selection of rejected intervals and interpolation.

with subtracted estimated offset. This can also be seen as direct locking of evaluation of accelerometer-based distance onto odometry (via a minimization process).

If an extreme offset (a slippage has occurred) is detected, the whole corresponding interval is marked as odometry-unreliable one and pure accelerometer is used for the calculation of traveled distance (with no estimation of the offset from odometry). The evaluation of accelerometer offset has to be treated in another way in order to guarantee proper conditions for the following double integration process in this case. To solution of this problem an extrapolation (or interpolation for slightly time-shifted processing) of preceding offset values before slippage (odometry-reliable interval) can be applied. The 3rd order splines provide reasonable results for interpolation, see Fig. 5.

In fact, the traveled distance estimation always uses accelerometer data and the major differences are only in the way of determination of current accelerometer offset. Then, the final traveled distance is easily computed by double integration with respect to the average offset of the accelerometer.

4 Experiments

The fusion algorithms for odometry and acceleration data were designed for use with train vehicles to serve as complementary substitute to GPS-based vehicle positioning. Experimental data were gathered with a setup carrying an incremental optical odometer offering a resolution of 400 pulses/rev. and industrial accelerometer type Crossbow CXL01LF. The exact reference forward position of the vehicle has been obtained from differential GPS receivers operating in RTK (Real Time Kinematics) mode with the order of about -0.01m accuracy. As the RTK mode of the GPS is not generally suitable for wide practical application due to its' slowness and additional accessories needed, it is very useful for evaluation of the thereunder achieved results.

The real experiment was performed on a real railway track with intentionally created slippage fields (by application of high accelerations and brakings) in specific parts of the path. This situation can be noted in the following Fig. 6 approximately by the 60th sec of the experiment runtime (circled).

The measured odometry and acceleration data were provided to the described fusion algorithm, the quality evaluation of which has been done by comparison with the GPS measurements and which were gathered synchronously. Therefore, the

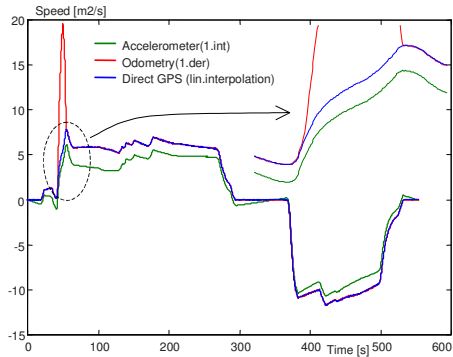


Fig. 6. Input data before processing.

comparison of the real travelled distance provided by the GPS with the result of our approach has been straightforward.

The most important step of the described approach stands in the precise estimation of accelerometer offset; mainly in time frames whenever odometry fails.

The final result of integration with respect to the estimated offset can be seen in the Fig. 7. The long-term accuracy depends on proper calibration of odometry. This means that the results of data fusion can't provide better accuracy than the odometry. However, the major odometry error has been successfully corrected. The remaining distance error after application of suited two-stage integration of acceleration with respect to current sensor offset is less than 1m after 9 minutes drive (see Fig. 7).

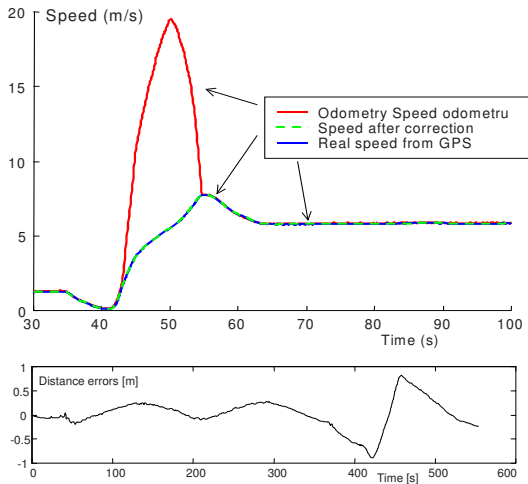


Fig. 7. Result of the correction algorithm and remaining distance error.

We suppose, remaining distance error arises from inaccuracy of odometer if the vehicle drives through curves. Turns induce a change of wheel effective diameter due the centripetal and centrifugal forces.

In order to test robustness of designed approach the designed approach has also been tested with partially artificial test data sets. These data sets are based on original real data, but additional wheel slippages are added by simulation in odometry data.

The Fig. 8 illustrates three simulated slippages besides of first real slippage from acceleration. The first simulated slippage is quite similar to the real one, the second one is very short but heavy and the last slippage simulates a hard-breaking state of the vehicle.

The following Fig. 9 shows result errors after performing the fusion of accelerometer and odometry data with elimination of slippages. The presented approach proofs to be very efficient for higher odometry errors in time frame from 1 to 15 seconds.

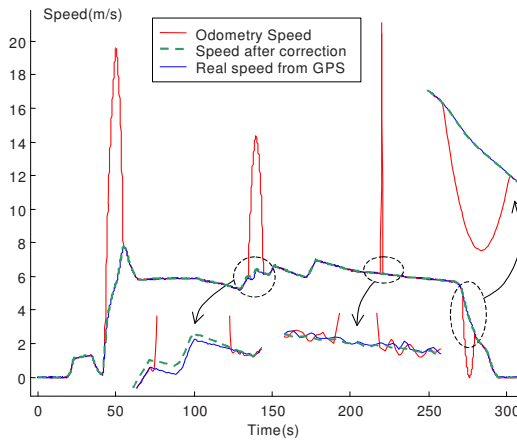


Fig. 8. Results with additional artificial slippage in odometry.

The only bottleneck of the introduced algorithm determined in the experiments can be seen in imperfect detection of extremely small and narrow odometry errors. The primary cause for this is likely to be the impossibility to identify small changes in accelerometer offset to determine these micro-slippages.

5 Conclusion

The presented contribution shows one of the possible and robust ways for utilization of inertial sensors as short and medium time substitute of the satellite navigation system. Long-term precision depends on calibration of the odometer, nevertheless local odometer error induced by wheel slippage is possible to be successfully detected and treated using an accelerometer. The described method is under development towards extension for full 2D localization and it is expected to be targeted on

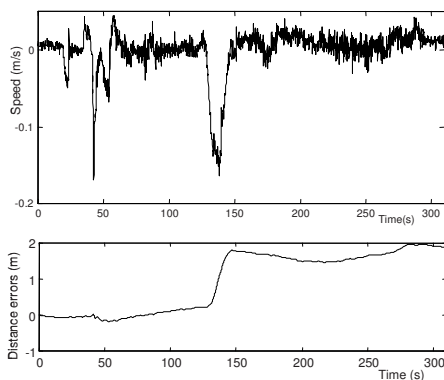


Fig. 9. Speed and distance differences between corrected and GPS data.

improvement of the estimation mechanism for acceleration sensor offset with the objective to achieve higher precision in path-integration. It is assumed, that possible solution to this might lead via dynamic optimisation of processing frames size and combination of their different sizes and/or combining with the sliding-window approach.

Acknowledgement

The presented research has been supported within the IST-2001-FET framework under project no. 38873 "PeLoTe". The work is also supported by the Ministry of Education of the Czech Republic within the frame of the projects "Decision making and Control for Manufacturing" number MSM 212300013.

References

1. B.Allotta, P.Toni, M.Malvezzi, P. Presciani, G.Cocci, V.Colla, "Distance", *Proc. of World Congress on Railway Research 2001*, Koeln, Germany, 2001
2. B.Barsahn, Hugh F. Durrant-Whyte, "Inertial Navigation System for mobile robots", *IEEE Transaction on robotics and automation*, vol.11 no. 3, pages 328–342, June 1995
3. A. Filip, H.Mocek, L.Bazant, "GPS/GNSS Based Train Positioning for safety Critical Applications", *Signal + Draht [93]*, vol.5, pp.16–21 (in German) pp.51-55 (in English)
4. A. Filip, L. Bazant, H. Mocek , J. Taufer, and V. Maixner, "Dynamic properties of GNSS/ INS based trainposition locator for signalling applications", *The proceedings of the Comrail 2002 conference*, Greece, 2002
5. A.Lawrence, *Modern Inertial Technology - Navigation Guidance, and Control*, ISBN 0-387-98507-7, Springer 1998.
6. R. Mázl, "Preliminary study for the train locator project - accelerometer and odometer data fusion", *Research report no. GLR 66/02*, CTU, FEE, Dep. of Cybernetics, The Gerstner Lab for Intelligent Decision Making and Control, Prague, 2002. (Czech lang.)
7. Y.Bar-Shalom, Thomas E. Fortmann, "Tracking and Data Association", *Volume 179 in Mathematics in science and engineering*, ISBN 0-12-079760-7, Academic press, 1988

An Experimental Study of Localization Using Wireless Ethernet

Andrew Howard, Sajid Siddiqi, and Gaurav S. Sukhatme

Robotics Research Laboratory
Computer Science Department
University of Southern California
Los Angeles, California, U.S.A
<http://robotics.usc.edu>

Abstract. This paper studies the use of wireless Ethernet (Wi-Fi) as a localization sensor for mobile robots. Wi-Fi-based localization relies on the existence of one or more Wi-Fi devices in the environment to act as beacons, and uses signal strength information from those beacons to localize the robot. Through the experiments described in this paper, we explore the general properties of Wi-Fi in indoor environments, and assess both the accuracy and utility of Wi-Fi-based localization.

1 Introduction

This paper presents an experimental study exploring the use of wireless Ethernet (Wi-Fi) as a localization sensor. Wi-Fi-based localization relies on the existence of one or more Wi-Fi devices in the environment to act as beacons, and uses signal strength information from those beacons to localize the robot. Compared with traditional localization sensors, such as cameras and laser range-finders, Wi-Fi devices are cheap, light-weight and have relatively low power consumption. Moreover, an increasing number of environments come pre-equipped with suitable beacons in the form of Wi-Fi access points. For robots that are too small or inexpensive to carry a laser range-finder or camera, Wi-Fi-based localization offers a viable alternative.

The basic method for Wi-Fi-based localization is as follows. First, a number of Wi-Fi devices are placed in the environment to act as beacons; pre-existing Wi-Fi access points, embedded devices, or other robots may serve in this role. Second, one or more robots is used to build a Wi-Fi signal strength map of the environment; this map specifies the expected signal strength for each beacon at every location in the environment. We assume that, during the mapping phase, robots are localized using some other technique. Finally, armed only with a signal strength map, a Wi-Fi adapter and odometry, robots may localize themselves using a variant of the standard Monte Carlo Localization algorithm [3,9]. Note that this approach is inspired by the work of a number of authors [2,8] on the subject of Wi-Fi-based localization. Our key contributions are the embedding of the problem within the context of Monte-Carlo Localization (MCL), the development of appropriate Wi-Fi signal strength maps, and the presentation of comprehensive experimental results.

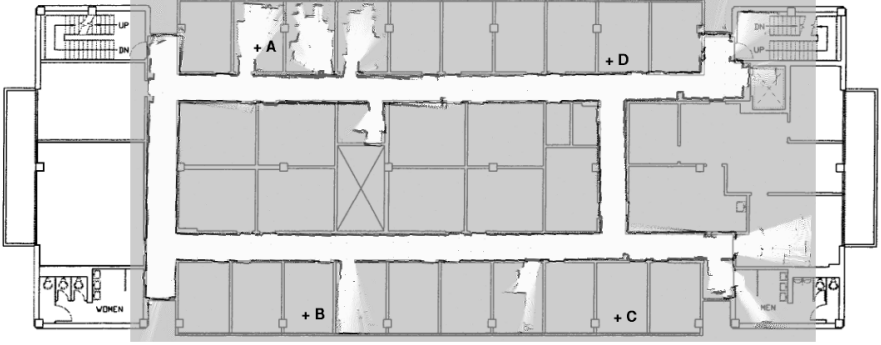


Fig. 1. Building floor-plans of the SAL2 environment, showing the location of the four wireless beacons A, B, C and D. The occupancy grid used for contact sensing and ground truth pose determination is also shown (free space is shown in white, occupied space in black and unknown space in gray).

The experiments described in this paper address four key questions: (1) How does Wi-Fi signal strength vary over time, and to what extent is it affected by day-to-day activity in human environments? (2) How does signal strength vary as a function of robot pose, and is it possible to construct signal strength maps capturing this variation? (3) Are signal strength measurements consistent across robots, such that the signal strength map acquired by one robot can be used to localize another? (4) What level of accuracy is achievable with Wi-Fi-based localization? In answering these questions, we aim to determine both the accuracy and the practical utility of Wi-Fi-based localization.

2 On Monte-Carlo Localization

For the sake of the discussion that follows, we will briefly sketch the basic theory underlying MCL (see [3] and [9] for a more complete presentation). MCL is a form of Bayes filtering; in the context of localization, the Bayes filter maintains a probability distribution $p(x_t)$ over all possible robot poses x at time t . We interpret the probability associated with each pose as our degree of *belief* that the robot is pose x at time t and denote this $Bel(x_t)$. The belief distribution is updated in response to two events: the robot performs some action, or the robot records a new sensor observation. The filter update rules have the following general form:

$$Bel(x_t) \xleftarrow{a_{t'}} \int p(x_t|x_{t'}, a_{t'}) Bel(x_{t'}) dx_{t'} \quad (1)$$

$$Bel(x_t) \xleftarrow{s_t} p(s_t|x_t) Bel(x_t) \quad (2)$$

where $a_{t'}$ is an action performed at time $t' < t$ and s_t is a subsequent sensor reading. Normalization factors have been omitted for the sake of clarity. The terms $p(s_t|x_t)$

and $p(x_t|x_{t-1}, a_{t-1})$ are known as the sensor and action models, respectively, and must be provided a priori. In this paper, we develop a sensor model for Wi-Fi signal strength (Section 5) and evaluate the utility of this model for robot localization (Section 6).

While conceptually simple, the Bayes filter can be difficult to implement. The sensor and action models tend to be non-parametric, and the pose distribution $Bel(x_t)$ is often multi-modal. MCL seeks to address this difficulty through the use of *particle filters*. Particle filters approximate the true distribution by maintaining a large set of weighted samples. Roughly speaking, each sample in the particle set represents a possible robot pose, and the filter update rules are modified such that the action update step 1 modifies the sample poses, while the sensor update step 2 modifies their weights. *Re-sampling* is used to weed out unlikely samples and focus computation on the more likely parts of the distribution. See [1] for a good tutorial on particle filters and re-sampling techniques.

3 Experimental Setup

The experiments described in this paper were conducted in a typical office environment consisting of rooms and corridors (Figure 1). Four wireless devices were placed at the indicated locations to act as beacons (two iPaq's, a laptop and an Intel Stayton unit). The devices were used in ad-hoc mode, and the iwsby utility (Linux) was used to collect signal strength information. The environment was pre-mapped with a scanning laser range-finder to produce the occupancy grid shown in Figure 1. During subsequent experiments, the occupancy grid was used in conjunction with a laser-based localization algorithm to generate "ground truth" pose estimates. These estimates were used for both Wi-Fi map generation and the evaluation of Wi-Fi-based localization.

Two Pioneer2DX robots (Fly and Bug) were used in these experiment. Each robot was equipped with odometry, a SICK LMS200 scanning laser range-finder, an Orinoco Silver 802.11b PCMCIA card, and a range-extender antenna. The antenna was placed in an unobstructed location on the top of each robot in order to minimize any correlation between signal strength and robot orientation. The robots use the Player robot device server [5,4], which includes drivers for measuring Wi-Fi signal strength and algorithms for laser-based (and now Wi-Fi-based) MCL.

In the remainder of this paper, we investigate the general properties wireless signal strength (in both space and time), the construction of signal strength maps, and the use of those maps for localization.

4 Properties of Wi-Fi Signal Strength

Figure 2(a) shows a plot of the signal strength recorded by one of the robots over a 48 hour period. The robot was located adjacent to beacon A in Figure 1, and recorded the signal strength for beacon B. The period captured includes two full working days, with people moving about in the corridors and offices, opening and closing

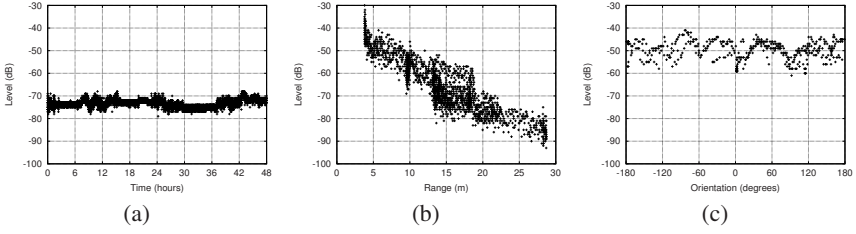


Fig. 2. Signal strength measurements for wireless beacon B. (a) Signal strength plotted as a function of time over a 48 hour period. (b) Signal strength as a function of beacon range. (c) Signal strength as a function of robot orientation.

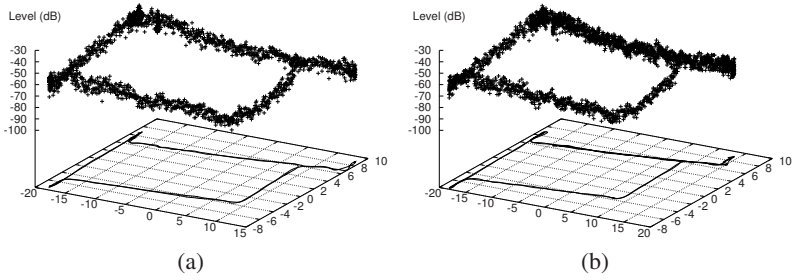


Fig. 3. (a) Signal strength recorded by robot Fly over two complete circuits of the environment. (b) Signal strength recorded by robot Bug over a similar circuit.

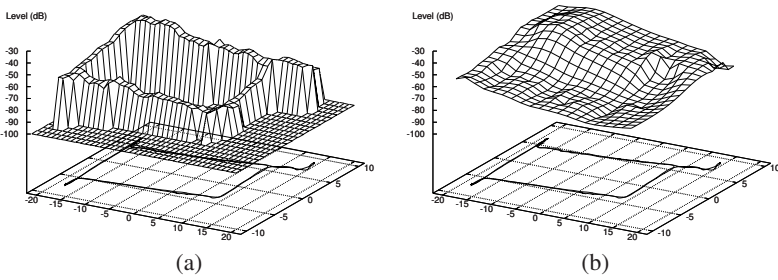


Fig. 4. Interpolated signal strength maps generated using the filters K_1 and $K_1 \cdot K_2$ (described in the text). The maps were generated using the sample set shown in Figure 2(a).

doors, and so on. Some of the variation in the signal strength plot is likely to be a result of such changes in the environment. More importantly, however, all of the variation is confined to a relative narrow band of around 10 dB.

Figure 2(b) shows a plot of signal strength as a function of range from one the beacons. This data was gathered by one of the robots over two complete laps around

the environment; the robot was localized using the laser-based method described above, allowing the range to the beacon to be accurately determined. In free-space, we expect signal strength (a logarithmic measure) to vary as $\log r$. In indoor environments, however, radio is known to have complex propagation characteristics, with reflections, refraction and multi-path effects [6]. Hence it is not surprising that while Figure 2(b) follows the correct general trend for free-space propagation, it also shows significant local departures from this trend. Figure 3(a) shows the same set of data plotted as a function of robot position. In this plot, we note that there is clear variation in signal strength across the environment, and that the local signal strength values remain consistent over multiple passes (within about 5 dB). Somewhat to our surprise, the signal strength values are also consistent when measured by different robots. Figure 3(b) plots the results generated by a second robot for a similar circuit of the environment; the signal strength measurements are indistinguishable from those acquired by the first robot. Finally, Figure 2(c) plots signal strength as a function of robot orientation. While there does appear to be some correlation between signal strength and orientation, this correlation is weak; the variance over the full range of orientations is at most twice that seen in the static time series plot.

Three important implications can be drawn from the data presented in Figures 2 and 3. (1) There is less variance associated with the position plot than the range plot; hence we expect that a signal-strength *map* should yield better localization results than a simple parametric model. (2) Raw signal strength measurements are consistent across different robots having identical hardware. This implies that a signal strength map acquired by one robot can be used to localize another, greatly increasing the practical utility of this approach. (3) Signal strength is largely invariant with respect to robot orientation, at least for the hardware configuration used in these experiments. This result greatly simplifies the construction of Wi-Fi signal strength maps.

5 Mapping Wi-Fi Signal Strength

While it is clearly impractical to probe the signal strength at every point in the environment, it is relatively easy to collect a representative set of samples and construct an interpolated map. We make two important assumptions: (1) during the sampling process, the robot's pose is known (in our case, this pose is provided by a laser-based system), and (2) signal strength is invariant with respect to orientation, reducing map making to a two-dimensional problem.

For simplicity, we encode the signal strength map using a regular grid. Each grid cell records the interpolated signal strength value at a particular location, and separate grids are used for each beacon. The grid is generated from raw signal strength data using a low-pass filter, as follows. Let $\Psi = \{(x_0, \psi_0), (x_1, \psi_1), \dots\}$ denote a set of samples such that each sample i has a position x_i and signal strength ψ_i . Let $\Phi = \{(x_0, \phi_0), (x_1, \phi_1), \dots\}$ denote the set of grid cells, where the interpolated signal strength ϕ_i at position x_i is given by:

$$\phi_i = \frac{\sum_j K(|x_j - x_i|) \psi_j}{\sum_j K(|x_j - x_i|)}. \quad (3)$$

$K(s)$ is a weight function whose value depends on the distance $s = |x_j - x_i|$ between the sample at x_j and the cell at x_i . There are many possible choices for the weight function $K(s)$; to date, we have achieved our best results using a combination of two filters. The first filter uses the weight function:

$$K_1(s) = \begin{cases} 1 & \text{if } s < d \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

This filter considers only those samples that lie within distance d of a cell, and generates an unweighted local average. Figure 4(a) shows the map generated by this filter ($d = 1$ m) when applied to the sample data shown in Figure 3(a). Note that the filter generates good values in those regions visited by the robot, but leaves large ‘holes’ in the unvisited portion of the map. To fill these holes, we apply a second filter with weight function:

$$K_2(s) = s^{-m} \quad (5)$$

where m is generally a low integer value. This is a fairly typical interpolation filter that considers all samples, but assigns higher importance to those closer to the cell. Figure 4(b) shows the final interpolated map.

To use the signal strength map for localization, we must augment it with an appropriate sensor model 2. If we assume that the sensor noise is normally distributed, we can write down the sensor model $p(\phi|x)$ for Wi-Fi signal strength:

$$p(\phi|x) = \exp \frac{-(\phi - \phi_i)^2}{2\sigma^2} \quad (6)$$

where ϕ_i is the interpolated signal strength for the cell i containing pose x , and σ^2 is the expected variance in the signal strength. Based upon time-series plots such as the one shown in Figure 2(a), we typically choose σ to be 10 dB. Note that it is not our intention to suggest that this particular sensor model (or the interpolation procedure describe above) is the the *best* possible model for Wi-Fi-based localization. Rather, we propose that this is a *sufficient* model, and seek to determine its utility empirically.

6 Localization

In order to assess the comparative utility of Wi-Fi-based localization, we compare the localization results achieved using three different combinations of sensors: Wi-Fi, contact, and Wi-Fi plus contact sensing. In all three cases we assume that odometry is also available. Note that the ‘contact’ sensor in our case is logical rather than physical, and simply asserts that the robot cannot be co-located with another object. Thus, given an occupancy map of the environment, the ‘contact’ sensor rejects those poses that lie in occupied space.

Our basic experimental methodology is as follows. Data was collected from a robot performing a series of circuits of the environment, and processed off-line using different combinations of the recorded sensor data. In all cases, the initial pose of

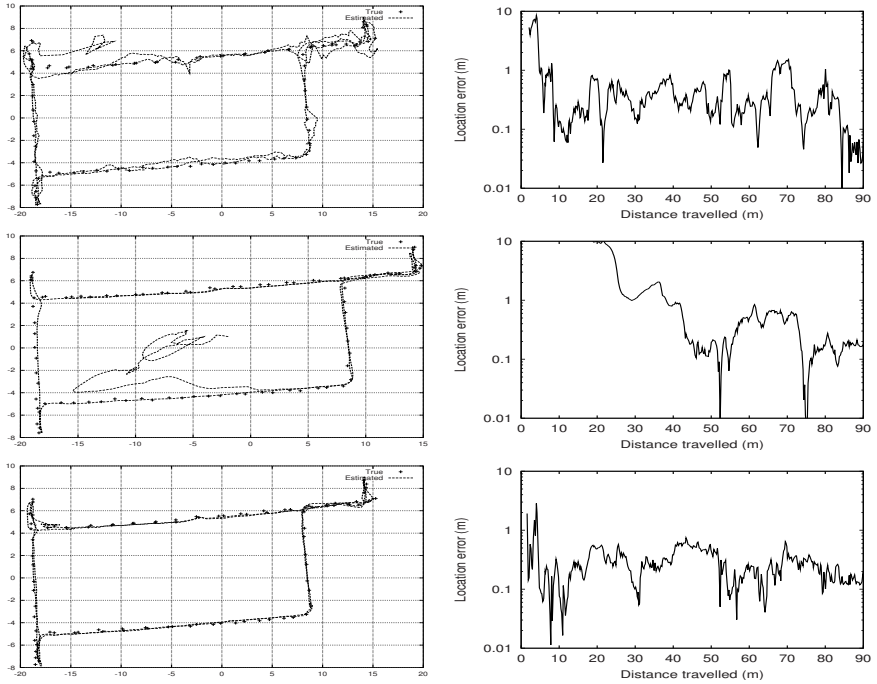


Fig. 5. Localization results using different combinations of Wi-Fi and contact sensing. The plots on the left show the estimated robot trajectory (the true trajectory is indicated by the ‘+’ symbols); the plots on the right show the error in the pose estimate as a function of the distance travelled by the robot. (Top) Wi-Fi sensing only. (Middle) Contact sensing only. (Bottom) Both Wi-Fi and contact sensing.

the robot was entirely unknown. Furthermore, *different robots* were used for the map acquisition and localization phases, and these two robots executed their circuits in *opposite* directions.

The combined localization results are presented in Figure 5. The top row shows the localization results using Wi-Fi sensing only: the left hand figure plots the estimated robot trajectory, while right hand figure plots the error in the pose estimate as a function of the distance travelled by the robot (i.e., the distance between the true pose and the estimated pose). The key feature to note is that the pose estimate converges very quickly (within about 10 m of robot travel) to a steady state error of 0.40 ± 0.09 m. The second row in Figure 5 shows the results using contact sensing only. Here, the estimate takes much longer to converge: the robot has travel around 80 m before it can gather enough data to make an unambiguous determination of the robot’s pose. The steady-state error, however, is only 0.26 ± 0.03 m; better than that obtained using the Wi-Fi sensor alone. The third and final row in Figure 5 shows the results of combining Wi-Fi and contact sensing. Here, convergence is rapid, and

the steady-state error is only 0.25 ± 0.02 m. It would appear that these two sensors complement each other extremely well: Wi-Fi ensures rapid convergence when the initial pose is unknown, while contact sensing improves the subsequent accuracy of the estimate. One can, of course, further improve the estimate by adding data from additional sensors; with sonar or laser range data, we expect to achieve steady-state errors of less than 0.10 m with convergence distances of a few meters.

The results described above were generated using all four of the Wi-Fi beacons placed in the environment. Fewer beacons can be used, with a consequent decrease in localization accuracy. The steady-state errors for different combinations of the beacon A, B, C and D shown in Figure 1 are as follows.

Beacons	Error (m)	Beacons	Error (m)
A	0.87 ± 0.34	A,B,C	0.45 ± 0.14
A,B	0.55 ± 0.12	A,B,C,D	0.40 ± 0.09

Given the basic geometry of trilateration in two dimensions, it comes as no surprise that best results are achieved using two or more beacons. It should also be noted that for this particular set of experiments, the location of the beacons was selected based on the likelihood that it would yield good localization accuracy. Other configurations may yield lower accuracy for the same number of beacons.

7 Conclusion

Four major conclusions can be drawn from the results presented in this paper. (1) Signal strength values are stable over time, and relatively unaffected by environmental changes induced by day-to-day activity. (2) Signal strength values vary relatively smoothly with increasing range from the beacon. As a result, it is possible to produce interpolated signal strength maps from a relatively sparse sampling of the environment. (3) Signal strength measurements are consistent across robots using identical Wi-Fi hardware, and thus maps generated by one robot can be used to localize another. (4) Given a sufficient number of beacons and a signal strength map, robots can be localized to within 0.50 m. Accuracy can be increased by adding additional beacons and/or other forms of sensing. Clearly, much experimental work remains to be done; this paper does not, for example, consider the effect of different environments, different beacon configurations, or heterogeneous hardware. Nevertheless, the results presented here indicate that Wi-Fi is a very effective localization sensor.

Resources

Wi-Fi-based localization has been incorporated into the Player robot device server [5], which can be downloaded from the Player/Stage web-site [4]. The data-sets used in this paper are also available on the Radish (Robotics Data Set Repository) web-site [7].

Acknowledgments

This work is sponsored in part by DARPA grants DABT63-99-1-0015 and 5-39509-A (via UPenn) under the Mobile Autonomous Robot Software (MARS) program.

References

1. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb. 2002.
2. P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.
3. F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA99)*, May 1999.
4. B. Gerkey, R. Vaughan, and A. Howard. Player/Stage homepage. <http://playerstage.sourceforge.net>, September 2001.
5. B. P. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*, Coimbra, Portugal, June–July 2003. To appear.
6. M. Hassan-Ali and K. Pahlavan. A new statistical model for site-specific indoor radio propagation prediction based on geometric optics and geometric probability. *IEEE Transactions on Wireless Communication*, 1(1):112–124, Jan 2002.
7. A. Howard and N. Roy. Radish: the robotics data set repository. <http://radish.sourceforge.net>, May 2003.
8. A. M. Ladd, K. E. Bekris, G. Marceau, A. Rudys, D. S. Wallach, and L. E. Kavraki. Using wireless Ethernet for localization. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, Sept 2002.
9. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence Journal*, 128(1–2):99–141, 2001.

MALOC – Medial Axis LOCALization in Unstructured and Dynamic Environments

Michael Fiebert¹ and Charles-Marie De Graeve²

¹ Siemens AG, Corporate Technology
81730 Munich, Germany
Michael.Fiebert@siemens.com

² Institut d'Informatique d'Entreprise
91025 Evry, France

Abstract. In this paper, the use of medial axis in mobile robot localization is extended. We describe the extraction of robust medial axis from a local occupancy grid to form a global map of the environment. We do not require any special geometric primitives to be present, and we allow substantial changes in the appearance of the environment, as long as the overall structure stays intact.

We then describe how to use such a map with Monte Carlo Localization. Two approaches are treated. The first one relies on extracting salient features (tripods) from the local grid. The second one makes use of salient points in the global map and works well, even if no useful medial axis can be extracted during localization due to many new objects in formerly free space. This work aims to develop an efficient navigation system that requires less assumptions about the environment and its dynamics than current systems do.

1 Introduction

While medial axis have first been introduced in the image processing community and lately gained new attention in the context of searching large image databases and processing 3D medical data [1–4], they have also been used in the area of map building and localization for mobile robots [5–7]. In this paper, we deal with the problem of mobile robot localization. In our approach we overcome restrictions that have been imposed in former medial axis based approaches, for example the need to drive on special paths while map building [5,7], or the need of line features present in the environment [6]. Most notably we address the disadvantage of medial axis, that they are not robust with respect to new objects in formerly free space. In this paper, the medial axis approach is developed to be useful in practice despite this basic problem. We focus on position tracking here and describe the generation of robust medial axis in some detail, since this forms the basis of our map. We omit further details of map building though. Loop closing and also global localization in a dynamic environment are beyond the scope of this paper. Our reason for using medial axis is to capture 'salient' features of the environment, i.e. features that remain more or less unchanged under a certain type of dynamics in the environment. Think of the changes in a department store where the goods and decoration change with the seasons, but the overall structure of the aisles stays more or less the same.



Fig. 1. Local occupancy grid from laser range measurements in a canteen. No line features are present and the dots are not invariant, because chairs are moved, and because of sensor noise. Nevertheless a structure is present, which is persistent over time.

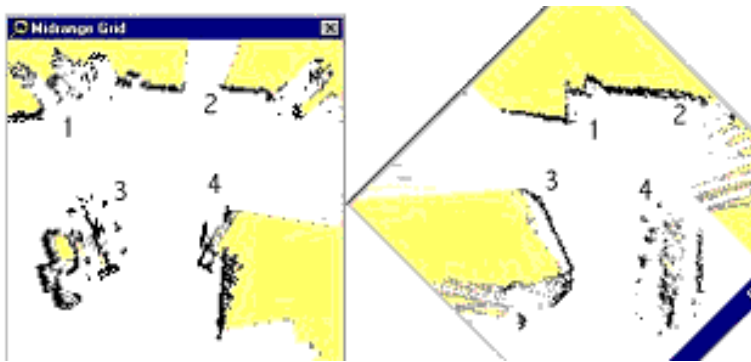


Fig. 2. A considerable change in appearance is shown, caused only by a slightly different sensor height. Specular reflection occurred at a metal plate (1) and the bumper of some shelves came into sight (3,4). Additionally a door (2) had been closed. (Grid maps roughly aligned)

Geometric features like lines, which are often used for mobile robot localization, may be all heavily influenced by this kind of environment dynamics, or not even be present at all, and can't be used here. Unfortunately, also the raw medial axis isn't stable either, but we can use a subset of it, which in fact is mostly invariant. We call this subset "robust medial axis". Currently, we assume a mostly static environment during map building, but allow for a considerable degree of extra dynamics during position tracking. Think of people walking or standing around or new advertising pillars being set up. The organization of this paper is as follows: in section 2, the robust medial axis is motivated and methods for its construction are explained. In section 3 methods to generate and evaluate configuration hypotheses of the robot are presented and used for Monte Carlo Localization. In section 4 we report some results and finally give a conclusion and outlook in section 5.

2 Robust Medial Axis

The medial axis was introduced by Blum [1], and is defined as the set of points in free space that have equal minimal distance to at least two obstacle points (similar to Generalized Voronoi Diagrams, see [2]). In our application it basically forms the skeleton of free space.

2.1 Objective of Robust Medial Axis

The objective of robust medial axis is to capture 'salient' features of the environment, i.e. features that remain more or less unchanged under a certain type of dynamics in the environment. This type of dynamics is difficult to define formally. Instead, we give a couple of examples:

- In a department store, the overall structure of the aisles is more or less constant, even if the presentation of the goods is changed regularly.
- In storage rooms, shelves may be more or less filled with goods of different appearance, but again, the overall structure of the aisles is fixed.
- In a cafeteria, the overall arrangement of tables and chairs doesn't change much. However, individual chairs are moved around a lot (see Figure 1).

These modifications can be seen as 'noise' in the environment itself. Our map should be mostly invariant with respect to this kind of dynamics. In a similar way, noise induced by the sensors should not be captured in the medial axis:

- Some sensor measurements are very sensitive to the exact sensor perspective. Driving over slightly uneven terrain or sharing a map with another robot sometimes reveals the same type of dynamics as when changing the presentation of goods, but here they result from the measurement process (see Figure 2).
- Sensor noise or quantisation errors cause smooth surfaces to appear rough and induce a lot of medial axis branches, just as small objects in a storehouse or the chairs in a cafeteria do.

In Figure 3a a medial axis is shown that obviously contains too much detail. As a consequence, much of the information is not salient, i.e. it is not typical of the environment, but arises from sensor noise or highly dynamical environment detail. Consequently, it does not contribute to the task of localization, but instead only increases the memory requirement and makes matching difficult. The robust medial axis does not contain this useless information. Therefore, it requires less space, the matching algorithm runs faster, and fewer ambiguities occur. In addition to environment and sensor properties, also the size of the robot can be taken into account. In our maps, any detail that is smaller in size than the minimum width for the passage of the robot is omitted. The resulting medial axis is therefore not only useful for localization, but also for motion planning, since the robot can travel along the medial axis.

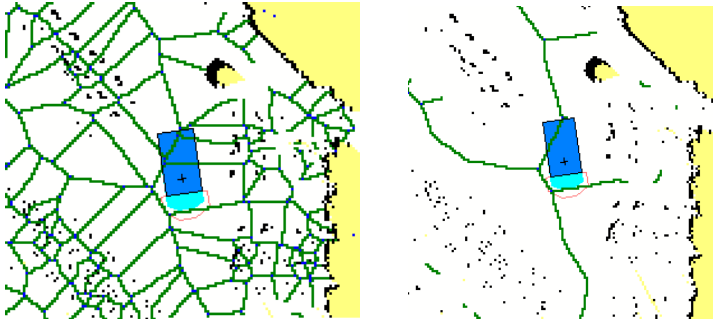


Fig. 3. Medial Axis (a) before and (b) after pruning.

2.2 Generation of Robust Medial Axis

We use the intermediate representation of a local grid map (ca. 10m*10m) instead of trying to generate the medial axis directly from sensor measurements. This allows for convenient fusion of different sensor modalities and is an important first step to filter out sensor noise by integrating measurements over time. Different sensor perspectives contribute to this grid map and allow the calculation of medial axis, even if two opposing boundaries of a free space area can not be seen from any single sensor perspective. In the examples shown in this paper, the grid maps are generated from laser scanner measurements. Successful tests have also been made with grid maps resulting from ultrasonic sensors and from a combination of both. Starting from the local occupancy grid, the cells belonging to the raw medial axis are obtained by a Distance Transformation. We use the algorithm described in [8], which finds the distance transformed in $O(n)$ time (n is the number of grid cells) by propagating Euclidian distances from top to bottom, from left to right and vice versa. Then the minimum of the four runs is taken. Due to the Euclidian metric the result is rotation invariant.

As said before, the raw medial axis contains too much detail (Figure 3a). But the propagations already provide us with two attributes that allow a decision about which parts can be removed:

- Distance of a medial axis point to the next obstacle point.
Note that this distance is attained at at least two distinct points. A point where the minimal distance is attained is called a generator of the medial axis point. The circle around the medial axis point with the radius corresponding to the distance to the next obstacle point is called the tangent circle.
- Distance between the generators.
This can be understood as a simplification of the residual functions described in [2].

Based on these attributes, the raw medial axis is pruned, i.e. certain points are dismissed from it:

- If the distance of the medial axis point to the next obstacle is below a threshold (for example the width of the robot), the point is removed from the medial axis.
- If the distance between the generators of a medial axis point is below a threshold, this medial axis point is removed.

Especially the second criterion is a very effective measure to eliminate unwanted branches originating from rough boundaries. These branches are unwanted because fine details are not considered "salient" even if they are an accurate picture of the reality at this time. Besides, they often are a result of sensor noise or quantization errors. Figure 3b shows the result of such a pruning operation.

2.3 Generation of Tripods

A tripod is derived from a point in the medial axis of topological order higher than two. Loosely speaking, these are the "crossroads" or "T-junctions" in the paths defined by the medial axis.

Because they are important landmarks, some additional care is taken to make them robust. The tripod captures the "T-junction" and consists of

- The point itself.
- The radius of the tangent circle.
- The intersection points of the medial axis with the tangent circle.
- Two angles between the lines from the medial axis point to the intersection points with the tangent circle.

Notice the last two points: instead of using generators, branching points are attributed with the intersecting points of the medial axis with the tangent circle. These points are very stable, whereas the generators can be instable if a boundary zigzags close to the circle. The instability of the generators only matters at branching points, because only here the tripod defines angles, which will be used for matching later.

Branching points with more than three generators very easily decompose to several points with three generators, which is why we break them up immediately when we encounter them. A branch point with four generators results in four different tripods in our map, all having the same center point. Due to the discrete generation of the medial axis in the grid map, branches with more than four generators cannot occur. For every leg of a tripod, a check is performed to find out whether this leg corresponds to a "real corridor" or a dead end. Therefore the distance from the medial axis to the next obstacle at the points where the medial axis intersects the tangent circle around the branching point is compared (see Figure 4). At dead ends it is significantly smaller (tip of the arrow in Figure 4) than at the branching point. Tripods with such legs are not used for matching.

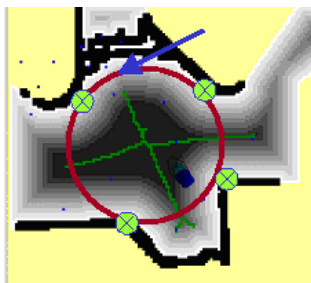


Fig. 4. Replacing generators (crosses) by the more stable corridor centers and discrimination between corridors and dead ends at junctions: the medial axis intersects the circle around a branch point once between two generators. These intersections are more stable than the generators and the value of the distance transformed at those points is a measure for the relevance of the corridor. At the tip of the arrow it is low and indicates an instable dead end.

2.4 Map Representation

Our map consists of all tripods and a set of other points of the medial axis. They may be chosen equally spaced or spacing is adapted according to the local curvature such that the modelling error stays below a certain bound. All points are attributed with the distance to the next obstacle and with their generators (see Figure 5). Theoretically the generators need not to be stored, since they can be recovered from the other data, but this would be a time consuming process. The extra memory is invested to speed up localization.

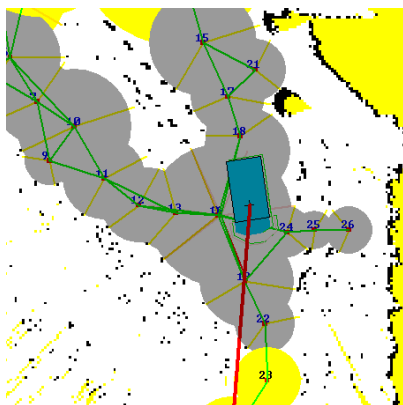


Fig. 5. Visualization of a partial map built from the medial axis in Figure 3a, drawn over the local grid.

Of course we store also the topological connections, although this would not be necessary for the localization process described below! They are very useful for path planning and map building though. We do not use a global coordinate system;

all position information is stored relative to neighboring points. This enables us to build locally consistent maps without the need for global consistency.

3 Localization

For localization, two operations on the described data structures are provided and described in this paragraph:

- Given a map and a local grid map, generate position hypotheses.
- Given a map, a position hypothesis and a local grid, generate a plausibility measure for the hypothesis to be correct.

In our navigation system, we use the Monte Carlo Localization (MCL) method, a probabilistic localization process based on particle filters [9]. Other localization methods could be used as well, e.g. multiple hypothesis tracking. In order to explain the use of the medial axis together with MCL, we briefly summarize it. In MCL the belief about the current configuration of the robot is modelled by a probability distribution, which is represented by a set of samples. Usually for global localization the samples are initialized with equal distribution. Then in alternating phases the samples are propagated according to odometry information, weighted and resampled according to the plausibility for each sample, which is based on the current and expected sensor readings. Since it has been found, that the original MCL algorithm counter-intuitively can yield worse results with better sensors, we use a variant: Monte Carlo Localization with Mixture Proposal Distribution [10]. In addition to the usual forward propagation step, in which samples of the current belief are propagated and rated by sensor measurements, a backward propagation step is performed. It produces samples directly from sensor information, propagates them backwards and rates them with the last belief. A medial axis match can be seen as very distinctive sensor information with which we can produce highly relevant samples for this backward propagation step. This allows us to confine the initial sample set to a small number of configurations instead of an uninformed equal distribution. Because we need fewer samples (about 150) the efficiency and accuracy of localization is improved. Additionally this enables us to recover quickly from very large localization errors (kidnapped robot problem).

3.1 Generation of Configuration Hypotheses

For use with MCL the requirements of the matching process are not very high. Additional hypotheses (and also lost ones) are tolerated without problems. Therefore a very simple approach is sufficient here. In other tasks, e.g. loop closing, more sophisticated matching algorithms are used, e.g. subgraph matching by constraint propagation. In the simple approach hypotheses are only generated when tripods are present in the local map. With the help of an index of all tripods of the global map, sorted by the radius of the tangent circle, possible matches can be found quickly. Each pair with similar radius and angles produces one or more (depending on the

symmetry of the tripod) candidates. Although the algorithm does work even when just all candidates are used, some checks are performed to reduce their number in order to keep the demand for particles low. Candidates get a bonus if they are very close (position and angle) to others. This reflects the observation that the correct hypothesis is generated by a lot of tripod pairings, while wrong hypotheses don't accumulate. Candidates get a malus, if their local distance to another tripod is different from the corresponding distance in the map.

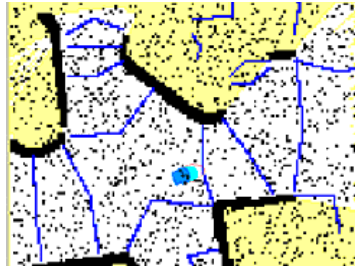


Fig. 6. A configuration hypothesis defines a projection from the medial axis map into the local grid. It thus can be evaluated by deciding how well the generators (endpoint of straight lines) fit into the grid. The evaluation is done by a lookup at those points in the distance transformed. The picture shows simulated data with heavy synthetic noise added.

3.2 Rate Configuration Hypotheses

Given a configuration hypothesis of the robot both in the global and the local map, we can project the generators of the global map into the local grid map. We expect obstacles at those positions. With disturbance due to sensor noise or dynamics that might not be true exactly. Therefore we do not look up the occupancy in the grid, but the distance to the next obstacle in the distance transformed of the grid. Then the sum of the squared (in order to punish large ones) distances gives a good measure of similarity which can be computed extremely fast since this is basically lookup. The necessary normalization is done in the MCL process, where the sample weights are normalized to sum up to one. Notice, that we need not extract any medial axis for this during position tracking. This approach is very robust with respect to new obstacles (Figure 6), like people walking around. Some amount of removed obstacles can also be tolerated.

4 Results

Successful tests have been performed on a pioneer robot (Figure 7) and with recorded data. In experiments in a canteen (ca. 800 m², small parts shown in Figures 1, 3) the robot localized very well, even with a map built from laser data that had been



Fig. 7. The pioneer robot used for most experiments, here with additional omnidirectional camera

recorded more than one year before. Figure 8 shows a map that has been built at night time in a mall where mostly closed shutters were seen by the sensors. Localization at daytime worked well, with errors on the order of some centimeters, even though no pre-processing was done to filter out the people walking around. Furthermore, the sensors were now seeing all kinds of goods instead of the shutters.

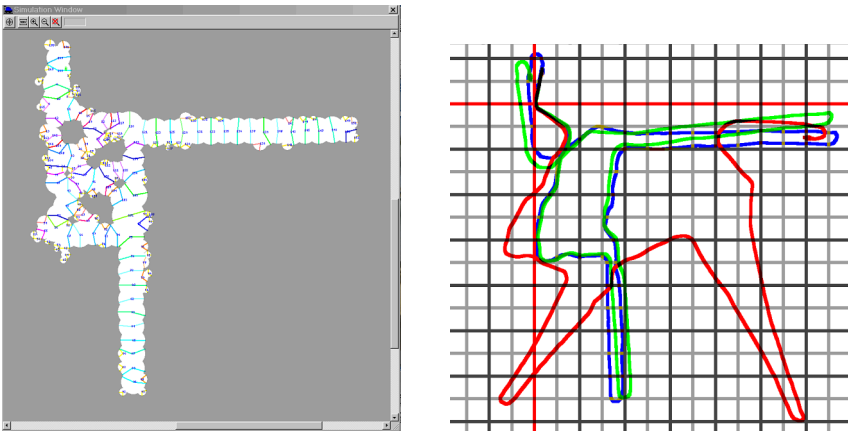


Fig. 8. Map built at nighttime in a mall and trajectories of a localization experiment at daytime. (red: odometry, green: odometry and gyroscope, orange: MALoc localization)

5 Conclusion and Outlook

Methods have been presented to efficiently extract robust medial axis for online map building on a mobile robot. Then it has been shown how to use such maps together with Monte Carlo Localization in a way that is very robust with respect to new objects in formerly free space. Besides a more detailed quantitative analysis of

the performance, the next step will be an extension that allows to cope with new objects not only when evaluating hypotheses, but also when generating hypotheses by means of matching. Also we want to more explicitly include the case of objects disappearing and handle areas where obstacles are very sparse.

Acknowledgement

This work was partially supported by the German Department for Education and Research (BMBF) under grant no. 01 IL 902 D0 as part of the lead project MORPHA.

References

1. H. Blum, "A transformation for extracting new descriptors of a shape". In Whaten-Dunn (ed.) *Models for the Perception of Speech and Visual Form*, MIT Press, Amsterdam, pp. 362-380, 1967.
2. R. Ogniewicz, M. Ilg, "Voronoi Skeletons: Theory and Applications," *Conference on Computer Vision and Pattern Recognition*, pp. 63-69, 1992.
3. P. Klein, T. Sebastian, B. Kimia, "Shape matching using edit-distance: an implementation," *Proceedings of ACM/IEEE Symposium on Discrete Algorithms (SODA)*, pp. 781-790, 2001.
4. M. Styner, G. Gerig, S. Joshi, S. Pizer, "Automatic and Robust Computation of 3D Medial Models Incorporating Object Variability," to be published in *International Journal of Computer Vision*.
5. H. Choset, K. Nagatani, "Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization Without Explicit Localization," *IEEE Transaction on Robotics and Automation*, vol. 17, no. 2, pp. 125-137, 2001.
6. D. Van Zwynsvoorde, T. Siméon, R. Alami, "Building topological models for navigation in large scale environments," *Proceedings of International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, pp. 4256-4261, 2001.
7. B. Kuipers, "The Spatial Semantic Hierarchy," *Artificial Intelligence*, vol. 119, no. 1-2, pp. 191-233, 2000.
8. P. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, pp. 227-248, 1980.
9. D. Fox, S. Thrun, W. Burgard, F. Dellaert, "Particle Filters for mobile Robot Localization". In A. Doucet, N. Fereitas, N. Gordon (eds.), *Sequential Monte Carlo Methods in Practice*, Springer, 2000.
10. S. Thrun, D. Fox, W. Burgard, "Monte Carlo Localization With Mixture Proposal Distribution," *Proceedings of AAAI National Conference on Artificial Intelligence*, Austin, Texas, pp. 859-865, 2000.

Market-Based Multirobot Coordination Using Task Abstraction

Robert Zlot and Anthony Stentz

The Robotics Institute
Carnegie Mellon University
robz@ri.cmu.edu
tony+@cmu.edu
<http://www.ri.cmu.edu>

Abstract. In this paper, we introduce a novel approach to multirobot coordination that works by simultaneously distributing task allocation, mission planning, and execution among members of a robot team. By combining traditional hierarchical task decomposition techniques with recent developments in market-based multirobot control, we obtain an efficient and robust distributed system capable of solving complex problems. Essentially, we have extended the *TraderBots* market-based architecture to include a mechanism that distributes tasks among robots at multiple levels of abstractions, represented as task trees. Results are presented for a simulated area reconnaissance scenario.

1 Introduction

Many applications of multirobot systems can be viewed in terms of problems dealing with scarce resources in highly uncertain, dynamic environments. The optimization problems involved are usually highly unscalable, making centralized solutions impractical. Distributed solutions can be faster and more reliable; although these benefits are often obtained in exchange for guarantees on solution quality. By extending market-based multirobot coordination architectures to deal with hierarchical representations of tasks, we produce a distributed mechanism that can quickly generate efficient solutions to complex optimization problems.

Our research addresses both the task allocation problem as well as multirobot planning. The multirobot task allocation problem can be stated as follows: *given a set of tasks and a group of robots, what is the optimal way to distribute the tasks among the robots?* In terms of mission planning, it is desirable to divide the planning among the robot team to exploit the presence of multiple processors and asymmetric local information, but it is difficult to do this while still ensuring that the resulting global plan is efficient.

This paper presents a task-oriented market mechanism in which the participating agents exchange tasks at multiple levels of abstraction, represented as task trees. This effectively distributes task decomposition among the robots, and creates a market where the best plans will dominate.

In the next section we discuss related work in both market-based multirobot coordination and in hierarchical task decomposition. We then detail our approach,

which introduces a unique market framework based on task trees. We first define the task tree representations that are used, followed by a description of our task tree market mechanism. Results from simulation for an area reconnaissance problem follow.

2 Related Work

We handle the coordination of multiple robots through the use of the *TraderBots*¹ market architecture. Well-known market mechanisms (such as auctions and contracts) are applied to a team of robots and software agents participating in a virtual economy. The market is designed such that the act of each robot trying to maximize its individual profit results in a globally efficient outcome. Market mechanisms have long been applied to software agents, beginning with Smith's Contract Net Protocol in 1980 [12]. Within the *TraderBots* economy, participants can negotiate contracts to execute tasks, or trade in resources required to complete such tasks (such as the use of specialized sensors or tools, storage capacity, and processor time). Market architectures are typically based on distributed control; however Dias and Stent propose a system which includes opportunistic centralized optimization within subgroups of the team [4]. Market-based implementations for multirobot systems have been successfully demonstrated in simulation [3,2,9] and several physical robot applications [6,14,16]. Approaches taken in some market-based systems make use of a central auctioneer to allocate tasks among the robots [9,6]. In contrast, we allow all of the participating robots and software agents to dynamically take on the roles of buyer and seller. To date, market-based multirobot coordination mechanisms have only considered trading tasks at a single level of abstraction.

Our work is also related to research in the area of AI hierarchical task networks (HTN). Task networks are collections of tasks which are related through ordering constraints. HTNs extend task networks by allowing the tasks in the network to be abstract tasks which can be decomposed to ultimately form an executable task network. Erol *et al.* [5] present an overview of HTN planning, and provide a sound and complete HTN planning algorithm.

Although there has been a great deal of focus on task decomposition and HTN or hierarchical planning in the AI literature, few deal explicitly with the case of multiagent planning. Clement and Durfee [1] present an approach for coordinating hierarchical plans of multiple agents. Several agents send their top-level plans to a central planner which searches for a way to merge them. If it cannot do so, then some plans may need to be expanded, which requires agents to send *summary information* at increasingly deeper levels of the tree until a feasible global plan can be produced.

Hunsberger and Grosz [7] describe a system where multiple agents are able to bid on subtasks which are related through task network structures allowing nodes to be grouped together as *roles*. The roles are then assigned by clearing multiple centralized combinatorial auctions. Our work differs in that we dynamically construct

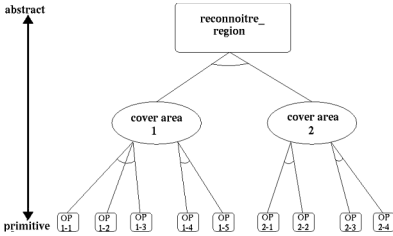
¹ The name *TraderBots* was recently adopted for our market-based coordination work that was started by Stentz and Dias in 1999 [13].

task hierarchies rather than using predefined recipes, and we allow the distribution of planning computation among agents. We also use the tree structure to dictate which tasks can be bid on in combination, rather than specifying roles or considering all combinations of tasks/roles to bid on.

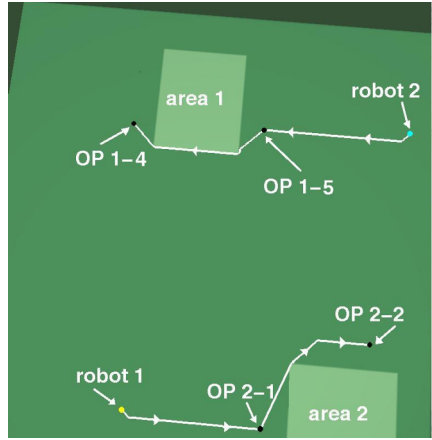
3 Task Tree Auction Mechanism

3.1 Task Trees

We represent composite tasks using task trees. A task tree is defined as $\mathcal{T} = (T, r, E)$, where T is a set of tasks or nodes, $r \in T$ is a unique root task (node), and E is a directed edge set which specifies parent-child relationships between the tasks in T (see Figure ??(a)). Each successive level of the tree represents a further refinement of an abstract task; the root is the most abstract task, and the lowest level contains primitive actions that can be executed by a robot or another agent in the system. Subtasks are related to their parents through logical operators, such as *AND* and *OR*. To execute an *AND* task, all of its subtasks must be executed; to execute an *OR* task, any one of its subtasks may be executed. Constructing a task tree involves performing a hierarchical decomposition on an abstract task.



(a)



(b)

Fig. 1: (a) An example task tree for an area reconnaissance scenario with two areas. The arcs on the edges represent *AND* operators, and the absence of an arc represents an *OR* operator. Note that for the *cover_area* tasks, there are two alternative decompositions specified in the tree. (b) A geometric view of a reconnaissance mission that can be represented by the task tree in (a).

3.2 Task Tree Auctions

The market in our system is primarily based on task tree auctions; contracts are sold for executing trees of tasks. When an auction is announced, each robot deter-

mines its valuation (which is based on cost estimates) for each task in the task tree, whether it is an interior node (abstract task) or a leaf (primitive/executable task). After computing its costs, a robot can then bid on the tasks that it expects are going to be profitable, attempting to win a contract to perform those tasks in exchange for payment. A specially designed auction clearing algorithm ensures that the auction is cleared optimally (minimizing cost) in polynomial time. The winner of the auction is responsible to the seller and must ensure that the tasks are completed (either by executing the task itself, or by subcontracting parts of the task to other teammates in future negotiations) before receiving payment. In general, the payment can depend on the quality of the job completed.

Any agent in the system can act as an auctioneer at any time. Some robots or agents may start with knowledge about the mission objectives, new tasks may be discovered while processing information acquired during execution, or a human operator may introduce them into the system. When a robot has new information about tasks or mission objectives, it can decompose those tasks (not necessarily completely) and call an auction for the corresponding task tree. Currently, each auction is for a single task tree.

Bid Valuation In the case of a primitive (leaf) task, a robot's bid for the task is based on the expected marginal cost of executing the task. If the robot wins a primitive task, it can insert it into its schedule and perform it at the appropriate time.

For an abstract (interior) task, the valuation is the cost of minimally satisfying the task represented by the corresponding tree node. For example, if the connective on the tree node is *AND*, the value of the task is the expected marginal cost of performing *all* of the subtasks (children) of the node. If it is an *OR* node, then it is the *minimum* expected cost of executing *one* of the children. However, the bidding robot may have a better plan than the auctioneer. This can be due to a difference in the current state or resources of the bidder, or due to asymmetric local information. The bidder can perform its own decomposition of the abstract task, and if the resulting plan is of lower cost, then it bases its bid on the abstract node on this new plan. If the plan is indeed a better plan, the bidder wins the task in the auction and it can replace the auctioneer's original plan with its own (it also may subcontract some or all of the new plan in subsequent auctions). This enables the distribution of planning among the robots.

Auction Clearing A task tree auction can be viewed as a special case of a combinatorial auction. A combinatorial auction is an auction in which there are multiple items up for sale, and the bidders can bid on any combination, or *bundle*, of items. In a task tree auction, *primitive tasks* correspond to *items* and *abstract tasks* roughly correspond to *bundles*². While in a general combinatorial auction any arbitrary set

² There is a slight distinction between an abstract *AND* task and a traditional bundle in that abstract tasks may be decomposed differently by a bidder – but, for the purposes of auction-clearing, this distinction is unimportant. Abstract *OR* tasks are also handled

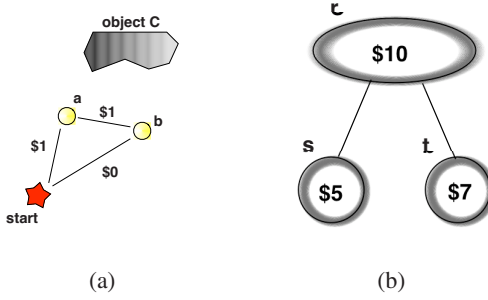


Fig. 2: An example partial task set with its task tree representation. The goal is to create a stereo image of object *C*, which can be achieved by capturing images from multiple view-points. **(a)** The starting location of a robot is shown along with the locations of two view-points. The estimated mobility costs between points are shown. **(b)** A task tree corresponding to the tasks in **(a)**. The two primitive tasks *a* and *b* are shown together with the abstract task *C*.

of items may form a bundle, in a task tree auction the structure of the tree dictates which bundles may be traded. For example, in figure 2 the available bundles are primitive tasks *a*, *b* (singleton bundles) and abstract task *C*. Combinatorial auctions allow bidders to express complementarity and substitutability between items or tasks. For example, in figure 2 the cost to drive to goal *a* followed by goal *b* (\$10) is less than the sum of the costs of driving to each one separately (\$12). A bidder can express this complementarity between the tasks by bidding on the 'bundle' *C* which has a cost that is less than the sum of the individual costs of its subtasks.

The process of allocating the tasks in the task tree to the highest bidder, is an instance of a combinatorial auction winner determination problem (CAWDP). The objective for the auctioneer is to clear the auction by selling the combination of items that would maximize revenue (in which case tasks have been sold to the robots that are expected to be best suited to execute them). CAWDP is known to be \mathcal{NP} -hard in general [11]. However, since the bids are structured as a tree, winner determination can be computed in polynomial time. We use a modified version of an algorithm due to Rothkopf *et al.* [10], which has time complexity $O(n \cdot |T|)$ (where *T* is the set of all nodes in the tree and *n* is the number of leaves in the original tree)³.

One important design consideration is the bidding language used for task tree auctions. There is inherently a tradeoff between the expressiveness and the simplicity of the bidding language for both the bidders and the auctioneers. The auctioneer must ensure that every bidder cannot win more than one bundle in a single auction because the individual bids do not take into account dependencies between the bundles. The simplest possible bidding language for a task tree auction is to allow bids on only one node of the tree. A slightly more complex specification (and the

differently since they represent alternatives among their subtasks, rather than a combination of subtasks.

³ Our algorithm is described in detail in [15].

language that we currently implement) is to choose all nodes along a root-to-leaf path, allowing the path to branch out to become multiple paths at *OR* nodes. Both of these languages ensure that bidders are awarded only one bundle with the current auction clearing algorithm. In contrast, an *exclusive-or* bid language would allow the robots to bid on all tree nodes, but with the connotation that the robot can be awarded only one of the bundles. This can lead to a more efficient outcome, but at a cost to the auctioneer of running a more complex auction-clearing algorithm. We are currently developing algorithms to deal with this more expressive bidding language efficiently.

Auctions can be called in a series of rounds. From an optimization standpoint, a round can be thought of as an improvement step in a distributed local search algorithm. If the auctions in a round are not simultaneous, then after each round the global solution is guaranteed to be no worse than it had been before the round started. Therefore, when starting with an initial feasible solution (or once the initial auction round has terminated) the system behaves like an *anytime* algorithm – a feasible solution is available quickly, and the quality of this solution can only be improved over time. In the case of a multirobot application, the robots can start to execute a feasible plan and at the same time improve the plan through further auctions while executing. The same mechanism can be applied to auction online tasks added to the system during execution (*e.g.* new tasks may be added by a human operator or autonomously generated based on information gathered by the robots).

4 Experiments

Our approach was tested using a graphical simulation of an area reconnaissance scenario. The objective for the robot team is to view all of several predefined named areas of interest (NAIs) within a large geographical area while minimizing (distance-based) travel costs. Each robot is equipped with a range-limited 360° line-of-sight sensor. The terrain is represented as a grid where each cell has associated mobility cost and utility values, representing the cost to traverse the cell and the benefit of viewing the cell respectively.

One robot (robot *R*) is initially aware of the global mission, *i.e.* it starts with an abstract task describing the overall mission. Robot *R* then performs a task decomposition, refining the global task into subtasks representing the task of covering each NAI, and in turn each of the NAI subtasks is broken down into several groups of observation points (OPs) from which a robot can view part of an NAI (Figure ??(b)). The global reconnaissance task is decomposed using a connected components algorithm, where connected areas that were marked with high utility are taken to be the NAIs. The NAI tasks are then decomposed by greedily selecting OPs that have the highest utility-cost tradeoff (a weighted sum of utility and cost is used). For computational purposes, we limit the number of possible OPs considered for each area to twelve per NAI (which surround the perimeter of the NAI). The NAIs are decomposed twice, producing two alternative plans for robot *R*'s initial decompositions.

Once the decomposition is complete, robot *R* holds a task tree auction, thereby distributing subtasks among the team. Other robots may bid on nodes in any level

in the tree, allowing them to perform their own decompositions for interior nodes. The auctions then proceed in rounds in which each robot optionally holds a task tree auction for one of its subtrees in a round-robin fashion.

The mission objective is achieved by the robot team visiting all of the required OPs, minimizing the overall distance traveled. This optimization problem is an instance of the multi-depot traveling salesman path problem (MD-TSPP), which can be stated as follows: *given a list of n cities (OPs), the inter-city costs, and m salesmen (robots) located at different start cities, find the set of m paths (each starting at one of the salesmen locations) that visits all n cities and minimizes the total travel cost.* The traveling salesman path problem (TSPP) is a special case of the MD-TSPP (with one salesman) and is a well-known \mathcal{NP} -hard problem. Thus MD-TSPP is also \mathcal{NP} -hard (and we cannot be guaranteed to find the optimal solution in polynomial time). Since it is not known which group of cities minimizes the objective function and still ensure area coverage, there is another layer of complexity added to the problem; that is, we may have to solve the MD-TSPP for exponentially many subsets of OPs.

5 Results

The above scenario was run varying the number of robots and NAIs, with 100 runs for each case. The robots and NAIs are randomly placed with a 255x255-cell grid at the start of the simulation. NAIs were randomly-sized rectangles with edge lengths drawn uniformly at random in the range of 22 to 32 grid cells. We compared the task tree algorithm with three other task allocation algorithms.

The first algorithm simply auctions off the leaf-level nodes of the task tree (decomposed by the first robot) to the highest bidder one at a time, in random order, until the entire tree is satisfied. If an interior node is satisfied by an intermediate allocation, then none of that node's children are sold in subsequent auctions. This algorithm is representative of the outcome that would be reached if replanning (when bidding on interior nodes) were not permitted (we will refer to this algorithm as "Fixed-Tree Leaf auction" algorithm or *FTL*).

The second algorithm is a greedy algorithm (which we will refer to as *GR*) that looks at all possible OP candidates (12 per AI) and auctions non-redundant OPs off sequentially until coverage is complete. All of the potential OPs are bid on in each round, but only the OP with the lowest-cost bid is sold. There is no tree decomposition involved – all potential OPs are considered. This algorithm behaves similarly to multirobot coordination architectures that greedily allocate the best suited task to the best suited robot from a central agent (e.g. [14,8,6]).

The third algorithm (*OPT*) computes the globally optimal solution. Since the problem complexity is highly exponential, we could only compute the optimal solution for very small problem instances.

We compared the overall solution cost of the task tree auction algorithm (denoted by c_{TT}) to the solutions produced by each of the other algorithms (c_{FTL} , c_{GR} and c_{OPT}). We also compared approximations of the execution times (t_{FTL} , t_{GR} , t_{OPT} and t_{TT}) of the different algorithms. The execution times were estimated by

timestamping the start and end times of each algorithm. Results are presented in table 1.

Table 1. Experimental results: comparison of solution costs and execution times (results shown are averages of the ratio of solution costs taken over 100 runs).

Robots	Areas	c_{TT}/c_{FTL}	c_{TT}/c_{GR}	c_{TT}/c_{OPT}	t_{FTL}/t_{TT}	t_{GR}/t_{TT}	t_{OPT}/t_{TT}
2	1	.85	1.03	1.19	.21	3.9	117
4	2	.86	.92		.14	2.7	
5	3	.88	1.00		.10	2.8	
7	5	.88	1.01		.08	3.4	
4	8	.91	1.10		.06	3.1	

In terms of solution cost, the task tree algorithm is 10-15% better than the *FTL* algorithm. One reason for this is that *TT* allows distributed replanning, so the *TT* solution is intuitively expected to be no worse than the *FTL* – in the worst case no replanning is done by the task tree algorithm and the original tree decomposition is preserved⁴. In addition, the task tree algorithm also has an advantage because it allows reallocation through task subcontracting, permitting agents to discard tasks that they may have been too hasty in purchasing earlier on. It should also be noted that if the solutions are, as we suspect, close to optimal, then a 10-15% margin of improvement is significant. We can see this in the 2-robot 1-area case in which we were able to compute the optimal solution. Here the task tree algorithm was only 19% worse than optimal, as compared to *FTL* which was almost 40% worse than *OPT*. The execution time listed for the task tree algorithm (t_{TT}) is the time taken to reach the local minimum cost; although *FTL* appears to run much faster, *TT* is an anytime algorithm and often reaches a lower cost than *FTL* long before it reaches its local minimum.

On average, the task tree algorithm and the *GR* algorithm produce about the same solution quality; however, the task tree algorithm is faster and does not rely on a central auctioneer. The execution time for the task tree algorithm shown in table ?? reflects the time taken to reach the locally optimal solution. Though *TT* found its local optimum three to four times faster than *GR*, the task tree algorithm produced a feasible solution in an even shorter time and improved that solution until it reached equilibrium at the time reflected in the table. The task tree algorithm guides the search quickly through a reduced search space without compromising the solution quality, while also allowing for a distributed search.

Table 2 shows the effects of allowing auction winners to replan abstract tasks that they have won. Task tree auctions were run twice on each of 100 instances. In one run task decomposition was performed only once at the beginning by the initial

⁴ *TT* almost always performed better than *FTL*, but there were a few exceptional cases where *FTL* did slightly better: since the task tree algorithm is a local search based on myopic cost estimates, the solution reached can depend on the order the tasks are allocated to each robot.

auctioneer – tasks were not permitted to be decomposed once the original plan was developed. In the second case, the full task tree algorithm was used, allowing further decompositions after abstract tasks exchanged hands. Table 2 compares the solution costs from these two scenarios. The improvement ranged between 3 and 10% of total cost. Intuitively, we would expect the solution to be more efficient when allowing full decomposition; however, the magnitude of the improvement will depend on the specifics of the application, such as the size of the grid used and how many alternative plans the auctioneers choose to offer in the initial decomposition.

Table 2. Experimental results: comparison of solution quality of TT algorithm with replanning vs. the solution quality of TT algorithm without replanning (results shown are averages of the ratio of the solution cost with replanning compared to the solution cost without replanning).

Robots	Areas	$c_{replan}/c_{no-replan}$
2	1	.97
4	2	.96
5	3	.90
7	5	.91
4	8	.97

6 Conclusions

We have introduced a new method for distributing execution and planning over a team of robots and software agents, which combines ideas from hierarchical planning with a market-based multirobot coordination architecture. Empirical results in computer simulation show that task tree auctions can produce cost-efficient solutions to difficult optimization problems in a time-efficient manner.

In future work we will look at extending the task description language to handle richer task constraints and interactions, such as partial order precedence relations and conflicts arising between tasks competing for the same resources. Another interesting issue to explore will be decisions on when decomposition should take place. In some instances it can be costly to decompose abstract tasks, so it may be beneficial to leave the decomposition to the buyers of the task, or until immediately before the task must be executed. This would require the ability to adequately reason about the costs and benefits of tasks at an abstract level. We are also working on alternative auction clearing algorithms that will allow the bidders to use a more expressive bidding language, leaving the burden of selecting which of a robot's bids to accept to the auctioneer. Additionally, we would like to address other issues such as further generalization of the tree structures, permitting commitments to be broken and subcontracts to be sold, and how to efficiently deal with replanning when new information is discovered that affects upcoming plans.

We are currently performing further experiments and are in the process of porting the system to a team of 10 ActivMedia Pioneer P2DX robots. The hardware and software infrastructure is in place and first results are expected in the very near future.

Acknowledgments

This work was sponsored by the U.S. Army Research Laboratory, under contract **Robotics Collaborative Technology Alliance** (contract number DAAD19-01-2-0012). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The authors also thank Bernardine Dias for helpful contributions and advice.

References

1. B. J. Clement and E. H. Durfee. Top-down search for coordinating the hierarchical plans or multiple agents. In *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 252–259, 1999.
2. M. B. Dias, D. Goldberg, and A. Stentz. Market-based multirobot coordination for complex space applications. In *The 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2003.
3. M. B. Dias and A. Stentz. A free market architecture for distributed control of a multi-robot system. In *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, pages 115–122, 2000.
4. M. B. Dias and A. Stentz. Opportunistic optimization for market-based multirobot control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.
5. K. Erol, J. Hendler, and D. S. Nau. Semantics for hierarchical task-network planning. Technical Report CS-TR-3239, University of Maryland College Park, 1994.
6. B. P. Gerkey and M. J. Matarić. Sold!: Auction methods for multi-robot control. *IEEE Transactions on Robotics and Automation Special Issue on Multi-Robot Systems*, 18(5):758–768, October 2002.
7. L. Hunsberger and B. J. Grosz. A combinatorial auction for collaborative planning. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS)*, 2000.
8. L. Parker. Adaptive heterogeneous multi-robot teams. *Neurocomputing, special issue of NEURAP '98: Neural Networks and Their Applications*, 28:75–92, 1999.
9. G. Rabideau, T. Estlin, S. Chien, and A. Barrett. A comparison of coordinated planning methods for cooperating rovers. In *Proceedings of the AIAA 1999 Space Technology Conference*, 1999.
10. M. H. Rothkopf, A. Pekec, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
11. T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1–54, 2002.
12. R. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12), 1980.
13. A. Stentz and M. B. Dias. A free market architecture for coordinating multiple robots. Technical Report CMU-RI-TR-99-42, Robotics Institute, Carnegie Mellon University, December 1999.
14. S. Thayer, B. Digney, M. B. Dias, A. Stentz, B. Nabbe, and M. Hebert. Distributed robotic mapping of extreme environments. In *Proceedings of SPIE: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII*, 2000.

15. R. Zlot and A. Stentz. Multirobot control using task abstraction in a market framework. In *Proceedings of the Collaborative Technology Alliances Conference*, 2003.
16. R. Zlot, A. Stentz, M. B. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proceedings of the International Conference on Robotics and Automation*, 2002.

Decentralised SLAM with Low-Bandwidth Communication for Teams of Vehicles

Eric Nettleton¹, Sebastian Thrun², Hugh Durrant-Whyte¹, and Salah Sukkarieh¹

¹ Australian Centre for Field Robotics, The University of Sydney, NSW, 2006, Australia
ericn, hugh, salah@acfr.usyd.edu.au

² Carnegie Mellon University, Pittsburgh, PA, 15213, USA

Abstract. This paper addresses the problem of simultaneous localization and mapping (SLAM) for teams of collaborating vehicles where the communication bandwidth is limited. We present a novel SLAM algorithm that enables multiple vehicles to acquire a joint map, but which can cope with arbitrary latency and bandwidth limitations such as typically found in airborne vehicle applications. The key idea is to represent maps in information form (negative log-likelihood), and to selectively communicate subsets of the information tailored to the available communication resources. We show that our communication scheme preserves the consistency, which has important ramifications for data association problems. We also provide experimental results that illustrate the effectiveness of our approach in comparison with previous techniques.

1 Introduction

The problem of simultaneous localisation and mapping (SLAM) has received significant attention in the past few years. While almost all research is focused on the single vehicle case, the multi-vehicle case has almost entirely been overlooked. This is at odds with the fact that many of the most promising robotics domains (military and civilian) involve teams of robots that jointly acquire maps [9,11,1]. Decentralised SLAM addresses the problem in which large numbers of vehicles cooperatively acquire a joint map, while simultaneously localising themselves relative to the map. For a decentralised architecture to be truly scalable, the individual communication requirements must be independent of the number of vehicles in the system. For it to be robust, it must be able to accommodate arbitrary latencies and bandwidth limitations in the communication network.

Most conventional SLAM algorithms are based on a seminal paper by Smith and Cheeseman [10]. This paper introduced the Extended Kalman filter (EKF) solution to the SLAM problem, which has been the basis of hundreds of contemporary implementations [5]. The EKF represents maps by the mean and covariance of a Gaussian distribution. Unfortunately, the communication of these entire maps for multi-vehicle problems is not scalable for large map sizes, as it requires the transmission of all N^2 elements in the covariance matrix to remain consistent.

This paper presents a new algorithm for multi-vehicle SLAM. Just as in our previous work [8], we use the information form of the EKF to represent information acquired by the vehicle. The update of the information form is additive; as a

result, incremental new information can be integrated across different vehicles with arbitrary network latencies. However, our previous work required communication bandwidths *quadratic* in the size of the map. This paper proposes an efficient communication scheme which makes it possible to communicate updates whose size is independent of the size of the map. By doing so, our approach can cope with arbitrary bandwidth limitations in multi-vehicle SLAM. Our approach is fully decentralised and can (in principle) scale to any number of vehicles.

The ability to communicate maps of arbitrary size is obtained by implementing a hybrid information filter/Covariance Intersect (CI) [4] algorithm on each communication link. This algorithm is implemented separately from the SLAM filter and is used solely to manage the inter-vehicle communication and ensure that information vehicles have shared does not get ‘double counted’. The formulation of the problem in information space is also exploited by using information metrics to maximise the amount of information in any communicated submap. Our approach is to communicate those features with the greatest information gain that has not yet been sent. The simulated results included in this paper illustrate that this algorithm functions both consistently and accurately.

The work described in this paper is part of the Autonomous Navigation and Sensing Experimental Research (ANSER) project and is aimed at the development of a multiple flight vehicle demonstration of decentralised SLAM. The system under development consists of four unmanned flight vehicles, each equipped with GPS and inertial sensors and two terrain payloads; a vision system and either a mm-wave radar or laser sensor. However, to implement decentralised SLAM on this system it is necessary to have an algorithm which can operate in a bandwidth constrained environment. The approach presented here overcomes this important obstacle and provides a technique that is scalable.

2 The Decentralised Architecture

The decentralised architecture used in this research is based on the information or inverse covariance form of the Kalman filter, and builds on the work of Grime [2] on decentralised tracking. It works by propagating ‘information’ to all vehicles or nodes in the network. The internal structure of each of these sensing nodes is illustrated in Figure 1.

This section gives an overview of the key elements of the architecture. A more detailed description can be found in [7].

2.1 The Information Filter

Using the conventional EKF representation of SLAM, the augmented state vector of the vehicle and map at time t_i given observations up to time t_j is written as $\hat{\mathbf{x}}(i | j)$ with an associated covariance $\mathbf{P}(i | j)$. However, when written in its equiv-

alent information form, the system is given by the information vector $\hat{\mathbf{y}}(i | j)$ and information matrix $\mathbf{Y}(i | j)$.

$$\hat{\mathbf{y}}(i | j) \triangleq \mathbf{P}^{-1}(i | j) \hat{\mathbf{x}}(i | j), \quad \mathbf{Y}(i | j) \triangleq \mathbf{P}^{-1}(i | j).$$

A complete derivation of the non-linear information filter, its propagation and update equations and its application to SLAM is contained in Thrun et al [12]. Maybeck [6] also includes a derivation of the information filter from the standard Kalman filter equations.

The interesting fact is that with regards to information integration, all update operations are *additive*. Addition is communicative. As a result, the specific order in which updates from other vehicles are applied is irrelevant to the results, provided the features do not change over time. This observation is of central importance in multi-vehicle SLAM, as the map features are specifically selected to be stationary. If map update messages from other vehicles arrive under arbitrary latency, they can still simply be added on in the information form regardless of their “age.”

2.2 Local Filter

The local filter is the local implementation of the SLAM algorithm, which generates information state estimates on the basis of observed, predicted and communicated information. Other infrastructure such as the channel filter and channel manager exist only to support the communication of information to and from other vehicles.

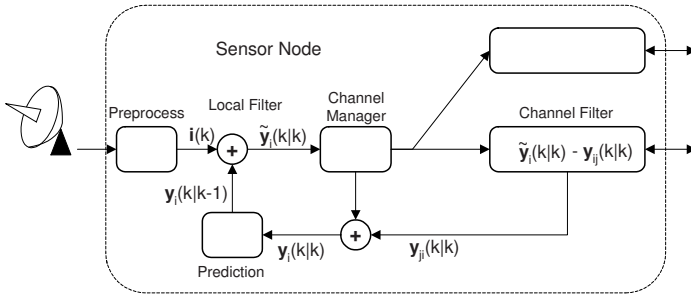


Fig. 1. The internal structure of a decentralised node

2.3 Channel Manager

The channel manager serves as the interface between the local filter and the channel filters (and through these, the other vehicles in the network). The channel manager collects incoming data from the channels at the time horizon, assimilates it using the additive update equations, and then communicates the result to the local filter which

can update the SLAM estimate in a single step. It also receives outgoing updated information states from the local filter and disseminates this to the channel filters for transmission.

2.4 Channel Filter

The channel filter is used to manage communication of map information between nodes, and as such is the focus of the algorithm presented in this paper. It serves two main functions; to keep track of information previously communicated on the channel (to ensure data is not double counted), and to synchronize incoming and outgoing information with the local SLAM filter. Information previously communicated is used to compute new information gain from other vehicles in the network. Synchronization serves to accommodate any delays in information or differences in timing between node filters at remote sites.

Information arrives asynchronously at each channel from remote vehicles. The channel filter calculates the new information received on any given channel and transmits this to the channel manager, before updating itself. In the event that the channel becomes blocked or disconnected, the channel filter effectively fuses the new data and cycles to the next available communication time.

The following section discusses the channel filter algorithm and how it can be formulated to handle the communication of submaps of an arbitrary size.

3 Constant Time Communication

In SLAM, updates come in two forms: measurement and motion updates. While each measurement update affects only a small number of values in the information form, motion updates modify *all* such values. Thus, after each motion command, previous techniques communicate all values in the filter, of which there are quadratically many in the size of the map. Such an approach imposes serious scaling limitations on multi-vehicle SLAM.

Our approach communicates arbitrarily small maps by carving out submaps of maximum information value. The idea is that the vehicle operates in the vicinity of specific landmarks in the map; hence communicating the submap that corresponds to those landmarks yields small messages whose size is independent of the total map size.

Unfortunately, such a scheme will inevitably lead to over-confident estimates if implemented using a standard information or Kalman filter: this is a side effect of the approximation of sending submatrices of the full update (which affects all parameters in the filter). Our new approach composes such sub-matrices in ways that maintain *map consistency*, that is, they guarantee that no vehicle ever overestimates its confidence in individual feature locations. This result is obtained by implementing a hybrid information filter/covariance intersect (CI) [4] algorithm in the communication channels. As the vehicles are communicating their information state estimates of selected map features, they will have information in common which

correlates them. In order to cope with this issue, an information estimate is maintained in the communication channel which keeps a record of this ‘common information’ separately from the SLAM filter. When information submaps arrive from another vehicle, they are fused using CI with this common information to yield a consistent, but conservative update. Since the update of an information filter is additive, the effective increment of new information that the communicated submap contributed is given by the difference between the channel estimate before and after the CI update. This increment of map information can then be safely added to the SLAM filter, completing the update cycle. It is important to note that the use of the conservative CI update is limited strictly to the communication algorithm, while the SLAM estimator is implemented using the information form of the Kalman filter.

Another key aspect of this algorithm is that the submaps being communicated are not static submaps such as those used to aid computational efficiency, but instead are a dynamically selected set of features chosen at each communication step based on some heuristic. It is this ability to formulate submaps of arbitrary size that allows the communication strategy to operate in constant time. Also, by formulating the problem in its information form, it is a simple matter to maximise the amount of information being transmitted in any given submap by selecting features which have experienced the greatest information gain.

The ability to communicate constant time maps is also achievable using other approaches such as that proposed by Williams [13]. However, this approach is not robust to communications failure as it essentially transmits increments of map data. If any increment is lost, it is irrecoverable. When formulated in the information form presented here it is possible to transmit the actual state estimate that contains all prior information about the map features. Using this approach, each communication automatically contains all information in prior messages as well as any new data. In order not to double count data, the channel filter is used at reception to subtract the common information and pass only the increment of new information to the SLAM filter.

The Split Covariance Intersect (SCI) algorithm proposed by Julier [3] is similar to the approach presented here. It uses a hybrid CI/Kalman filter approach to manage the complexity of large maps, and in the process notes that the same approach can be extended to multiple vehicles. However, as the SCI method does not use the channel filter concept to track any information nodes have in common, it is not able to make total use of information should complete maps ever be transmitted.

3.1 Extracting the Submap to Communicate

Our approach to selecting features to communicate is to dynamically select those which *maximise the information gain* down any particular channel. This is done using the features that have the greatest amount of information that has not yet been communicated to other nodes. As the problem is already formulated in information space, this is a trivial exercise and can be done simply by subtracting the amount of information transmitted (recorded in the channel filter) from the current information state. A submap of some arbitrary size M^2 , where $0 \leq M \leq N$, can then

formed using the M features with the *greatest information gain*. This ability to form an arbitrary size submatrix to communicate is a powerful concept as it allows the algorithm to function without needing to communicate the entire map of size N^2 . While it is possible to use practically any method of selecting the features to send, the ‘maximum information gain’ technique is used in this paper with highly satisfactory results.

Once the features to transmit have been selected, it is necessary to integrate out the effect of all of the other states from this submap. This can be done by defining a projection matrix G_m such that $G_m \hat{\mathbf{y}}(k | k)$ extracts only those features in the map which are to be transmitted, and another projection G_v such that $G_v \hat{\mathbf{y}}(k | k)$ contains all other states. Using the notation $\mathbf{y}^*(k | k)$ and $\mathbf{Y}^*(k | k)$ to define the information vector and information matrix of the submap to be sent, we integrate out the effect of the other states:

$$\begin{aligned}\mathbf{y}^*(k | k) &= G_m [\hat{\mathbf{y}}(k | k) - \mathbf{Y}(k | k) G_v^T (G_v \mathbf{Y}(k | k) G_v^T)^{-1} G_v \hat{\mathbf{y}}(k | k)] \\ \mathbf{Y}^*(k | k) &= G_m [\mathbf{Y}(k | k) - \mathbf{Y}(k | k) G_v^T (G_v \mathbf{Y}(k | k) G_v^T)^{-1} G_v \mathbf{Y}(k | k)] \\ &\quad \times G_m^T\end{aligned}\quad (1)$$

The small information submap given by $\mathbf{y}^*(k | k)$ and $\mathbf{Y}^*(k | k)$ is now sent to the channel filters prior to transmission to another node.

3.2 Channel Update

As the channel filter maintains an estimate of the common information between nodes, the only states it will have are those that have been communicated. Since the vehicle information is never communicated, the channel filters will never maintain any states other than the map. Therefore, as map features are by definition stationary, the prediction model of the channel filter is an identity matrix and the state at time t_{k+1} is equal to the state at t_k .

If the full map of N^2 features is being transmitted, the channel filter can be updated simply using:

$$\begin{aligned}\mathbf{Y}_{Chan}(k | k) &= \mathbf{Y}_{Chan}(k | k - 1) + [\mathbf{Y}^*(k | k) - \mathbf{Y}_{Chan}(k | k - 1)] \\ &= \mathbf{Y}^*(k | k) \\ \hat{\mathbf{y}}_{Chan}(k | k) &= \hat{\mathbf{y}}_{Chan}(k | k - 1) + [\mathbf{y}^*(k | k) - \hat{\mathbf{y}}_{Chan}(k | k - 1)] \\ &= \mathbf{y}^*(k | k)\end{aligned}\quad (2)$$

This simple update is only possible as the map information $\mathbf{Y}^*(k | k)$ and $\mathbf{y}^*(k | k)$ are of the dimension of the complete map and all cross information terms are being transmitted. For the constant time communication algorithm this is not the case and a different update must be performed.

The channel filter update when sending submaps of arbitrary size is performed using CI. While this method will yield a conservative result, it is necessary in the absence of the complete N^2 map information estimate. When the information submap

given by $\mathbf{y}^*(k | k)$ and $\mathbf{Y}^*(k | k)$ arrives the CI update of the channel filter can be done by inflating the submap to the state dimension using an appropriate projection G_m . Selection of the CI weighting parameter ω is done to minimise the determinate of the result.

$$\begin{aligned}\mathbf{Y}_{Chan}(k | k) &= \omega \mathbf{Y}_{Chan}(k | k - 1) + (1 - \omega) G_m^T \mathbf{Y}^*(k | k) G_m \\ \hat{\mathbf{y}}_{Chan}(k | k) &= \omega \hat{\mathbf{y}}_{Chan}(k | k - 1) + (1 - \omega) G_m^T \mathbf{y}^*(k | k)\end{aligned}\quad (3)$$

Following this process the channel filter is completely updated and the information submap given by $\mathbf{y}^*(k | k)$ and $\mathbf{Y}^*(k | k)$ is transmitted to other nodes. It is important to note that this submap contains the entire state history of those features in it and not simply an increment of information or a batch of observations. In formulating the problem this way, the algorithm is robust to any communication failures as every message contains not only new information, but all information in previous messages as well. As the information from any previous messages is stored in the channel filter as common information, it can be removed easily by the receiving node to ensure that information is not fused more than once. For example, if a particular submap of features is transmitted but for some reason never received, the information in the lost message is automatically recovered next time those features are communicated.

When the receiving node gets the new submap information, it updates its own channel filter using exactly the same update steps as above. Once updated, it calculates the increment of new information it has just received from node i that has not already been fused locally at node j .

$$\begin{aligned}\mathbf{Y}_{ij}(k | k) &= \mathbf{Y}_{Chan}(k | k) - \mathbf{Y}_{Chan}(k | k - 1) \\ \hat{\mathbf{y}}_{ij}(k | k) &= \hat{\mathbf{y}}_{Chan}(k | k) - \hat{\mathbf{y}}_{Chan}(k | k - 1)\end{aligned}\quad (4)$$

This information increment is then sent to the local filter to be fused into the SLAM estimate.

3.3 Fusing Information from Other Nodes at the Local Filter

When the local filter receives the information $\mathbf{Y}_{ij}(k | k)$ and $\hat{\mathbf{y}}_{ij}(k | k)$ from the channel filter it must use this information in the SLAM estimate. In order to do this, it is necessary to firstly define a matrix G_s that inflates the map to the dimension of the entire SLAM state by padding vehicle elements with zeros.

The update is then done by adding the new information from the other node as:

$$\begin{aligned}\hat{\mathbf{y}}(k | k) &= \hat{\mathbf{y}}(k | k - 1) + G_s \hat{\mathbf{y}}_{ij}(k | k) \\ \mathbf{Y}(k | k) &= \mathbf{Y}(k | k - 1) + G_s \mathbf{Y}_{ij}(k | k) G_s^T\end{aligned}\quad (5)$$

It is worth noting that this update step is identical to updating with information from locally attached sensors.

4 Results

The constant time communications algorithm was implemented and run in a multi-vehicle simulation to test and evaluate its performance with respect to a number of other communication strategies. The different communications strategies used were:

1. No communication between platforms - each vehicle operates independently using only its own observations.
2. Transmission of the complete information map of N^2 elements at every communication step.
3. The constant time communications strategy where a submap of only 5 features were transmitted at each communication step. The features to include in the transmitted submap were selected at transmission time to be those which had the greatest amount of information not yet sent.

The simulation used 100 vehicles moving in an environment of 100 features. Figure 2 (h) illustrates this simulation world, marking the feature locations and a number of the vehicle paths. A non-linear tricycle motion model was implemented to estimate the position and orientation $[x \ y \ \phi]$ of each vehicle within the map. Each was equipped with a range/bearing sensor which gave observations to features in the forward hemisphere of the vehicle at a frequency of 1Hz. Vehicles were able to communicate map information to their neighbours in the network at a frequency of 0.1Hz.

The results of the simulation are illustrated in Figure 2. The benefit of communicating information can be seen in Figure 2 (a)-(g) as the error and 2σ bounds of both the N^2 and constant time communication algorithms are significantly better than the case when the vehicles do not communicate at all. Of the communication schemes, the N^2 performs the best as it always propagates the entire map. However, the constant time communications algorithm can be seen to give results that are only slightly worse than this. Furthermore, it does this by transmitting submaps of at most 5 features each communication step, instead of all 100. Since the number of elements required for communication is quadratic in the size of the map, the bandwidth savings for this problem are significant when using the constant time method.

The results indicate that the constant time algorithm approaches the N^2 strategy quite quickly even though it is not transmitting as much information. This occurs as the submap that is transmitted is dynamically selected to contain those features which will give the greatest information gain. This trend illustrates the idea that good information about a small number of features increases the map accuracy significantly. This concept is further illustrated in Figure 2 (i) which shows that although increasing the size of the communicated submaps does increase the quality of the estimates, the rate at which the quality changes decreases as the submaps get larger. This is to be expected as communication of the whole map will include features that have not been observed recently and which will therefore not contribute large amounts of new information. Communicating larger submaps using this con-

stant time algorithm simply has the effect allowing the estimate to converge toward the N^2 result faster.

5 Conclusion

This paper has presented an algorithm for constant time communications in multiple vehicle SLAM problems. Using this method, the communication requirements for the decentralised system are independent of the number of features in the SLAM map. It was shown that the effect of communicating these smaller submaps maintains map consistency while providing only slightly less information than transmitting the entire map. However, should there be an opportunity to transmit the complete map of N^2 features, the algorithm is able to make complete use of the information and recover any information discarded through previous conservative CI updates.

References

1. W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2000.
2. S. Grime and H.F. Durrant-Whyte. Data fusion in decentralized sensor networks. *Control Engineering Practice*, 2, No 5:849–863, 1994.
3. S. Julier and J. Uhlmann. Building a million beacon map. In *Sensor Fusion and Decentralised Control in Robotic Systems IV*, volume 4571, pages 10–21, 2001.
4. S. Julier and J. Uhlmann. General decentralised data fusion with covariance intersection (ci). In D. Hall and J. Llinas, editors, *Handbook of Data Fusion*. CRC Press, 2001.
5. J. Leonard, J.D. Tardós, S. Thrun, and H. Choset, editors. *Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots (W4)*. ICRA Conference, Washington, DC, 2002.
6. P.S. Maybeck. *Stochastic Models, Estimation and Control, Volume 1*. Academic Press Inc., New York, 1979.
7. E. Nettleton. *Decentralised Architectures for Tracking and Navigation with Multiple Flight Vehicles*. PhD Thesis, The University of Sydney, 2003.
8. E.W. Nettleton, P.W. Gibbens, and H.F. Durrant-Whyte. Closed form solutions to the multiple platform simultaneous localisation and map building (SLAM) problem. In *Proc. SPIE*, volume 4051, pages 428–437, 2000.
9. R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proceedings of the AAAI National Conference on Artificial Intelligence*. AAAI, 2000.
10. R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.
11. C. Thorpe and H. Durrant-Whyte. Field robots. In *Proceedings of the 10th International Symposium of Robotics Research (ISRR'01)*, 2001.
12. S. Thrun, D. Koller, Z. Ghahmarani, and H. Durrant-Whyte. SLAM updates require constant time. In *Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics*, Nice, France, 2002.
13. S. Williams. *Efficient Solutions to Autonomous Mapping and Navigation Problems*. PhD Thesis, University of Sydney, 2001.

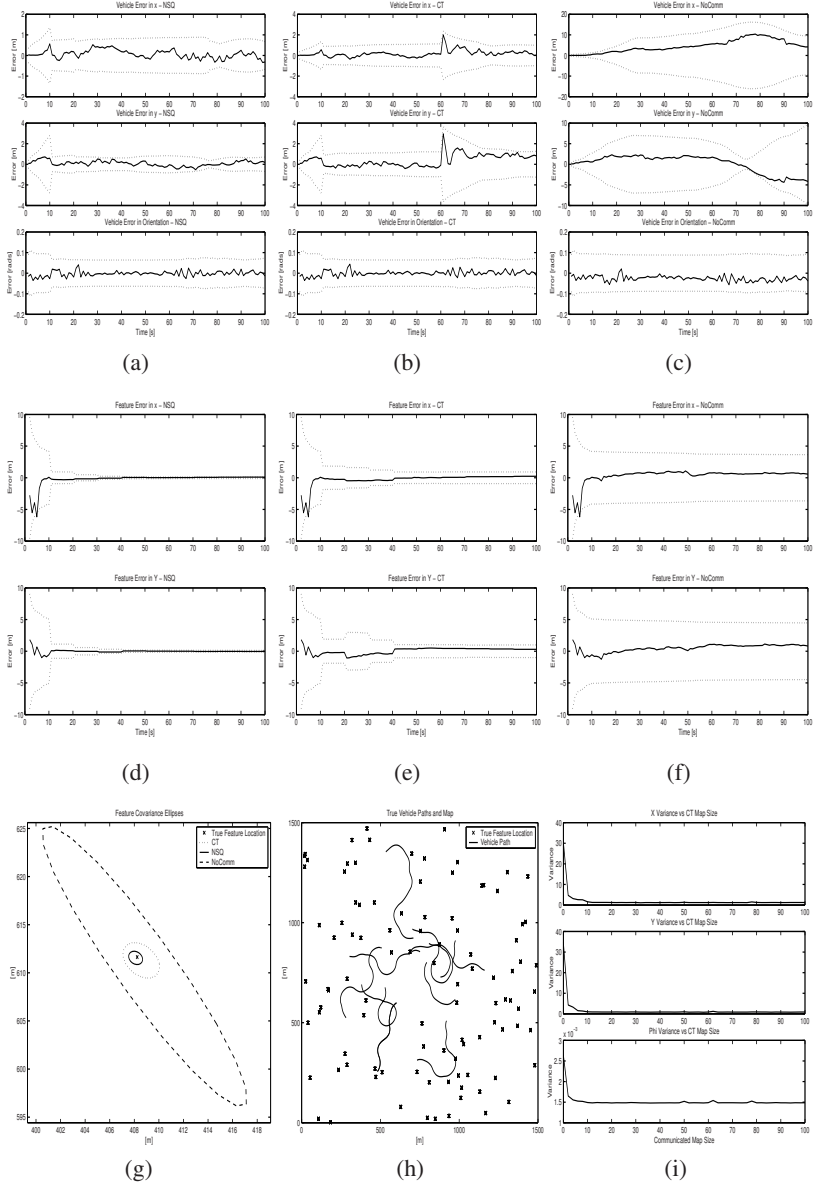


Fig. 2. Simulation results. The error and 2σ bounds for the vehicle states are given for the (a) N^2 , (b) constant time and (c) no communications strategies. The error and 2σ bounds for a typical feature are also given for the (d) N^2 , (e) constant time and (f) no communications strategies. The covariance ellipse for a feature under the different communication strategies is illustrated in (g). The simulation world is shown in (h). For clarity, only 10 of the 100 vehicle paths are shown. (i) shows the mean variance in x, y and ϕ from all vehicles over the period of the simulation for different maximum map sizes using the constant time communications algorithm.

Case Studies of a Borehole Deployable Robot for Limestone Mine Profiling and Mapping

Aaron Morris¹, Derek Kurth¹, Daniel Huber¹, Warren Whittaker², and Scott Thayer¹

¹ Robotics Institute: Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{acmorr, dekurth, dhuber, sthayer}@cs.cmu.edu

² Workhorse Technologies, LLC.
484 West 7th Avenue
Homestead, PA 15120
chuck@workhorsetech.com

Abstract. Inherent dangers in mining operations motivate the use of robotic technology for addressing hazardous situations that prevent human access. In the context of this case study, we examine the application of a robotic tool for map verification and void profiling in abandoned limestone mines for analysis of cavity extent. To achieve this end, our device enables remote, highly accurate measurements of the subterranean voids to be acquired. In this paper we discuss the design of the robotic tool, demonstrate its application in void assessment for prevention and response to subsidence, and present results from a case study performed in the limestone mines of Kansas City, Kansas.

1 Introduction

The existence of subterranean void spaces, such as the cavities created by mining, is a hazard to active mining operations and a constant threat to surface developments. When abandoned, these underground spaces can accumulate tremendous quantities of water and threaten to flood encroaching active mines (as occurred in Queecreek, PA on July 2002 [1]) or subside to form sinkholes on the surface [2]. Although both hazards have devastating consequences if left unchecked, subsidence is the major concern for aboveground development since it degrades the structural integrity of the overlaying land. With an estimated 500,000 abandoned mines in the United States, subsidence is a formidable problem for community and city infrastructure [3].

In response to the situation, a remediation technique was developed to restore the structural integrity of land situated above a domeout. Utilizing flyash (a waste product from coal fired electric generating plants) and water, a cement-like slurry is created. This slurry is poured into domeouts through boreholes (Figure 1). Once completely backfilled and cured, the domeout is considered stable [4]; however, a successful backfill operation is only achieved with the proper preparation and mine information. The underlying challenges posed by backfilling are (1) obtaining reliable void dimensions and volume estimations, (2) verification and identification of domeout location on the mine map, and (3) registration of the mine map in a surface coordinate frame.

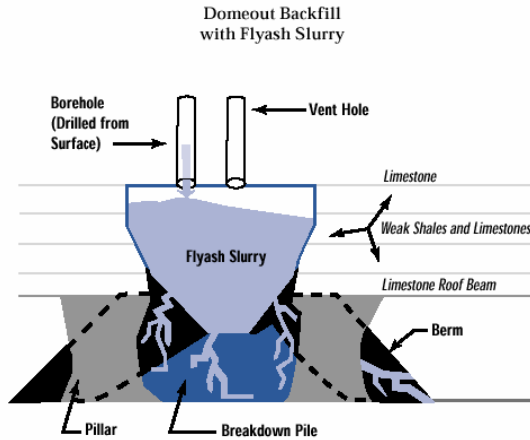


Fig.1. The process of backfilling a domeout with coal flyash [4].

2 Related Methods

Current methods for underground void detection include non-intrusive techniques such as ground-penetrating radar (GPR) and microgravity as well as direct methods such as borehole-deployed cameras and human surveying. With GPR, a surveyor acquires information about subsurface structures by analyzing the response of high frequency electromagnetic waves propagated through the ground. These waves are reflected back to the surface when they encounter a sharp change in the electrical properties of underground matter, and this response can be analyzed to identify voids or other features of interest. Due to attenuation of the EM waves, GPR is typically limited to depths of no more than 30m, with scan resolution worsening proportionate to scan depth, making this method appropriate chiefly for shallow investigations [5,6].

Microgravity techniques require careful measurement of variations in the Earth's gravitational field, which can be used to infer void presence, depth, and shape. Again, while non-intrusive and non-destructive like GPR, this method is viable only to depths on the order of tens of meters [7].

When human access is possible, inspectors may explore and evaluate mine remnants, but this may pose great personal risk to the surveyor, particularly when surveying around domeouts or other collapses. When conditions preclude human access, remote cavity detection can be obtained by borehole drilling. A hole is bored from the surface to an assumed mine cavity at a known depth and map information is inferred by the presence or absence of ground materials. To augment borehole findings, cameras may be deployed to visually affirm the presence of void space in situations where a cavity is found [8, 9]. The caveat with borehole and camera techniques is that they provide limited information. Qualitative range data (distance measurements) cannot be extracted and void extent

is not explicitly measurable. As a result, boreholes must be densely distributed across the encompassing mine area, requiring a substantial investment of time, money, and resources.

3 Platform Design

The robotic tool we developed for remote subterranean void analysis has several operational advantages over past techniques. Nicknamed Ferret, it establishes a physical presence in a mine cavity enabling a “first hand” perspective of the void (unlike non-intrusive methods where information is inferred). This physical presence is attained without human contact in or around the mine, which removes the risks faced by surveyors in subterranean inspections. In addition, Ferret provides quantitative information on cavity extent that is difficult or impossible to obtain using borehole camera systems. In the following sections, we describe the mechanical, electrical, and software systems that compose the Ferret and empower it to retrieve cavity data in a rugged and unforgiving environment.

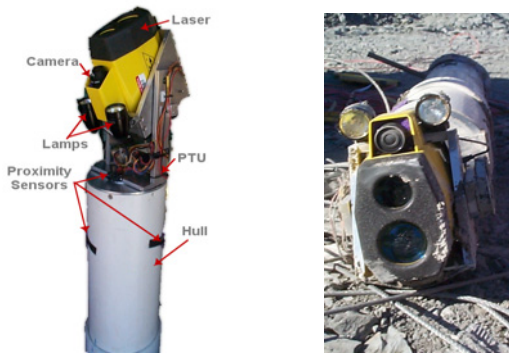


Fig.2. (Left) Side-view of Ferret. (Right) Ferret after 2nd deployment.

3.1 Mechanical System

The Ferret device is a relatively simple mechanism that provides extreme robustness and reliability in harsh environmental conditions. Ferret’s primary mapping sensor is a single-point, long-range, low-reflectivity laser that performs measurements on the cavity enclosure. To obtain a semi-spherical scan of the void, the laser is actuated on a pan and tilt unit (PTU) with an effective motion range of 360° in the horizontal plane and 150° in the vertical plane. The PTU is driven with two 12 Vdc motors that independently control each degree of freedom. The motors are highly geared to minimize backlash and maximize the accuracy of angular displacement.

Above the laser and PTU is a cylindrical hull constructed of 6” PVC pipe that encapsulates sensitive computing and sensing modules. At the upper end of the

hull, a thick aluminum plate provides an interface connector and support bar. The interface connection enables a communications and power linkage between Ferret and the surface control station. The support bar connects Ferret to a 3/16 in. diameter steel cable that suspends the device from the surface during an operation. This cable is drawn through a motor-driven winch and pulley system for deployment and retrieval.

3.2 Electrical and Sensing System

As previously mentioned, the primary mapping sensor utilized on Ferret is a surveying laser manufactured by the Atlanta Optics CompanyTM. The laser is capable of range measurements in excess of 50m on materials found in mines like coal, limestone, shale, and bedrock. The laser is equipped with a magnetic compass and a two-axis tilt sensor to detect pitch and roll. A low-light camera is mounted alongside the laser with two small lamps to allow visualization of Ferret's progress. Furthermore, the camera provides visual feedback to assist human operators in teleoperation of Ferret.

Just above the laser, an analog IR proximity sensor is mounted to the PTU frame to provide distance measurements at small ranges (12 cm) perpendicular to Ferret's hull. The limited measuring range serves as a breach detection system that signals to the command station when the cavity opens beyond the sensors perceptive range. Two other digital IR sensors are mount along the hull's side to assist in determination of device depth.

Ferret is also equipped with encoders, magnetic homing switches, and a hull-mounted compass. These sensors provide Ferret and its laser orientation information relative to a global bearing (i.e. the Earth's magnetic field) as well as means of control.

Ferret's central computing is located within the hull and consists of an embedded 18.432 MHz Rabbit 2000 microprocessor. This processor is responsible for coordination of actuator controller modules, sensor interfaces, and data relay to a computer located on the surface. Communication between the 8-bit core and the surface computer is handled through an Ethernet connection.

3.3 Software System

Computation for Ferret is distributed among the several processors located internal and external to the device. Within Ferret, JR Kerr motor controllers drive the PTU actuators; a laser controller issues averaged distance and orientation measurements; and sensor data is routed to interface ports. All these controller and IO devices communicate with the Rabbit core over serial ports and each device is poled for incoming data. If a message is received, the data is extracted and utilized for mechanism control and relayed to the command station.

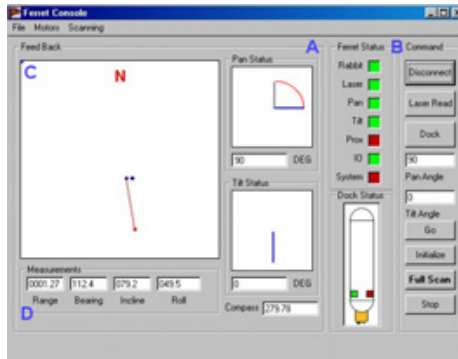


Fig.3. Ferret User Control Interface.

Above low-level data flow and controller communication, procedures are defined that examine laser orientation and distance measurements to manipulate angular velocity. In scanning mode, Ferret moves the within specified angular bounds (defined from the command station) by monitoring the pan and tilt angular positions. Encountering a pan bound switches the direction of rotation and detecting a tilt bound terminates the scan. Furthermore, the speed of rotation is altered in proportion to the current range measurement to maintain consistent arc lengths between range readings.

On the surface, a computer command station interacts with Ferret over a TCP/IP interface. This computer contains a user control interface (UCI) that allows velocities, accelerations, angular bounds, angle positions, and scan parameters to be set by an operator. Visual indicators in the form of dials and status lights provide sensor feedback so that the surface team can gage Ferret's state throughout operation. The UCI also handles logging, provides data visualization, and allows the user to directly manipulate the PTU for laser and camera direction.

Software that performs post analysis of the data is handled offline and will be discussed in the results section of this paper.

4 Case Study Review

Over a three-day period in January 2003, Ferret was deployed into an abandoned limestone mine beneath Kansas City for void assessment. The operation was conducted at the request of the landowner who sought to permit overtop a partially backfilled limestone mine. Prior to Ferret, the primary means of gathering information was borehole logs and borehole cameras; however, these methods proved inadequate for determining void extent and unreliable for making decisions on structural integrity.

4.1 Operational Overview

Each day of the operation, Ferret scanned a separate section of the mine from one of three borehole vantages. These boreholes were approximately 500 ft. spaced from one another with an average depth of 200 ft. Scan time averaged 3.5 hours acquiring approximately 38,000 range measurements (a limit imposed by the laser controller over that time frame). In addition, site engineers actively monitored Ferret scans and video to supply real-time feedback and augment data findings.

Environmental conditions were rugged and challenged the functionality of Ferret. For example, the surface temperature was 15° F while mine temperature was approximately 50° F so that high levels of humidity led to large amounts of water vapor. As a result, dense steam columns emanated from the boreholes and instantly fogged camera and laser lenses upon immersion. This thick cloud of water vapor also moistened the soil along the perimeter of the borehole forming a viscous mud that adhered to the hull, collected in gears, and flooded all openings. Nevertheless, Ferret performed the scans and acquired void data undaunted by conditions of the borehole.

4.2 Data Analysis

Upon data acquisition, rigorous analysis was performed offline. Figure 4 shows the point-cloud and mesh created from one of the three scans. In this form, range data permitted a unique 3D geometric visualization that allowed landmark identification of limestone mine features. In this particular scan, a “cuttings” pile (a collection of soil formed during the borehole reaming process), mine columns, and a backfill slope can be seen. These features provided insight on borehole location within the mine and assisted engineers in understanding the work preformed in the previous backfill operation.

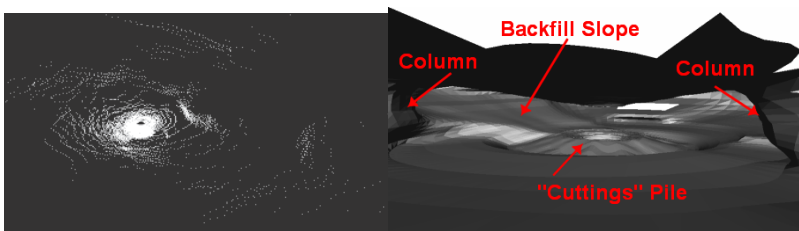


Fig.4. (Left) 3D point-cloud of mine. (Right) Mesh view from inside of mine.

Slices of this 3D data permitted dimensional analysis of the voids. Figure 5 shows plots taken from a domeout scan. The cavity size indicated no previous backfill and small pockets of long-range measurements suggested expansion into other sections of the mine. Due to its state, multiple scans were taken at varying elevations.

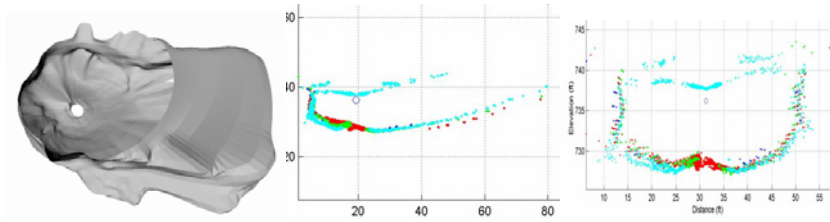


Fig.5. (Left) Domeout Scan. (Middle) Side contour plot. (Right) Front contour plot.

By varying the laser's vantage, individual scans were geometrically accurate only in local coordinate frames. Motion of the sensor between scans caused the scan sequence to be misaligned. To correct for this misalignment, these data sets were registered in a common coordinate system using an algorithm for simultaneous registration of multiple 3D data sets. For this purpose, we utilized an implementation of Neugebauer's algorithm [10] (essentially a multi-view extension of the well-known iterative closest point algorithm [11]) to facilitate registration. The algorithm repeatedly adjusts the transforms between scans, minimizing the distance between the surfaces in regions where two scans overlap. Once the scans are aligned in the same coordinate system, the combined data provides a denser and more complete model of the void (Figure 7).

For mine map correction, a combination of several image-processing techniques and constrained searching methods were used to identify scan locations on the mine map. To start, Euclidean 3D data was projected into the XY plane to match the map space and form a template that governed the search for optimal placement. These plots provided keying features where high point accumulation signified edges of objects like columns and walls and sparse accumulation indicated open space (periodic measurements from the ceiling and floor).

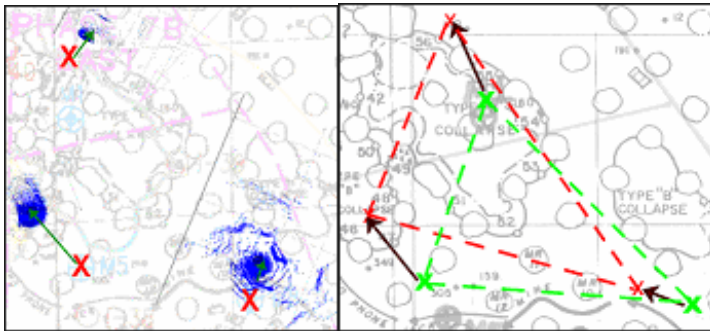


Fig.6. Mine map analysis. (Left) All three projection plots placed in same map. (Right) Calculated warp (X's mark estimated borehole placement prior to Ferret and arrows point to determined positions).

Utilizing template-matching techniques [12], column edges and solid landmarks on the mine map were correlated with high-density point clusters. To determine an optimal alignment, the search sought to

1. Minimize the number of points lying inside of solid structures
2. Minimize the number of rays from the laser origin to range points that intersected solid structures
3. Maximize the number of points that reside in close proximity to the border of solid structures

In addition, prior compass readings, estimates of borehole locations, and identified features from the 3D models were assimilated to reduce the search space. Figure 6 shows the projection plot and its correlated position overlaid onto the mine map.

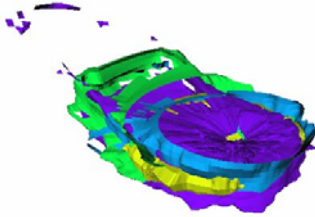


Fig.7. Registration of multiple 3D laser scans.

To conclude mine map analysis, the coordinate frame of the map was registered to surface (or global) coordinates. Using the global position of each borehole's surface-side location (obtained from a survey team and GPS positioning), the corresponding mine map was warped into place (Figure 6).

5 Summary and Conclusions

In summary, the full operation (comprised of three boreholes) consumed three days of fieldwork and three weeks of analysis. Hole 1 was situated over a partially backfilled portion of the mine on the southeastern side of the site. The gap height was approximately 3 ft. and range measurements extended out to 90 ft. Hole 2 was located in a domeout in the western section of the site with no previous backfill. The domeout was roughly 60 ft. wide, 95 ft. long, and 16 ft tall. Hole 3 was in the northern section of the site with no perceivable backfill. Due to the unregistered mine map, however, over 50% of the scan area in Hole 3 was blocked by an adjacent mine pillar. Furthermore, the cuttings pile accumulated almost to the

ceiling and created a narrow band of viewing space for scanning. Therefore, the scan was only able to determine cavity extent.

The data and analysis generated by Ferret resulted in a decision to proceed with a second backfill operation. In utilizing Ferret, the number of boreholes required to make this assessment was reduced by 40% according to site engineers. Instead of inference from camera imagery, quantitative data was available to declare a possible subsidence threat. Currently, arrangements are in progress to recall Ferret to the site to assist during the backfill process.

From this experience, several strengths and weaknesses in the functionality of Ferret were discovered that merit mentioning in this paper. First, the success of borehole-deployable scanning devices at acquiring cavity information is highly dependent upon the location of the boreholes. In this study, both hole 1 and hole 2 provided advantageous locations to view the mine. Hole 3, however, is an example case where poor locality (near solid objects) severely limited the amount of obtainable range information.

Another important issue is device size. In its current configuration, Ferret can access 12 in. diameter boreholes without difficulty. Holes of this size reduce the economical advantage that Ferret employs. A size reduction that permits deployment in 6 in. diameter boreholes would greatly improve the costs for drilling access holes; however, laser equipment of this form factor with the robustness, reliability, and price of the previous Ferret is difficult to fabricate.

In conclusion, the Ferret is a field-operational robotic device that provides a remote, underground presence to acquire highly accurate 3D models. In this Kansas City case study, Ferret was shown to be a valuable component in realizing, measuring, and mapping abandoned limestone mine sections and domeouts for structural analysis and remediation. Furthermore, this study suggests the flexibility of Ferret technology to other application domains such as verification of abandoned coalmines and situational analysis of subterranean hazardous waste disposals.

Acknowledgements

The land owning company and engineering firm in Kansas City who learned of our technology developments and had the insight to implement us into their plans and Workhorse Technology, LLC that funded and supported the development of Ferret and mine mapping initiatives at Carnegie Mellon University.

References

1. M. Ross and M. Roth. "All Nine Alive: The Story of the Quecreek Mine Rescue." *Pittsburgh Post-Gazette* 04 August 2002.
2. Beck, Barry F. and Herring, J.Gayle (eds): Geotechnical and environmental applications of karst geology and hydrology - *Proceedings of the 8th multidisciplinary conference on sinkholes & karst, Louisville, Kentucky, USA, 1-4 April 2001*

3. MSHA Public Safety Campaign Stresses That Mines and Minors Don't Mix. MSHA Press Release, U.S. Dept. of Labor, 18 April 2001.
4. P. Burton, A. Kreeger, B. Shefchik. "The Use of Flyash in Mine Stabilization." *TechBriefs*, 2002 vol. 2.
5. "Ground Penetrating Radar." *U.S. Environmental Protection Agency Field Analytic Technologies Encyclopedia*, <http://fate.clu-in.org/> (current May 2003).
6. G. Senechal, F. Hollender, D. Russet. "Near Surface Characterization of a Limestone Site Using Borehole and Surface Geophysics." *Geophysical Research Abstracts* Vol. 5.
7. "Microgravity Surveying," *Keele University Applied & Environmental Geophysics Research Group*, <http://www.esci.keele.ac.uk> (current May 2003).
8. Mark Products, Inc. GeoVISION™ Video Borehole Inspection System. Online documentation: www.marksproducts.com, December 2002.
9. B. Wells. "The Latest Developments in Laser Profiling, Borehole Deviation, and Laser Enhanced Videometry." *Measurement Devices Ltd.*, 1999.
10. P. Neugebauer. "Reconstruction of Real-World Objects Via Simultaneous Registration and Robust Combination of Multiple Range Images," *International Journal of Shape Modeling*, 1997. Vol. 3, pp 71-90.
11. P. Besl and N. McKay. "A Method of Registration of {3D} Shapes," *jPAMI*, Feb. 1992. Vol. 14, pp 239-256.
12. D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*, Prentice Hall, Ed. 1, 2002.

Bayesian Programming for Multi-target Tracking: An Automotive Application

Christophe Coué, Cédric Pradalier, and Christian Laugier

Inria Rhône-Alpes & Gravir CNRS

firstname.lastname@inrialpes.fr

<http://www.inrialpes.fr/sharp/>

Abstract. A prerequisite to the design of future Advanced Driver Assistance Systems for cars is a sensing system providing all the information required for high-level driving assistance tasks. In particular, target tracking is still challenging in urban traffic situations, because of the large number of rapidly maneuvering targets. The goal of this paper is to present an original way to perform target position and velocity estimation, based on the occupancy grid framework. The main interest of this method is to avoid the decision problem of classical multi-target tracking algorithms. Obtained occupancy grids are combined with danger estimation to perform an elementary task of obstacle avoidance with an electric car.

1 Introduction

Unlike regular cruise control systems, Adaptive Cruise Control (ACC) systems use a range sensor to regulate the speed of the car while ensuring collision avoidance with the vehicle in front. ACC systems were introduced on the automotive market in 1999. Since then, surveys and experimental assessments have demonstrated the interest for this kind of systems. They are the first step towards the design of future Advanced Driver Assistance Systems (ADAS) that should help the driver in increasingly complex driving tasks. The use of today's commercially available ACC systems is pretty much limited to motorways or urban expressways without crossings. The traffic situations encountered are rather simple and attention can be focused on a few, well defined detected objects (cars and trucks). Nonetheless, even in these relatively simple situations, these systems show a number of limitations: they are not very good at handling fixed obstacles and may generate false alarms; moreover, in some 'cut-in' situations, *ie* when the insertion of an other vehicle in the detection beam is too close to the vehicle, they may be taken by surprise.

A wider use of such systems requires to extend their range of operation to some more complex situations in dense traffic environments, around or inside urban areas. In such areas, traffic is characterized by lower speeds, tight curves, traffic signs, crossings and "fragile" traffic participants such as motorbikes, bicycles or pedestrians. A prerequisite to a reliable ADAS in such complex traffic situations is an estimation of dynamic characteristics of the traffic participants, such as position and velocity. This problem is basically a *Multi-Target Tracking* problem.

Our goal in this paper is to present an original method of multi-target tracking, based on *occupancy grids* framework [1,2]. This method avoid the major drawbacks

of classical approaches, in situations where the number of targets is not a crucial information. To validate the method, an application to the longitudinal control of a robotic golf car named "Cycab" is presented. This vehicle is equipped with a Sick laser range finder, allowing the system to estimate position and speed of the targets relatively to the Cycab.

To deal with uncertain or incomplete knowledge, the *Bayesian Programming* (BP) [3] method is applied in the framework of Bayesian theory as depicted in [4].

The paper is organized as follows: §2 presents the Bayesian Programming concept. The next section overviews multi-target tracking algorithms. Then the §4 presents our estimation algorithm, and the §5 the longitudinal control of the cycab.

2 Bayesian Programming

Any model of a real phenomenon is inherently incomplete. There are always some hidden variables, not taken into account in the model that influence the phenomenon. Furthermore, perception and control are inherently uncertain, *ie* including perception and control errors. Rational reasoning with incomplete and uncertain information is quite a challenge. Bayesian Programming addresses this challenge relying upon a well established formal theory: the probability theory [4].

In this framework, a Bayesian Program is made up of two parts: a *description* and a *question*.

The description can be viewed as a knowledge base containing the a priori information available on the problem at hand. It is essentially a joint probability distribution.

Given a distribution, it is possible to ask *questions*. Questions are obtained first by partitioning the set of variables into three sets: (1) *Searched*: the searched variables, (2) *Known*: the known variables, and (3) *Free*: the free variables. A question is then defined as the distribution $P(\textit{Searched} \mid \textit{Known})$.

Given the description, it is always possible to answer a question, *ie* to compute the probability distribution $P(\textit{Searched} \mid \textit{Known})$. An inference engine has been implemented to automate Bayesian inference. As general Bayesian inference problem has been shown to be NP-Hard [5], much work is dedicated to applicability and complexity reduction of the inference.

3 Multiple Target Tracking

As mentionned earlier, the problem addressed in this paper is basically a *Multi-Target Tracking* problem. The objective is to collect *observations*, *ie* data from the sensor, on one or more potential *objects* in the environment of the vehicle, and then to estimate the objects' position and velocity. Classical approach is to track the different objects independently, by maintaining a list of *tracks*, *ie* a list of currently known objects.

The main difficulty of multi-target tracking is known as the *Data Association* problem. It includes observation-to-track association and track maintenance problems. The goal of observation-to-track association is to decide whether a new sensor observation corresponds to an existing track or not. Then the goal of track maintenance is to decide the confirmation or the deletion of each existing track, and the initiation of new tracks. Numerous methods exist to perform this data association problem [6–8]. A complete review of the tracking methods with one or more sensors can be found in [9].

Urban traffic scenarios are still a challenge in multi-target tracking area. The data association problem is intractable in situations involving numerous appearance, disappearance and occlusions of a large number of rapidly manoeuvring targets.

To avoid the data association problem, we do not maintain a list of tracks. We prefer to estimate the occupied and free space of the environment of our vehicle. We were inspired by *occupancy grids* [1,2], which is extensively used for mapping and localization [10]. According to us, this kind of environment representation provides enough information to perform vital behaviors. This will be illustrated in the § 5.

The next section presents our method to estimate a 4-dimensional occupancy grid of the Cycab environment.

4 Estimation of the Occupancy Grid

4.1 Bayesian Program

The main idea of occupancy grids is to tessellate the environment of a robot into cells, and then to estimate from sensor data the probability that each cell is full or empty. To avoid a combinatorial explosion of grid configuration, the cell states are estimated as *independent* random variables.

Occupancy grids framework was extensively used for mapping and localization. Of course, for an automotive application, it is impossible and useless to model the whole environment of the vehicle with a grid. Thus we will model only the near-front environment of our vehicle. As we want to estimate the relative position and the relative velocity of objects, each cell of our grid correspond to a position and a speed relatively to the Cycab. Thus our grid is 4-D.

Fig 1 presents the Bayesian Program for the estimation of the occupancy probability of a cell. To simplify notations, a particular cell of the grid is denoted by a single variable X , despite the grid is 4-D. The number of sensor observations at time k is named N^k . One sensor data at time k is denoted by the variable Z_i^k , $i = 1 \dots N_k$. The set of all sensor observation at time k is noted Z^k . The set of all sensor observations until time k is referred by the notation $\mathcal{Z}_{1:k}$.

A variable called the “matching” variable and noted M^k is added. Its goal is to specify which observation of the sensor is currently used to estimate the state of the cell. The result of the inference for the estimation is given by:

$$P(E_X^k | X^k \mathcal{Z}_{1:k}) = \frac{1}{\alpha} \sum_{M^k=1}^{N^k} \left(P(E_X^k | X^k \mathcal{Z}_{1:k-1}) \prod_{s=1}^{N_k} P(Z_s^k | [M^k = s] E_X^k X^k) \right),$$

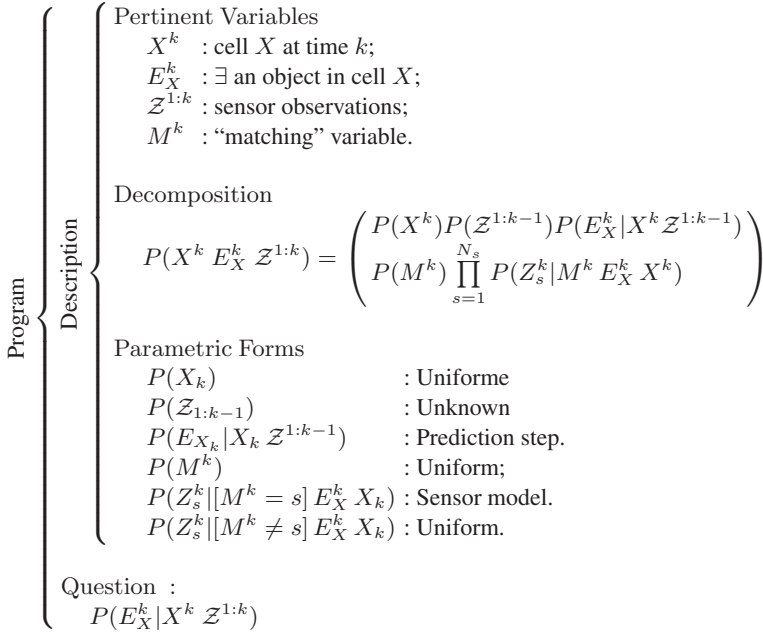


Fig. 1. Estimation Step at time k

where α is a normalization constant.

During the inference, the sum on this variable allows to take into account all sensor observations to update the state of one cell.

One has to remark here that the estimation step is performed without explicit association between tracks and observations.

To improve the estimation of the grid, we add a *prediction step*. The goal of the prediction is to provide an *a priori* knowledge for the estimation. It is based on a dynamical model of the object, which include the vehicle movements, denoted by the variable U^{k-1} . Such prediction step is classical in target tracking, in sequential estimation techniques such as the Kalman filter [11]. The corresponding Bayesian program is presented by fig 2.

The result of the inference for the prediction is given by:

$$P(E_X^k | X^k \mathcal{Z}^{1:k-1} U^{k-1}) = \frac{1}{\alpha} \int_{\substack{X^{k-1} \\ E_{X^{k-1}}}} \left(\begin{array}{c} P(X^{k-1}) P(E_X^{k-1} | X^{k-1} \mathcal{Z}^{1:k-1}) \\ P(X^k | X^{k-1} U^{k-1}) \\ P(E_X^k | E_X^{k-1} X^k X^{k-1}) \end{array} \right), (1)$$

where α is a normalization constant.

In general, this expression cannot be determined analytically, and cannot even be computed in real time. Thus an approximate solution of the integral has to be computed.

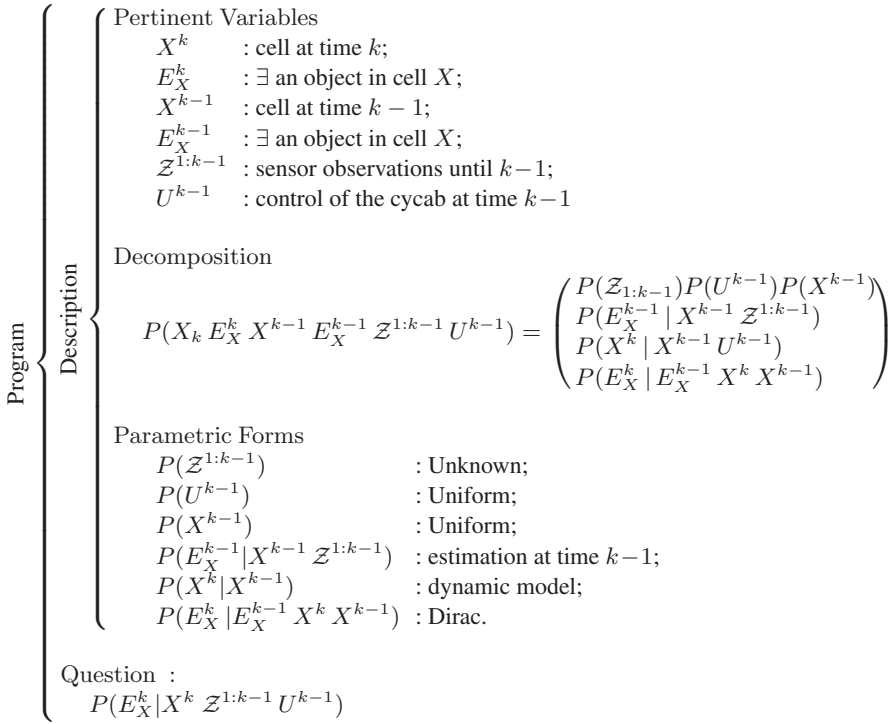


Fig. 2. Prediction Step at time k

Our approximation algorithm is based on the basic idea that only few points can be used to approximate the integral. Thus, for each cell of the grid at time $k-1$, we compute the probability distribution $P(X^k | X^{k-1} U^{k-1})$. A cell x^k is drawn according to this probability distribution. Then the cell X^{k-1} is used to update only the predicted state of the cell x^k . The complexity of this algorithm increases linearly with the number of cells in our grid, and ensures that the most informative points are used to compute the integral appearing in (1).

Thus the estimation of the occupancy grid at time k is done in two steps. The prediction step uses the estimation step at time $k-1$ and a dynamical model to compute an *a priori* estimate of the grid. Then the estimation step uses this prediction and the sensor observations at time k to compute the grid.

4.2 Experimental Results

To test the estimation of occupancy grids presented in the previous section, both a simulator and the real Cycab vehicle were used.

Fig 3 shows first results of estimation and prediction steps, for a static scene. The upper left scheme depicts the situation : two static objects are present in front of the

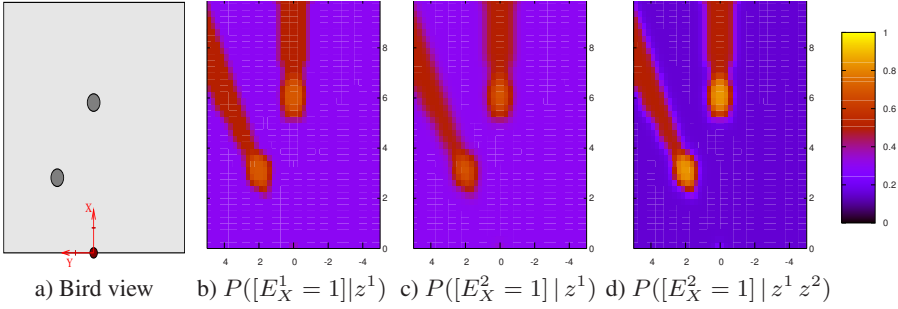


Fig. 3. First example of grid estimation, for a static scene.

cycab. These two objects are fixed. The cycab is static too. Thus only 2-dimensional grids are depicted, corresponding to object's position at a null speed. Fig 3b represents the occupancy grid, knowing only the first sensor observations. The color corresponds to the probability that a cell is occupied. In this case, the two objects are detected by the sensor. Consequently, two areas with high occupancy probabilities are visible (orange areas). These probability values depends on probability of detection, probability of false alarm, and on sensor precision. All these characteristics of the sensor are taken into account in the sensor model. The cells hidden by a sensor observation have not been observed. Thus we can not conclude about their occupancy. That explains the two areas of probability values near to 0.5 (red areas). Finally, for cells located far from any sensor observation, the occupancy probability is low (purple areas).

Fig 3c represents the occupancy grid, after the prediction step. Because all the scene is static, the result of the prediction step is quite similar to the result of the first estimation step.

Fig 3d represents the occupancy grid after the second sensor observations. Of course, this figure is quite similar to the fig 3b, but two remarks have to be noted:

- for cells corresponding to an objects, the occupancy probability is higher than in the first estimation. This is due to the *a priori* knowledge, which was uniform for the first estimation, and given by the prediction step for the second estimation;
- for cells corresponding to free areas, the occupancy probability is lower than in the first estimation. This is also due to the prediction step.

Fig 4 shows an extract of a short sequence of successive prediction and estimation results. Its goal is to demonstrate the robustness of our approach to objects occlusions, without any special logic. The first row describes the situation : the cycab is immobile, a static object is located 5 meters in front of it. A second object is moving from right to left. In the situation depicted by the figs 4b.1 and 4c.1, the moving object is hidden by the static one, and thus is not detected by the Sick laser range finder.

Second and third rows present respectively results of the prediction step and of the estimation step. We choose to represent only the cells of the grids corresponding to relative speed equals to $\dot{x} = 0.0$, $\dot{y} = 1.0 \text{ m/s}^{-1}$, which is close to the speed of

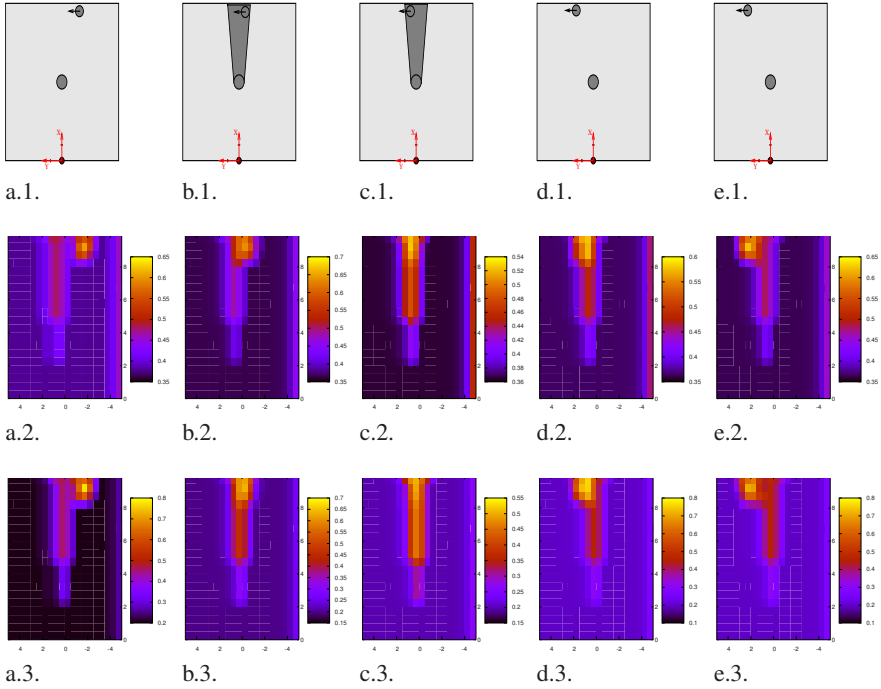


Fig. 4. A short sequence of a dynamic scene. The first row describes the situation : a moving object is temporary hidden by a static object. The second row shows the predicted occupancy grids, and the third row shows the result of the estimation step. The grids are 2-dimensional, and shows the probability $P([E_X^k = 1] | x y [\dot{x} = 0][\dot{y} = 1.0])$.

the moving object. The color represents the occupancy probability of the cells. Be careful that the color code is different for each sub-figure.

An area of high occupancy probability is well defined in figs 4a.2 and 4a.3. This area corresponds to the moving object. We remark an area of occupancy probability values equals to 0.5, which corresponds to the cells hidden by the static object. The fig 4b.2 presents the result of the prediction step, based on the grid presented in fig 4a.3, and on a dynamic model. This prediction shows that an object should be located in the area hidden by the static object. Consequently, even if this object is not detected by the laser, an area of high occupancy probability is found in the fig 4b.3. In the same way, the fig 4c.2 predicts that a moving object should be located behind the static object. Thus we still found in fig 4c.3 an area of occupancy probability values greater than 0.5. Of course, the certainty in object presence, *ie* the values of the occupancy probability in the grid, decreases when the object is not observed by the sensor.

In figs 4d.3 and 4e.3, the moving object is no longer hidden by the static object. Thus it is detected by the laser, and the occupancy probability values increase.

Occupancy grids such ones presented in figs 3 and 4 don't provide precise informations about the vehicle environment. In particular, the number of objects is not estimated. Thus one can think that they are useless in the automotive context.

The next section shows that occupancy grids provide enough information about the environment to perform basic and vital behaviors. The goal is to control the cycab in order to avoid "hazardous" obstacles of the environment.

5 Longitudinal Control of the Cycab

As mentioned in [2], the cell state can be used to encode a number of properties of the robot environment. Properties of interest for robot programming could include occupancy, observability, reachability, etc. In the previous section, it was used to encode the occupancy of the cell. In this section, we show how it could be used to encode the *danger* of the cell. By this way, the cycab is longitudinally controlled by combining the occupancy and the danger of all cells.

5.1 Estimation of Danger

For each cell of the grid, the probability that this cell is hazardous is estimated. This estimation is done without considering the occupancy probability of the cell. Thus we estimate the probability distribution $P(D_X^k | X^k)$, for each cell X of the cycab environment. D_X^k is a boolean variable that indicates whether the cell X is hazardous or not.

As a cell X of our grid represents a position and a velocity, the TCPA (Time to the Closest Point of Approach) and the DCPA (Distance to the Closest Point of Approach) can be estimated for each cell. Thanks to TCPA and DCPA, the estimation of the danger is more intuitive than if we had considered directly the relative speed encoded in the grid : the lower the DCPA and the shorter the TCPA, the more hazardous the cell.

Fig 5 shows the cells for which danger probability is greater than 0.7. Each cell is modeled with an arrow: the beginning of the arrow indicates the position, the length and the direction indicates the speed. First, we can see that any cell located close to the cycab is considered as hazardous, whatever the speed is. For other locations, the more hazardous cells are those which speed is in the direction of the cycab. As we consider relative speed in the danger grid, this grid does not depend of the actual cycab velocity.

5.2 Control of the Cycab

Our goal here is to control the longitudinal speed of the cycab, in order to avoid moving objects. The behavior we want the cycab to adopt is very simplistic : brake or accelerate whether it feels itself in danger or not.

To program this behavior, we consider simultaneously for each cell X of the environment its danger probability (given by the distribution $P(D_X^k | X^k)$ explained in

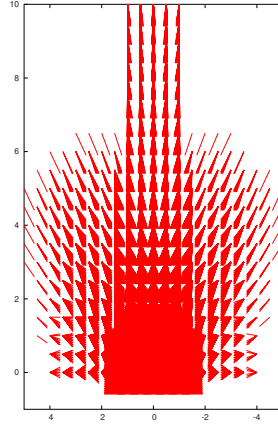


Fig. 5. Cells of high danger probabilities. For each position, arrows model the speed.

the § 5.1) and its occupancy probability (given by the distribution $P(E_X^k | \mathcal{Z}^{1:k} X^k)$ explained in the § 4). We look for the most hazardous cell that is considered as occupied, that is:

$$\max_{X^k} \{P(D_X^k | X^k), \text{ with } P(E_X^k | X^k) > 0.5\}.$$

Then the longitudinal acceleration of the cycab is decided according to this level of danger and to its actual velocity. The four possible commands are : emergency brake, brake, accelerate or keep the same velocity.



Fig. 6. Example of the cycab control.

Fig 6 illustrates this basic control of the cycab. In this example, a pedestrian appears suddenly. The cycab brakes to avoid him. This basic control allows the cycab to adopt secure behaviors, such as keeping a safety distance from a car preceding itself. Stop and go is also ensured with this control. Videos illustrating these behaviors should be available at <http://www.inrialpes.fr/sharp/people/coue/>.

What it is to be noted here is that no decision is taken before the choice of the command applied to the cycab. In particular, we do not know the exact number of object located in the cycab environment. Furthermore, exact positions and velocity of these objects are not estimated.

6 Conclusion

This paper addressed the problem of 4-D occupancy grid estimation in an automotive context. According to us, this grid can be an alternative to complex multi-target tracking algorithms for applications which does not require information such as the number of objects. To improve the estimation, a prediction step has been added. Thanks to this prediction, the estimation of the grid is robust to temporary occlusions between moving objects. To validate the approach, an application involving the Cycab vehicle has been shown. The Cycab is longitudinally controlled in order to avoid obstacles. This basic behavior is obtained by combining the occupancy probability and the danger probability of each cell of the grid.

Future developments will include: a) improvements of the approximation algorithm for the prediction step. These improvements should allow to estimate a bigger grid, which is required to control a car in urban areas. b) fusion of the occupancy grid with higher-level information, such as GPS maps, to better estimate the danger of the situation.

Acknowledgement

This work was partially supported by the European project IST-1999-12224 “*Sensing of Car Environment at Low Speed Driving*” (<http://www.carsense.org>).

References

1. H. P. Moravec, “Sensor Fusion in Certainty Grids for Mobile Robots,” *A.I. Magazine*, vol. 9, pp. 61–74, 1988.
2. A. Elfes, “Using Occupancy Grids for Mobile Robot Perception and Navigation,” *IEEE Computer, Special Issue on Autonomous Intelligent Machines* pp. 46–57, 1989.
3. O. Lebeltel et al., “Bayesian Robot Programming”, *Autonomous Robots* January, 2004.
4. E. T. Jaynes, *Probability Theory: the Logic of Science*, Cambridge University Press, UK, 2003.
5. G. Cooper, “The Computational Complexity of Probabilistic Inference Using Bayesian Belief Network,” *Artificial Intelligence*, vol. 42, 1990.
6. Y. Bar-Shalom and X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Artech House, 1995.
7. H. Gauvrit et al, “A Formulation of Multitarget Tracking as an Incomplete Data Problem,” *IEEE Trans. on Aerospace and Electronic Systems*, vol. 33, num. 4, 1997.
8. R.L. Streit and T.E. Luginbuhl, “Probabilistic Multi-Hypothesis Tracking,” *Naval Undersea Warfare Center Division Newport*, Technical Report, num. 10,428, 1995.
9. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 2000.
10. “Robotic Mapping : a Survey,” *Exploring Artificial Intelligence in the New Millenium*, Morgan Kaufman, 2002.
11. “A New Approach to Linear Filtering and Predictions Problems,” *Journal of Basic Engineering*, March, 1960.

Optimal Search for a Lost Target in a Bayesian World

Frédéric Bourgault¹, Tomonari Furukawa², and Hugh F. Durrant-Whyte¹

¹ ARC Centre of Excellence for Autonomous Systems (CAS)
Australian Centre for Field Robotics
The University of Sydney, Sydney, NSW 2006, Australia
{f.bourgault, hugh}@acfr.usyd.edu.au
<http://www.acfr.usyd.edu.au>

² ARC Centre of Excellence for Autonomous Systems (CAS)
School of Mechanical and Manuf. Engineering
The University of New South Wales, Sydney, NSW 2052, Australia
t.furukawa@unsw.edu.au

Abstract. This paper presents a Bayesian approach to the problem of searching for a single lost target by a single autonomous sensor platform. The target may be static or mobile but not evading. Two candidate utility functions for the control solution are highlighted, namely the Mean Time to Detection, and the Cumulative Probability of Detection. The framework is implemented for an airborne vehicle looking for both a stationary and a drifting target at sea. Simulation results for different control solutions are investigated and compared to demonstrate the effectiveness of the method.

1 Introduction

“We are sinking fast. Position ten miles south of San Remo...”¹

When rescue authorities receive a distress signal time becomes critical. The probability of survival decreases rapidly in a matter of hours when lost at sea. The prime focus of a rescue mission is to search for and find the castaways in the smallest possible amount of time. The search, based on some coarse estimate of the target location, must often be performed in low visibility conditions and despite strong winds and high seas making the location estimate even more uncertain as time goes by. Keeping these time and physical constraints in mind, what should be the optimal search strategy?

This paper presents a Bayesian framework that integrates and predicts the effects of the observations and the process model on the target distribution. The control solution formulation is then described for a single autonomous vehicle searching for a single non evading, but possibly mobile target.

The paper is organized as follows. First, the Bayesian filtering algorithm that accurately maintains and updates the target state probability density function (PDF) is described in Sect. 2. Section 3 describes the searching problem, highlights two utility candidate functions and formulates the control optimization problem. Section 4 implements the framework for an airborne search vehicle and investigates the

¹ Mayday from yacht *Winston Churchill*, April 1959 [10]

control solutions and the effectiveness of the approach for both a stationary, and a drifting target. Finally, conclusions and ongoing research directions are highlighted.

2 Bayesian Analysis

This section presents the mathematical formulation of the Bayesian analysis from which the control solutions presented in this paper are derived. The Bayesian approach is particularly suitable for combining, in a rational manner, non-linear motion models and heterogeneous non-Gaussian sensor measurements with other sources of quantitative and qualitative information [8][1].

In Bayesian analysis any quantity that is not known is modelled as a random variable. The state of knowledge about such a random variable is expressed in the form of a probability density function (PDF). Any new information in the form of a probabilistic measurement or observation is combined with prior information using the Baye's theorem in order to update the state of knowledge and form the new a posteriori PDF. That PDF forms the quantitative basis on which all control decisions or inferences are made.

In the searching problem, the unknown variable is the target state vector $\mathbf{x}^t \in \mathcal{X}^t$ which in general describes its location but could also include its attitude, velocity, etc. The analysis starts by determining the a priori PDF of \mathbf{x}^t , $p(\mathbf{x}_0^t | \mathbf{z}_0) \equiv p(\mathbf{x}_0^t)$, which combines all available information including past experience. For example, this prior PDF could be in the form of a Gaussian distribution representing the prior coarse estimate of the parameter of interest. If nothing is known about the parameter, a least informative approach is to represent this knowledge by a uniform PDF. Then, once the prior distribution has been established, the PDF of the target state at time step k , $p(\mathbf{x}_k^t | \mathbf{z}_{1:k})$, can be constructed recursively, provided the sequence $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$ of all the observations made by the sensor(s) on board the search vehicle, \mathbf{z}_k being the observation (or the set of observations, if multiple sensors) made a time step k . This recursive estimation is done in two stages: prediction and update.

2.1 Prediction

A prediction stage is necessary in Bayesian analysis when the PDF of the state to be evaluated is evolving with time i.e. the target is in motion or the uncertainty about its location is increasing. Suppose we are at time step k and the latest PDF update, $p(\mathbf{x}_{k-1}^t | \mathbf{z}_{1:k-1})$ (from the the previous time step) is available. Then the predicted PDF of the target state at time step k is obtained from the following Chapman-Kolmogorov equation

$$p(\mathbf{x}_k^t | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k^t | \mathbf{x}_{k-1}^t) p(\mathbf{x}_{k-1}^t | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}^t \quad (1)$$

where $p(\mathbf{x}_k^t | \mathbf{x}_{k-1}^t)$ is a probabilistic Markov motion model. If the motion model is invariant over the target states, then the above integral is simply a convolution.

Practically, this convolution is performed numerically by a discretization of the two PDF's on a grid, followed by the multiplication of their Fast Fourier Transforms (FFT)'s, followed by an inverse FFT of the produce to retrieve the result.

2.2 Update

At time step k a new observation \mathbf{z}_k becomes available and the update is performed using Bayes rule where all the observations are assumed to be independent. The update is performed simply by multiplying the prior PDF (posterior from the prediction stage) by the new conditional observation likelihood noted $p(\mathbf{z}_k|\mathbf{x}_k^t)$ as in the following

$$p(\mathbf{x}_k^t|\mathbf{z}_{1:k}) = K p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1}) \cdot p(\mathbf{z}_k|\mathbf{x}_k^t) \quad (2)$$

where the normalization factor K is given by

$$K = 1 / \int [p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1})p(\mathbf{z}_k|\mathbf{x}_k^t)] d\mathbf{x}_k^t \quad (3)$$

Practically, the multiplication of (2) is performed numerically by multiplying together the corresponding elements of a grid.

3 The Searching Problem

This section describes the equations for computing the probability of detection of a lost object referred to as the target. For further details on the searching problem the reader is referred to [7] and [6].

If the target detection likelihood (observation model) at time step k is given by $p(\mathbf{z}_k|\mathbf{x}_k^t)$ where $\mathbf{z}_k = D_k$ for which D_k represents a “detection” event at t_k , then the likelihood of “no detection”, given a target state \mathbf{x}_k^t is given by its complement

$$p(\overline{D}_k|\mathbf{x}_k^t) = 1 - p(D_k|\mathbf{x}_k^t) \quad (4)$$

At time step k , the conditional probability that the target does not get detected during a sensor observation, $p(\overline{D}_k|\mathbf{z}_{1:k-1}) = q_k$, depends on two things: the ‘no detection’ likelihood (4), and the latest target PDF $p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1})$ (from the prediction stage (1)). In fact q_k corresponds exactly to the volume under the surface formed when multiplying the two together (element-by-element for each given target state \mathbf{x}_k^t) as in the following

$$p(\overline{D}_k|\mathbf{z}_{1:k-1}) = \int p(\overline{D}_k|\mathbf{x}_k^t)p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1})d\mathbf{x}_k^t = q_k \quad (5)$$

Hence q_k is given by the reduced volume (< 1) under the target state PDF after having been carved out by the ‘no detection’ likelihood in the update stage (2), but before applying the normalization factor to it. Notice that this volume is exactly the inverse of the normalization factor K (see (3) for a ‘no detection’ event ($\mathbf{z}_k = \overline{D}_k$)), so $q_k = 1/K$ and is always smaller than 1. The joint probability of failing to detect

the target in all of the steps from 1 to k , noted $Q_k = p(\overline{D}_{1:k})$, is obtained from the product of all the q_k 's as follows

$$Q_k = \prod_{i=1}^k p(\overline{D}_i | \overline{D}_{1:i-1}) = \prod_{i=1}^k q_i = Q_{k-1} q_k \quad (6)$$

where $\overline{D}_{1:i-1}$ corresponds to the set of observations $\mathbf{z}_{1:i-1}$ where all observations are equal to \overline{D} . Therefore, in k steps, the probability that the target has been detected, denoted P_k , is given by

$$P_k = 1 - Q_k \quad (7)$$

It is also possible to compute the probability that the target gets detected for the first time on time step k , denoted p_k , as follows

$$p_k = \prod_{i=1}^{k-1} p(\overline{D}_i | \overline{D}_{1:i-1}) [1 - p(\overline{D}_k | \overline{D}_{1:k-1})] = \prod_{i=1}^{k-1} q_i [1 - q_k] = Q_{k-1} [1 - q_k] \quad (8)$$

which in turn by summing over k provides a sequential method for evaluating P_k as

$$P_k = \sum_{i=1}^k p_i = P_{k-1} + p_k \quad (9)$$

For this reason P_k will be referred to as the ‘cumulative’ probability of detection at time k to distinguish it from the conditional probability of detection at time k which is equal to $1 - q_k$. Notice that as k goes to infinity, the cumulative probability of detection increases towards one. With k increasing, the added probability of detection p_k gets smaller and smaller as the conditional probability of detection $(1 - q_k)$ gets discounted by a continuously decreasing Q_{k-1} .

The mean time to detection (MTTD) is the expectation of the number of steps required to detect the target

$$E[k] = \sum_{k=1}^{\infty} k p_k = \text{MTTD} \quad (10)$$

The goal of the searching strategy could either be to maximize the chances of finding the target given a restricted amount of time by maximizing P_k over the time horizon, or to minimize the expected time to find the target by minimizing the MTTD. The difficulty though in evaluating the MTTD lies in the fact that one must in theory evaluate p_k for all k 's up to infinity.

3.1 Optimal Trajectory

Optimality is always defined in relation to an objective, or utility function [9]. For the searching problem there are two suitable candidates to evaluate a trajectory utility, namely the cumulative probability of detection P_k (9), and the MTTD (10).

For an action sequence $\mathbf{u} = \{u_1, \dots, u_{N_k}\}$ over a finite time horizon of length $T = N_k dt$, we thus have as an objective function either

$$J(\mathbf{u}, N_k) = \sum_{i=k}^{k+N_k} p_i = P_{k+N_k} - P_k \quad \text{or} \quad J(\mathbf{u}, N_k) = - \sum_{i=k}^{k+N_k} i p_i \quad (11)$$

The optimal control strategy \mathbf{u}^* is the sequence that maximizes that utility subject to the vehicle limitations $\mathbf{u}_{LB} \leq \mathbf{u} \leq \mathbf{u}_{UB}$.

$$\mathbf{u}^* = \{u_1^*, \dots, u_{N_k}^*\} = \arg \max_{\mathbf{u}} J(\mathbf{u}, N_k) \quad (12)$$

For the searching problem, because early actions strongly influence the utility of subsequent actions, the longer the time horizon, the better the computed trajectory. However, the computational cost follows the ‘‘curse of dimensionality’’ and with increasing lookahead depth the solution becomes intractable. In practice only solutions for very restricted lookahead length are possible. One way to increase the lookahead without increasing the cost of the solution too much is to use piecewise constant control sequences (see [5] and [2]) where each control parameters is maintained over a specified number of time steps. Such control solutions are said to be ‘quasi-optimal’ as they compromise the global optimality of the control solution for a lower computation cost, but nevertheless, depending on the problem at hand, often provide better trajectories than the ones computed with the same number of control parameters but with shorter time horizons.

3.2 One-step Lookahead

Planning with a time horizon of only one step is an interesting special case of the searching problem as both objective functions reduce to $J(\mathbf{u}, 1) = p_k$.

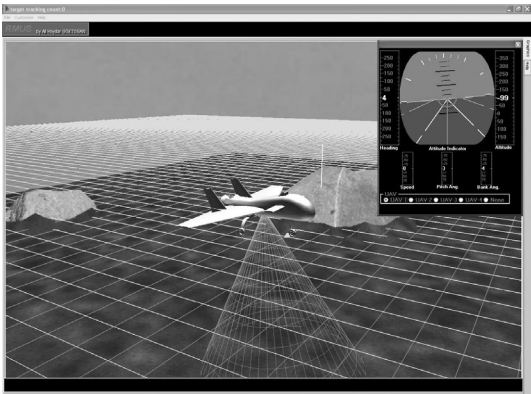
Also, because $p_k = Q_{k-1}(1 - q_k)$ (8), maximizing p_k at time step t_k is equivalent to maximizing $(1 - q_k) = p(D_k | \mathbf{z}_{1:k-1})$, the conditional probability of detecting the target (which corresponds to the volume under the surface resulting from the multiplication of the ‘detection’ likelihood with the predicted target PDF), or conversely minimizing $q_k = p(\overline{D}_k | \mathbf{z}_{1:k-1})$ (5), the conditional probability of ‘not detecting’ the target (volume under the surface resulting from multiplying the ‘no detection’ likelihood with the predicted target PDF). As will be seen in the results section 4.4, this greedy from of searching strategy provides very sensible control solutions at very low computational costs.

4 Application

The goal of the work presented in this paper is to ultimately implement and demonstrate the framework for an autonomous search on one of the ACFR’s unmanned air vehicle (UAV) as shown in Fig. 1a. ACFR has also developed a high fidelity simulator (Fig. 1b) of the UAV’s hardware, complete with different sensor models,



(a)



(b)

Fig. 1. Application: (a) one of the Brumby Mark-III uav’s been developed at ACFR as part of the ANSER project. This flight vehicle has a payload capacity of up to 13.5 kg and operational speed of 50 to 100 knots; (b) display of the high fidelity simulator

on which the flight software can be tested before being implemented on board the platform almost without any modifications [3].

The rest of this section describes the implementation of the Bayesian searching framework for a single airborne vehicle searching for a single non-evading lost target that could either be stationary or mobile. However, the method is readily applicable to searching problems of all kinds, be it ground, underwater or airborne search for bushfires, lost hikers, enemy troops in the battlefield, or prospection for ore and oil, or even to search for water or evidence of life on another planet.

4.1 Problem Description

The problem chosen for the illustration of the framework involves the search by an airborne vehicle for a life-raft lost at sea. The search platform is equipped with a GPS receiver, i.e. assuming perfect localization, and a searching sensor (e.g radar, human eye, infrared or CCD camera) that can be modelled by a likelihood function (over range and bearing) hence relating the control actions to the probability of finding the target. There is one observation (full scan) made once every second. The sensor is assumed to have perfect discrimination i.e. no false target detection. However, it may fail to call a detection when the target is present i.e. miss contact.

The omnibearing sensor's maximum range (400m) is much smaller than the size of the searching area (2km x 2km). Drift current and winds (of up to 30 knots) affect the target distribution over time in a probabilistic way through the process model. The target PDF is of general form (i.e. non-Gaussian) and is evaluated and maintained on a discrete grid. As the length of the search is limited by the vehicle fuel autonomy, the utility function selected is given by (11) (left) and consists of maximizing the cumulative probability of finding the target in a fixed amount of time.

4.2 Motion Prediction

Vehicle Model The vehicle pose prediction model used for the planning purposes is the following discrete time non-linear constant velocity model

$$x_{k+1}^s = x_k^s + \frac{2V}{u_k} \sin\left(\frac{1}{2}u_k dt\right) \cos\left(\theta_k^s + \frac{1}{2}u_k dt\right) \quad (13)$$

$$y_{k+1}^s = y_k^s + \frac{2V}{u_k} \sin\left(\frac{1}{2}u_k dt\right) \sin\left(\theta_k^s + \frac{1}{2}u_k dt\right) \quad (14)$$

$$\theta_{k+1}^s = \theta_k^s + u_k dt \quad (15)$$

where the turn rate control command u_k is maintained over the time interval dt . For $u_k dt \ll 1$, i.e. turn rate close to zero, (13) and (14) reduce to

$$x_{k+1}^s = x_k^s + V dt \cos(\theta_k^s) \quad (16)$$

$$y_{k+1}^s = y_k^s + V dt \sin(\theta_k^s) \quad (17)$$

The maximum turn rate amplitude ($u_{max} = \pm 1.1607$ rad/s) corresponds to a 6g acceleration, the UAV's manoeuvre limit at $V = 50$ m/s (100 knots).

Process Model The model of the target state evolution noted $p(\mathbf{x}_k^t | \mathbf{x}_{k-1}^t)$, also called the target process, or motion model maps the probability of transition from a given previous state to \mathbf{x}_k^t , the target state at time t_k . It is defined by the target's equations of motion and the known statistics of the wind and the drift currents orientations and speeds. In this example, the life-raft is assumed to be drifting in the same direction and at a velocity proportional to the wind velocity. It was found that a joint distribution combining a Gaussian distribution for the wind direction with mean μ_θ and variance σ_θ^2 , and a Beta distribution for the velocity amplitude v where $v \in [0, v_{max}]$ as in the following expression

$$p(v, \theta) = \frac{c}{v_{max}} \left(\frac{v}{v_{max}}\right)^{a-1} \left(1 - \frac{v}{v_{max}}\right)^{b-1} \frac{1}{\sqrt{2\pi\sigma_\theta\mu_v}} e^{-\frac{(\theta-\mu_\theta)^2}{2\sigma_\theta^2}} \quad (18)$$

where the mean velocity $\mu_v = a/v_{max}(a+b)$, and a, b, c are the Beta distribution parameters, with $c = \frac{(a+b+1)!}{(a-1)!(b-1)!}$, seems to agree well in many cases with real wind data. The nice characteristics of a Beta distribution, over a Gaussian distribution for example, is that the distribution is defined only on a limited interval which is physically more realistic, and the function can also be skewed to various degrees by

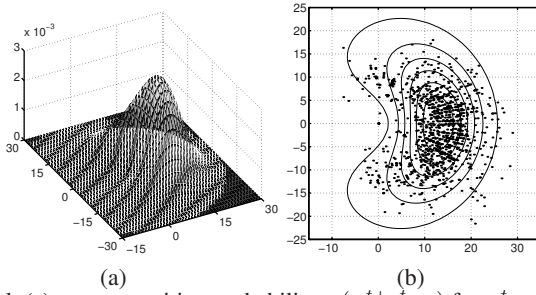


Fig. 2. Motion model: (a) target transition probability, $p(\mathbf{x}_k^t | \mathbf{x}_{k-1}^t)$ for $\mathbf{x}_{k-1}^t = [0, 0]$, and (b) its corresponding contour plot with actual wind data

adjusting the parameters a and b to match the actual data. Figure 2a shows a 3D plot of the target transition probability where $a = 4$, $b = 5$, $\sigma_\theta = \frac{\pi}{4}$ and $v_{max} = 30$ m/s. Figure 2b shows the contour plot of the function in good agreement with real wind data². For the problem described in this paper the same parameters were used except that the maximum wind velocity was set to 60 m/s giving a mean velocity of about 20 m/s (10 knots). Notice though that applying the convolution of the target prior PDF with the motion model multiple times is the same mathematically as convolving the motion model with itself multiple times and then convolving the results with the prior target PDF. The convolutions of the motion model with itself renders it more and more Gaussian like, even if the function was really far from being a Gaussian in the beginning. Therefore, for a very long searching plan, or for the case where observations only come very sporadically, a Gaussian approximation to the motion model is satisfactory.

4.3 Observation Model

The observation or sensor model is a probabilistic function representing the likelihood of the target being detected, or not ($\mathbf{z}_k = D$ or \overline{D}), conditioned on the sensor location and the state of the world.

It is not a trivial task to accurately model the sensor as many factors affect its performance: the distance to the target, the target footprint and reflectance, the transmission attenuation, and other environmental factors such as temperature, clutter and obstructions, etc.

For the purpose of this paper an active sensor model such as a downward looking millimeter wave radar was selected. It is assumed that the life raft has a radar reflector mounted on its canopy. For such a sensor, the approximate signal power, S' , received at the antennae after illumination of a target located at a distance d can be described by the following formula:

$$S' = C \frac{S A_{ant} A_t \rho}{16\pi^2 d^4} e^{-2\alpha d} \quad (19)$$

² Wind data measured at the MIT sailing pavillon on the Charles River, Cambridge, MA. Thanks to Eric Wile. <http://cbiwind.org>

where S is the emitted power, A_t and A_{ant} are the target and antennae footprints respectively, ρ is the target backscattering coefficient and α is the transmission attenuation factor which is greatly affected by the size, and density of the particles (e.g. rain) in the atmosphere. The constant C accounts for other environmental factors (e.g. background noise, temperature, etc) and could be a function of d .

If the probability of target detection is a function of the received power and the signal-to-noise ratio, then the following expression should hold true

$$\frac{P}{P_{std}} = \frac{S'}{S'_{std}} \Rightarrow P = P_{std} \frac{S'}{S'_{std}} \quad (20)$$

where by design, the reference, or ‘standard’ detection likelihood, P_{std} has a value of one (or less) for a given amount of received signal power S'_{std} evaluated at $\{d_{std}, \alpha_{std}\}$. Hence, by plugging (19) into the right side of (20), and after reduction, a closed form expression for the detection likelihood is obtained:

$$P = P_{std} \frac{d_{std}^4}{d^4} e^{-2(\alpha d - \alpha_{std} d_{std})} = p(\mathbf{z}_k = D_k | \mathbf{x}_k^t) \quad (21)$$

where the distance parameter $d (= \sqrt{h^2 + r^2})$ is a function of the vehicle altitude h and the ‘ground’ range r ($r^2 = (x^t - x_k^s)^2 + (y^t - y_k^s)^2$) to the target. In this paper the following parameter values were used: $P_{std} = 0.8$, $d_{std} = 250$, $h = 250$, and $\alpha = \alpha_{std} = 1/250$. Figure 3 illustrates the corresponding detection likelihood and its complement for a case where the sensor is located above $x = y = 0$. Another

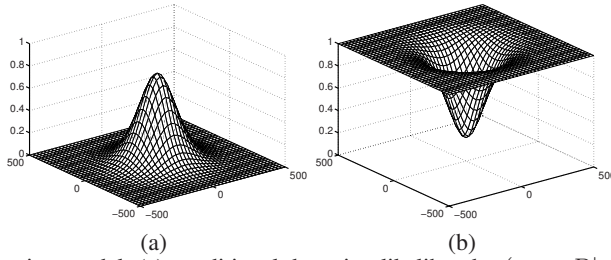


Fig. 3. Observation model: (a) conditional detection likelihood, $p(\mathbf{z}_k = D | \mathbf{x}_k^t)$ for $\mathbf{x}_k^s = [0, 0]$; (b) conditional detection complement likelihood (likelihood of ‘miss’), $p(\mathbf{z}_k = \overline{D} | \mathbf{x}_k^t) = 1 - p(\mathbf{z}_k = D | \mathbf{x}_k^t)$

very important parameter, not considered, that would contribute to a decrease in the detection likelihood with the ‘ground’ distance would be the height and wavelength of the seas. In this paper it is assumed that the radar reflector is always above the wave crests ensuring a direct line of sight to the emitting antennae.

Notice that in general, the detection likelihood of (21), $p(\mathbf{z}_k = D | \mathbf{x}_k^t)$, should be conditioned on the uncertain sensor state, \mathbf{x}_k^s , and written $p(\mathbf{z}_k | \mathbf{x}_k^t, \mathbf{x}_k^s)$. Hence, it should be convolved with the latest sensor state pdf, $p(\mathbf{x}_k^s | \mathbf{z}_{1:k}^s)$, to obtain $p(\mathbf{z}_k | \mathbf{x}_k^t)$ prior to using it in the update equation (2). In this paper, perfect localization is assumed so $p(\mathbf{z}_k | \mathbf{x}_k^t, \mathbf{x}_k^s) = p(\mathbf{z}_k | \mathbf{x}_k^t)$.

On a practical note when implementing an observation model, it is important that the function be smooth, and that it decreases progressively to zero without any steps. Otherwise, the objective function becomes jagged, effectively creating a multitude of local minima along the function. This quantization effect is due to the discretization of the target state PDF over the grid and has a very adverse effect for the convergence of the control optimization algorithm making it very difficult to obtain, if at all, the proper control value.

Also, because of the various assumptions made when modelling the observation likelihood, one must be aware of the possibility of discrepancies between the computed results and what would be the actual probability of detection. For computing accurately the ‘cumulative’ probability of detection (9), one would have to use an accurate observation model obtained through extensive in situ experimental testing of the search sensor. Nevertheless a theoretical model, as obtained in (21), provides a reasonable approximation for P_k , and is certainly sufficient for planning purposes, as well as for evaluating different solutions and comparing between them.

4.4 Results

For all the results presented in this section, the initial target PDF is assumed to be a symmetric Gaussian distribution centered at $x = y = 0$ with a standard deviation of 500m, and the searching vehicle is flying at an altitude of 250m, with the following initial pose $\mathbf{x}_0^s = [x_o^s = -900, y_o^s = -900, \theta_o^s = 0]$.

Stationary Target Figure 4 shows the resulting ‘greedy’ (1-step lookahead) search trajectory and the corresponding 3D views of the target PDF evolution at different stages as the search progresses from 0 to 300 seconds. Although this solution is very cheap computationally it often produces reasonable plans as it corresponds to maximizing the local payoff gradient. However because of the myopic planning, the vehicle fails to detect higher payoff values outside its sensor range and would keep spiraling further and further away from the center as can be seen on Fig. 4d. Figure 4e displays in solid line the conditional probability of detection ($1 - q_k$) obtained at every time step t_k . The dashed line represents the actual probability p_k that the target gets detected for the first time on that time step, which is the same as the solid line but discounted by Q_{k-1} . Notice the peaks in both functions as the search vehicle flyby over a mode in the PDF. Figure 4f shows the ‘cumulative’ probability P_k that the target has been detected by time step t_k . It is obtained from the integration of the payoff function (dashed line) from Fig. 4e. Another phenomenon to notice about the greedy search is the fact that because the volume under the PDF is always equal to one, as the vehicle traverses a mode of the function (e.g. when it crosses the original PDF mode for the first time (Fig. 4a), it has the effect of pushing away the probability mass hence increasing the entropy of the distribution, consequently making it harder and harder to increase the utility as time passes. The phenomenon will be referred to as the scattering effect.

Intuitively, for a given fixed trajectory length, one could imagine that instead of rushing to the PDF’s peak as in the greedy solution, the optimal strategy would be

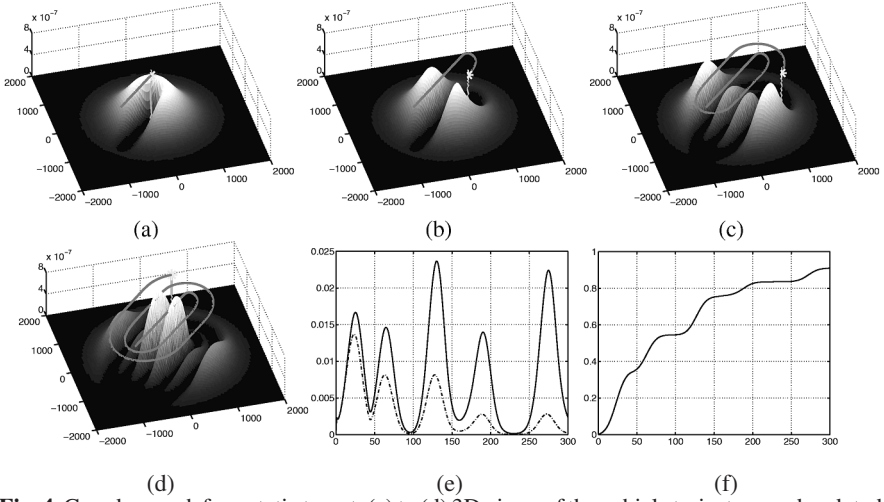


Fig. 4. Greedy search for a static target: (a) to (d) 3D views of the vehicle trajectory and updated target PDF at time $t_k = 25, 60, 180$ and 300 respectively; (e) conditional (solid line) and ‘discounted’ (dashed line) probability of detecting the target on step k ($p(\mathbf{D}_k | \mathbf{z}_{1:k}) = 1 - q_k$, and $p_k = Q_k(1 - q_k)$); (f) accumulated probability of detection P_k

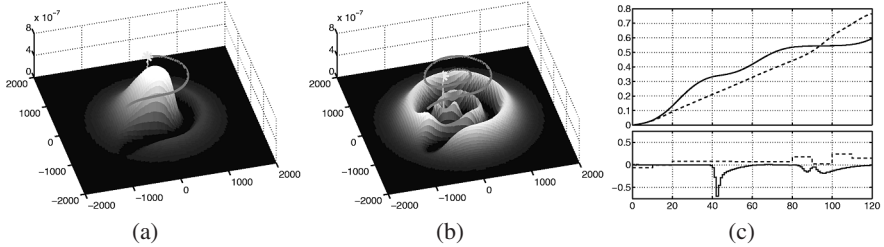


Fig. 5. Trajectory optimization: (a), (b) quasi-optimal path for a 120s search (12 control parameters maintained for 10s each) at time $t_k = 60$ and 120 respectively; (c) comparison between P_k evolutions (top), and control selections $u(k)$ ’s (bottom) for the ‘greedy’ solution (solid line), and the quasi-optimal solution (dashed line)

to circle around the peak but without flying over it, in such a manner as to plow the probability mass towards the peak, effectively compressing it (reducing the entropy), in order to increase the payoff of the last observations. In fact, as shown on Figs. 5a and 5b, this is exactly what happens. The piecewise constant ‘optimal’ control solution with 12 parameters, for a 120s trajectory, shows the path spiraling in instead of spiraling out. The comparison between the utility function evolutions (Fig. 5c) shows what one would anticipate. The greedy solution first gets a head start as it goes straight to the peak to finish with $P_{120} = .59$, but the ‘quasi-optimal’ solution progresses steadily to ultimately finish with $P_{120} = .77$, a 29% increase.

Drifting Target This section demonstrates the method for a drifting target with a process model as described in Sect. 4.2. The optimization technique is the same

used as for the static target, but the computational costs are increased by a few fold as the convolution operation needed for the target prediction stage is quite costly. This is also compounded by the fact that because the PDF is moving, a larger grid is necessary, making it even more costly to perform the convolution and the optimization. Nevertheless, the greedy solution is still very effective. The 3D plots of the search evolution are shown on Fig. 6. This time, though because of the lack of anticipation intrinsic to the greedy solution, it does not quite reach as high a level of cumulative detection probability (solid line on Fig. 8) as it did for the static case ($P_{300} = .71$ vs. $.91$). Figure 7 shows the results for a piecewise constant control

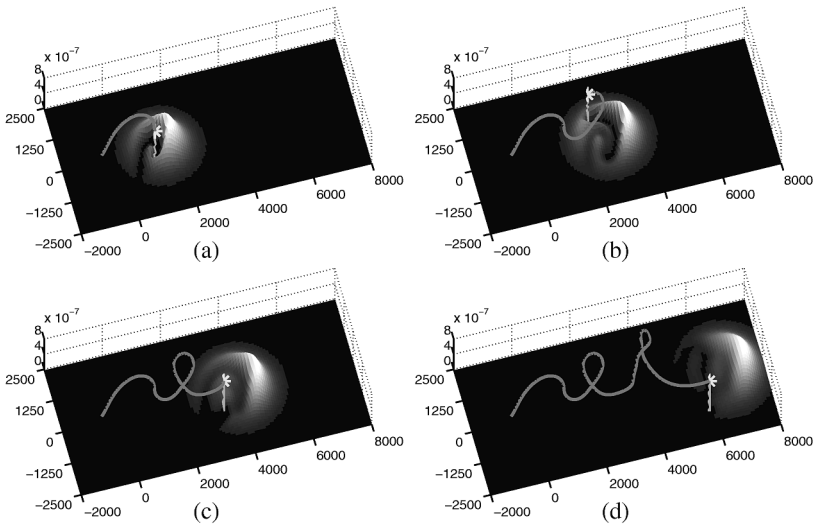


Fig. 6. Greedy (1s lookahead) search for a drifting target: (a) to (d) 3D views of the searching vehicle trajectory and updated target PDF at time $t_k = 60, 120, 180$, and 300 respectively

solution with a time horizon of 30s split into three parameters and recomputed every 10s as in a feedforward control strategy. Comparing Fig. 7a with Fig. 6a really shows the positive effect of anticipation. This effect is also seen in the ultimate value of P_k ($.87$ vs. $.71$), a gain of over 22% (Fig. 8). The computational cost though is about 25 times higher.

5 Summary and Future Work

This paper introduced a general Bayesian framework for the searching problem of a single target. The approach presented explicitly considers the search vehicle kinematics, the sensor detection function, as well as the target arbitrary motion model. It was demonstrated to find efficient search plans that maximize the probability of finding the target given a fixed time limit by maintaining an accurate target location PDF of general form, and by explicitly modelling the target's process model.

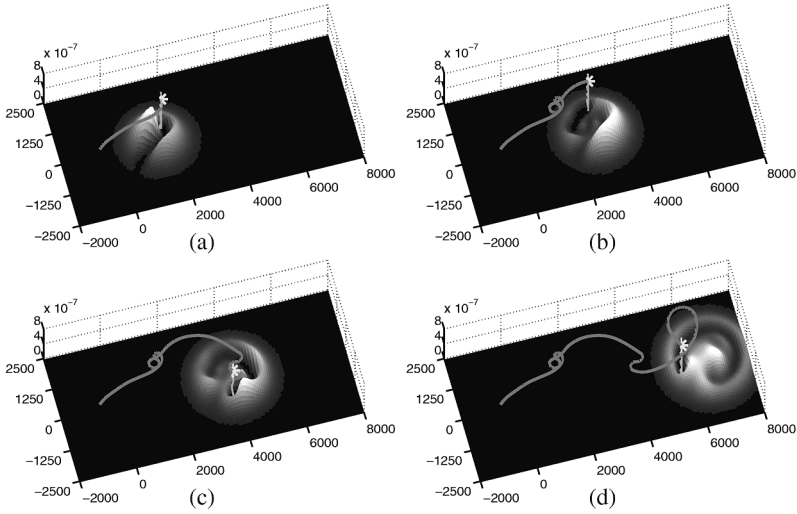


Fig. 7. Feedforward piecewise constant parametric control for a drifting target: 3D views of vehicle trajectory and updated target PDF at time $t_k = 60, 120, 180$, and 300 respectively. The planning is done with 30s lookahead (3 control parameters each maintained for 10s) and replanned every 10s.

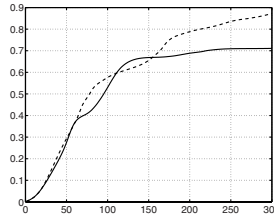


Fig. 8. Drifting target results: comparison between the detection probability P_k results from the greedy, 1s lookahead (solid line), and the piecewise constant (dashed line) solution from Figs. 6 and 7

The control solutions presented included the special case of a one step lookahead solution. This greedy solution demonstrated quite sensible trajectories for a very low computational cost. The ‘quasi-optimal’ solution for the static target was obtained with a piecewise constant control parametrization. It showed that the optimal solution for a given trajectory length initially delays its reward and tries to concentrate the probability mass into one location, reducing the entropy, and hence delaying the “scattering” effect, in order to reap greater benefits later in time. For the drifting target, it was shown that increasing the lookahead depth over the greedy solution improved the trajectory efficiency, at the expense of greater computational costs, by providing it with a sense of ‘anticipation’.

Having a sensor range much smaller than the searching area cause the target PDF to rapidly become very non-Gaussian even if it was originally the case. Because of the nature of the search problem it is very important to be able to keep track of the complete target distribution. Any grid based approach such as the one followed in this

paper is intrinsically subject to the curse of dimensionality, and as soon as one needs to increase the search area, the resolution of the grid, or the number of dimensions in the state-space, computational costs tend to get out of hand. As part of the ongoing research effort, techniques such as Monte Carlo methods, or particle filters [4], as well as the so called kernel methods are being investigated. A decentralized version of the Bayesian framework presented in this paper for a multiple vehicle search is also part of ongoing investigations.

Beyond the demonstration of the approach on a single and then multiple real autonomous platforms, the ultimate objective of this research is to eventually have multiple platforms participating in actual search and rescue (SAR) missions with real-time cooperative planning and fully integrated human in the loop inputs. As shown by the results presented, the technique has the potential to greatly improve on current SAR protocols, which in turn could be critical in saving human lives.

Acknowledgement

This work is partly supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. The authors wish to thank Ali Göktoğan from ACFR, the developer of the RMUS simulator, for his assistance with the simulation implementation. Also, thanks to Erik Wile from MIT for the wind data.

References

1. J.O. Berger. *Statistical decision theory and Bayesian analysis*. Springer series in statistics. Springer-Verlag, New York, 2nd edition, 1985.
2. T. Furukawa. Time-subminimal trajectory planning for discrete nonlinear systems. *Engineering Optimization*, 34:219–243, 2002.
3. A.H. Goktogan, E. Nettleton, M. Ridley and S. Sullarieh. Real time multi-uav simulator. In *IEEE International Conference in Robotics and Automation*, Taipei, Taiwan, 2003.
4. N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, April 1993.
5. H.J.W. Lee, K.L. Teo, V. Rehbock, and L.S. Jennings. Control parametrization enhancing technique for time optimal control problems. *Dyn. Sys. and Appl.*, 6(2):243–262, April 1997.
6. J.S. Przemieniecki. *Mathematical Methods in Defence Analyses*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 2nd edition, 1994.
7. L.D. Stone. *Theory of Optimal Search*, volume 118 of *Mathematics in Science and Engineering*. Academic Press, New York, 1975.
8. L.D. Stone, C.A. Barlow, and T.L. Corwin. *Bayesian Multiple Target Tracking*. Mathematics in Science and Engineering. Artech House, Boston, 1999.
9. K.L. Teo, C.J. Goh, and K.H. Wong. *A Unified Computational Approach to Optimal Control Problems*. Longman Scientific and Technical, 1991.
10. Debbie Whitmont. *An Extreme Event. The compelling, true story of the tragic 1998 Sydney-Hobart Race*. Random House Pty Ltd, Sydney, Australia, 1999.

Terramechanics-Based Analysis and Traction Control of a Lunar/Planetary Rover

Kazuya Yoshida¹, Toshinobu Watanabe^{1,2}, Noriyuki Mizuno¹, and Genya Ishigami¹

¹ Dept. of Aerospace Engineering, Tohoku University
Aoba 01, Sendai 980-8579, Japan
yoshida@astro.mech.tohoku.ac.jp
<http://www.astro.mech.tohoku.ac.jp>

² currently working for KOMATSU

Abstract. This paper presents analysis of traction mechanics and control of a lunar/planetary rover based on the models obtained from terramechanics. A case study has been conducted for a rover test bed to negotiate a slope of loose soil such as regolith that covers most of lunar surface. The tire traction force is modeled as a function of the vertical load and slip ratio of the wheel. Bekker's terramechanic formulae are employed to derive an improved practical model that calculates net traction force, referred to as Drawbar Pull, with a reasonable precision. Experiments are carried out in two phases. First, the physical behavior of a wheel on loose soil is observed using a single-wheel test bed, then the empirical parameters of the tire and soil are identified. Second, the slope climbing capability is studied by using a rover test bed that has independently driven four wheels. The traction margin and slip margin are defined to be used in a traction control. In the slope experiment, it turned out that the climbing capability was saturated at 14 degrees due to the lack of enough driving torque in wheels. But theoretical investigation suggests that this is not the limitation of terrain trafficability and climbing capability can be improved by increased driving torque and proper load distribution.

1 Introduction

Mobile robots, or *Rovers* play a significant role to expand our scientific knowledge in current and future lunar/planetary missions [1][2]. To achieve advanced mission goals, rovers are expected to travel much longer distance over more challenging terrains and perform more complex scientific tasks. Corresponding to such growing attention, there are an increasing number of research activities conducted in various institutions [3]-[9].

Recently, intensive research has been made on a physics based model that involves traction mechanics between the wheel and terrain. This approach recalls a classical terra-mechanics model then successfully applies it to simulate, plan and control the rover motion for improved performance [10]-[16]. For example, Iagnemma et al. [12] developed an on-line method to identify terra-mechanic parameters of soil (friction angle and cohesion stress) using onboard sensory data. Grand et al. [13] developed a sophisticated simulation model that takes the flow of loose soil under the wheel into account. The group of present authors also has been investigating

the tire-soil traction mechanics paying attention to the *slip ratio* as a key parameter [14]-[16].

In this paper, a slip based model is investigated into some details for a rover that travels over loose soil. Particularly, it is known that the surface of Moon is mostly covered by fine grained soil, called *Lunar Regolith*. The process of formation and the composition of materials can be different though, loose soils are commonly observed among planetary bodies that have a solid, non-gaseous surface structure. On loose soils, wheels are easy to slip or spin, then lose traction. The soil under the slipping wheel is removed so that the wheel sink into the soil. This *dynamic sinkage* can be modeled as a function of the slip ratio. The rolling resistance of the wheel increases according to the sinkage. Finally, the net traction force of the tire, which is referred to as *Drawbar Pull* or *DP*, is derived as a function of the slip ratio.

The slip-based traction model on loose soil is applied to evaluate the slope climbing capability of a rover. According to the slope inclination, the component of the gravity load in the tangent direction to the slope surface increases as a resistance, and the component normal to the slope decreases. Both of them result to reduce the DP. The *traction margin* and *slip margin* are defined to be used as a state index in a traction controller. At a point where the increased gravity resistance balances the maximum achievable value of DP, there remains no traction margin or slip margin. This is the point to give the limitation in terms of maximum terrain trafficability.

Experiments are carried out in two phases. First, the physical behavior of a wheel on loose soil is observed using a single-wheel test bed, then the empirical parameters of the tire and soil are identified. Second, the slope climbing capability is studied by using a rover test bed that has independently driven four wheels. Through the experiments, the validity of the developed model is confirmed in the evaluation of the traction forces with a reasonable precision.

2 Modeling of Traction Mechanics

2.1 Slip Ratio

As a key variable to describe the state of the tire traction, the slip ratio s of each wheel is defined as follows:

$$s = \begin{cases} (r\dot{\theta}_w - v_w)/r\dot{\theta}_w & (r\dot{\theta}_w > v_w : \text{accelerating}) \\ (r\dot{\theta}_w - v_w)/v_w & (r\dot{\theta}_w < v_w : \text{braking}) \end{cases} \quad (1)$$

where

- r : radius of the wheel
- θ_w : rotation angle of the wheel ($\dot{\theta}_w = \omega$)
- $r\dot{\theta}_w$: tire circumference velocity
- v_w : traveling velocity of the wheel

In this definition, the slip ratio is positive when the vehicle is accelerating and negative when braking.

2.2 Tire-Soil Interaction Mechanics

Figure 1 depicts a general model of a wheel on deformable soil. For such a model, a set of formulae have been established in the filed of terramechanics [17][18]:

$$\tau(\theta) = (c + \sigma(\theta) \tan \phi) \left(1 - \exp\left[-\frac{r}{k} \{\theta_f - \theta - (1 - s)(\sin \theta_f - \sin \theta)\}\right]\right) \quad (2)$$

where

- $\sigma(\theta)$: normal stress under the wheel
- $\tau(\theta)$: share stress under the wheel
- c : cohesion stress of the soil
- ϕ : internal friction angle of the soil
- b : wheel width
- k : share displacement under the wheel

By integrating σ and τ over the entire contact area, from θ_f to θ_r , we obtain the normal and tangential forces exerting on the wheel:

$$W = rb \left\{ \int_{\theta_r}^{\theta_f} \sigma(\theta) \cos \theta d\theta + \int_{\theta_r}^{\theta_f} \tau(\theta) \sin \theta d\theta \right\} \quad (3)$$

for normal force that is balanced to the vertical load,

$$DP = rb \left\{ \int_{\theta_r}^{\theta_f} \tau(\theta) \cos \theta d\theta - \int_{\theta_r}^{\theta_f} \sigma(\theta) \sin \theta d\theta \right\} \quad (4)$$

for net tangential force that is called *Drawbar Pull*, and

$$T = r^2 b \left\{ \int_{\theta_r}^{\theta_f} \tau(\theta) d\theta \right\} \quad (5)$$

for the wheel's driving torque.

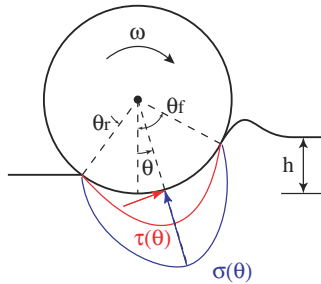


Fig. 1. A tire model on deformable soil

2.3 Normal Stress Distribution

In the above equations, if the distribution of normal stress under the wheel $\sigma(\theta)$ is given, everything will be determined. However, the determination of the stress distribution is relatively complicated.

There are several models proposed for this stress distribution, by Wong [17] for example, though, we alternatively propose the following model for better representation of experimental data:

$$\sigma(\theta) = \sigma_{max} \left(\frac{\cos \theta - \cos \theta_f}{\cos \theta_m - \cos \theta_f} \right)^n \quad (6)$$

for $\theta_m < \theta < \theta_f$, and the distribution profile is symmetric for $\theta_r < \theta < \theta_m$.

Here, θ_m is an angle at which maximum normal stress σ_{max} is observed.

$$\theta_m = c_1 s \theta_f \quad (7)$$

$$\sigma_{max} = \left(\frac{k_c}{b} + k_\phi \right) (r)^n (\cos \theta_m - \cos \theta_f)^n \quad (8)$$

2.4 Static and Kinetic Sinkage

In Equation (8), the term in the first parentheses is from Bekker's formula of a static stress p under a flat plate:

$$p = \left(\frac{k_c}{b} + k_\phi \right) h_s^n \quad (9)$$

where h_s is a static sinkage.

If Equation (9) is applied to a wheel, the following equation is obtained.

$$W = \int_{-\theta_s}^{\theta_s} \left(\frac{k_c}{b} + k_\phi \right) (r (\cos \theta - \cos \theta_s))^n r b d\theta \quad (10)$$

$$h_s = r(1 - \cos \theta_s) \quad (11)$$

From these equations, the tire-ground contact angle θ_s with the static sink and the static sinkage h_s are numerically obtained when the vertical load W is given.

On the other hand, the kinetic sinkage in a steady state, h_k , can be modeled by

$$h_k = c_2 s \quad (12)$$

Once the static and kinetic sinkages are determined, the front contact angle θ_f is kinematically given by:

$$\theta_f = \cos^{-1} \left(1 - \frac{h_s + h_k}{r} \right) \quad (13)$$

Finally, the rear contact angle θ_r is modeled using a soil restitution ratio c_3 .

$$\theta_r = c_3 \theta_f \quad (14)$$

2.5 Procedure to Obtain DP

Here, a procedure to obtain DP is summarized.

1. The vertical load W and slip ratio s are given as input.
2. θ_s is obtained from Equation (10), then h_s is obtained from (11).
3. h_k is obtained from Equation (12), then θ_f and θ_r are determined from Equations (13) and (14), respectively.
4. θ_m is determined from Equation (7), then σ_{max} is determined from (8).
5. The normal stress distribution $\sigma(\theta)$ is determined from Equation (6).
6. From Equation (1), $\tau(\theta)$ is determined.
7. Finally, by substituting $\sigma(\theta)$ and $\tau(\theta)$ into Equation (4), the Drawbar Pull is obtained.

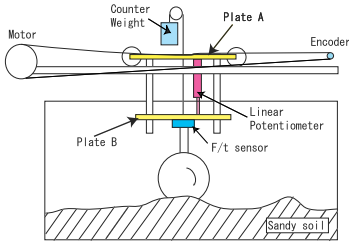


Fig. 2. A schematic of the single wheel test bed

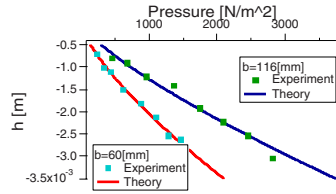


Fig. 3. Static sinkage for different plate widths: theory and experimental plots (dry sand)

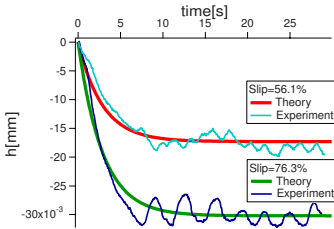


Fig. 4. Time history of kinetic sinkage for different slip ratios: theory and experimental plots (dry sand)

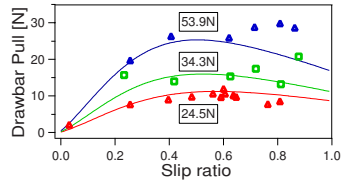


Fig. 5. Drawbar Pull for different vertical loads: theory and experimental plots (dry sand)

3 Experimental Identification of Tire-Soil Parameters

For the verification of the above traction model and the identification of parameters, experiments were carried out using a single wheel test bed.

Figure 2 depicts a schematic of the test bed.

In this test bed, the wheel rotates and translates with arbitrary velocity respectively so that the traction at various slip conditions is observed. The measurements are made for wheel's rotation velocity, translational velocity, vertical distance (sinkage), and the vertical and horizontal forces by a force sensor (described as F/T sensor in Figure 2).

The experiments were carried out for both dry sand and simulated material for lunar soil called *Regolith Simulant*.

The followings are typical experimental results for dry sand. Figure 3 depicts the experimental results of the static sinkage for different plate width to verify Equation (9). From this result, the parameters k_c , k_ϕ , n were identified.

Figure 4 depicts the time history of the wheel sinkage with different slip ratios. These time histories can be model by a first-order delay. Each sinkage converges to a value corresponding to each slip ratio at the steady state, and thus the constant c_2 of Equation (12) is identified.

Figure 5 depicts the profile of the Drawbar Pull as a function of the slip ratio for different vertical loads. The experimental plots are relatively scattered although, the plots are fit in the theoretical curves.

For the *Regolith Simulant*, the same characteristics were observed but each parameter shows clear difference from dry sand. The identified parameters for both dry sand and *Regolith Simulant* are summarized in Table 1.

Table 1. Identified terrain-wheel parameters

	Dry Sand	Regolith Simulant	
c	0.6	0.8	[kN/m ²]
ϕ	27	38	[deg]
r	0.09	0.09	[m]
b	0.1	0.1	[m]
k	0.2	0.1	[m]
k_c	6.2×10^5	1.8×10^6	[N/m ⁿ⁺¹]
k_ϕ	1.1×10^4	2.0×10^6	[N/m ⁿ⁺¹]
n	1.14	1.53	
c_1	0.4	0.6	
c_2	0.03	0.03	[m]
c_3	1.1	1.1	

4 Rover Test Bed

The rover test bed was developed at Tohoku University with four metal wheels. The rover has the dimension of 0.95 [m](length) \times 0.70 [m](width) \times 0.56 [m](height) and weighs about 25 kg in total (see Figure 6.) The wheels are the same that was tested in the single wheel test bed. Each wheel, in the diameter of 0.09 [m], is

made of aluminum and surrounded by paddles of 0.01 [m] height. Driving motor is embedded inside the wheel.

In the main body, compact CPU boards and motor drivers are mounted to control the driving torque of the wheels using pulse width modulation (PWM.) The CPU cards communicate with a host computer through wired (RS-232C) or wireless (Ethernet) connection.

5 Slope Climbing Experiment

A number of experiments have been carried out to test the performance in traversing various natural and artificial terrains. Among them, this paper focuses on the experiments to negotiate a slope of the simulant.

Figure 6 shows a picture of the experimental setup with the rover. The facility, located at National Aerospace Laboratories, Mitaka, Japan, comprises a flat rectangular vessel in the size of 1.5 by 2.0 meters filled with 10 cm deep of the simulant. The vessel can be inclined up to 30 degrees. The soil can be smoothed by an automated tool so that a consistent condition is provided for each experiment.

During the experiments, the angular velocity of each wheel was measured by an encoder. The traveling velocity of the rover was measured by an optical sensor system. Then the slip ratio of each wheel was monitored.

Figure 7 depicts a typical raw data for one of the wheels. The wheel was at rest at the beginning, then started to rotate. The angular velocity was controlled to keep a constant value. The rover located on a slope was gradually accelerated then the traveling velocity was saturated at a certain value. As a result, the plot of the slip ratio begins from 1, then gradually decrease and converge to a certain level, around 0.3 in this case, at the steady state.

If this steady state slip ratio is plotted on the curve of the drawbar pull, Figure 8 is obtained. This figure suggests that the net traction force can be increased by accepting higher value of the slip ratio. We propose to term the margins for such increments *traction margin* and *slip margin*, respectively. The slip ratio can be increased by giving a higher angular velocity or a larger driving torque in the wheel. In order to achieve the maximum value of the traction force, a controller can be designed to regulate the wheel velocity, or driving torque, using the slip margin as a reference information. Note that, if the slip ratio increases beyond the peak, the drawbar pull goes down, and the wheel will start spinning. We should avoid such a situation.

6 Maximum Slope Climbing Capability

As shown in Figure 9, according to the increasing slope inclination θ , the component of the gravity load in the tangent direction to the slope surface, $W \cos \theta$, increases as a gravity resistance, and the component normal to the slope decreases by $W \sin \theta$. This normal component is distributed between from the front (upper) and rear (lower) wheels in the inverse proportion to L_1 and L_2 . By this effect, the normal load on the front wheel becomes relatively small, then yielding a smaller DP.



Fig. 6. A setup for slope climbing experiments by a 4-wheel drive rover test bed on Regolith Simulant

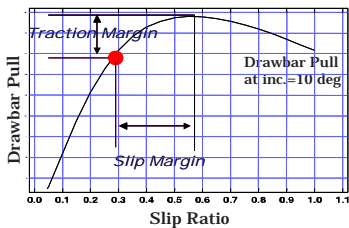


Fig. 8. The definition of *slip margin* and *traction margin*

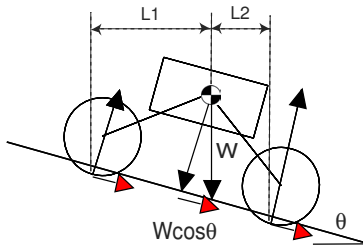


Fig. 9. A model of a rover on a slope

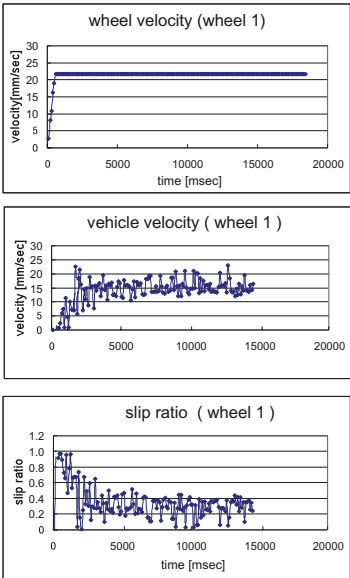


Fig. 7. A typical raw data from the slope climbing experiment

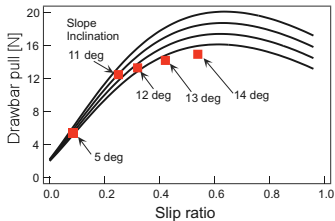


Fig. 10. Theoretical (solid lines) and Experimental (boxes) DP for various slopes

Figure 10 depicts the curves of DP of the front wheel for different slope inclinations. The highest curve is for 11 degrees, and the lowest one is for 14 degrees. The steady state slip ratios observed in the experiments are also plotted in the same figure. The plots show that the steady state slip increases, in the other word, the slip margin decreases, according to the slope inclination. And at the inclination of 14 degrees, the remaining traction and slip margins become close to zero. Eventually in the experiment, the rover could not climb more than 14 degrees due to the lack of enough driving torque in the wheels. However, this angle is not the limitation

of terrain trafficability, since the theoretical curve suggests that there remain some margins.

In order to improve the climbing capability, one suggestion is of course to increase the driving torques in the wheels. Another suggestion is to relocate the position of the rover's center of the gravity according to the slope inclination so that L_1 and L_2 of Figure 9 become even, then the load distribution are equalized for all wheels. This can be done by an external moving mass or an active suspension mechanism.

7 Conclusions

In this paper, the authors have investigated the slip-based traction mechanics of a lunar/planetary rover that travels over natural rough terrain. Special attention was made on the tire-soil mechanics on loose soil. Classical terramechanic formulae were employed to develop a detailed model by which the net traction force, Drawbar Pull, was predicted from a slip condition with a certain accuracy.

Experiments were carried out with a single wheel test bed to observe the physical phenomena of the soil. The relationship of Drawbar Pull versus the slip ratio was verified by the experiments. Specific parameters to characterize the soil and tire traction have been also identified.

Slope climbing performance of a rover on Regolith Simulant was studied by experiments. The traction margin and slip margin were defined to be used in a traction control, and the limitation of the climbing angle was given at zero margins. To exploit maximum terrain trafficability, the rover should have enough power and a sort of load equalization mechanism.

References

1. <http://mars.jpl.nasa.gov/MPF/> (as of June 2003)
2. <http://mars.jpl.nasa.gov/missions/> (as of June 2003)
3. C.R.Weisbin,et. al.; "Autonomous Rover Technology for Mars Sample Return," *Proc. i-SAIRAS'99*, ESTEC, The Netherlands, June, 1999, pp.1–10.
4. J.Aizawa, N.Yoshioka, M.Miyata, Y.Wakabayashi; "Designing of Lunar Rovers for High Work Performance," *Proc. i-SAIRAS'99*, ESTEC, The Netherlands, June, 1999, pp.63–68.
5. Y.Kuroda, K.Kondo, K.Nakamura, Y.Kunii, T.Kubota; "Low Power Mobility System for Micro Planetary Rover *Micro5*," *Proc. i-SAIRAS'99*, ESTEC, The Netherlands, June, 1999, pp.77–82.
6. M.Tarokh, Z.Shiller, S.Hayati; "A Comparison of Two Traversability Based Path Planners for Planetary Rovers," *Proc. i-SAIRAS'99*, ESTEC, The Netherlands, June, 1999, pp.151–157.
7. J. Balaram; "Kinematic Observers for Articulated Rovers," *Proc. 2000 IEEE Int. Conf. of Robotics and Automation*, CA, USA, April, 2000, pp.2597–2604.
8. S.Hayati, R.Arvidson; "Long Range Science Rover (Rocky7) Mojave Desert Field Tests," *Proc. i-SAIRAS'97*, Tokyo, Japan, July, 1997, pp.361–367.
9. W.R.Whittaker, D.Bapna, M.W.Maimone, E.Rollins; "Atacama Desert Trek: A Planetary Analog Filed Experiment," *Proc. i-SAIRAS'97*, Tokyo, Japan, July, 1997, pp.355–360.

10. Farritor, S., Hacot, H., and Dubowsky, S.; "Physics-Based Planning for Planetary Exploration," *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, 278-283, 1998.
11. Iagnemma, K., and Dubowsky, S., "Mobile Robot Rough-Terrain Control (RTC) for Planetary Exploration," *Proceedings of the 26th ASME Biennial Mechanisms and Robotics Conference*, DETC 2000.
12. Iagnemma, K., Shibly, H., Dubowsky, S., "On-Line Traction Parameter Estimation for Planetary Rovers," *Proceedings of the 2002 IEEE Int. Conf. on Robotics and Automation*, pp. 3142-3147, 2002.
13. C.Grand, F.Ben Amar, P.Bidaud, "A Simulation System for Behaviour Evaluation of Off-road Mobile Robots," *4th International Conference on Climbing and Walking Robots*, Sept. 2001, Germany.
14. K.Yoshida, H.Asai and H.Hamano, "Motion Dynamics of Exploration Rovers on Natural Terrain: Experiments and Simulation," *Proc. of the 3rd International Conference on Field and Service Robotics*, Finland, 281-286, June, 2001.
15. K. Yoshida, H. Hamano; "Motion Dynamics of a Rover With Slip-Based Traction Model," *Proc. 2002 IEEE Int. Conf. on Robotics and Automation*, pp.3155-3160, 2002.
16. K. Yoshida, H. Hamano; "Motion Dynamics and Control of a Planetary Rover With Slip-Based Traction Model," *SPIE 16th Int. Symp. on Aerospace/Defense Sensing, Simulation, and Controls*, SPIE4715-33, 2002.
17. Wong, J. Y., *Theory of Ground Vehicles*, John Wiley & Sons, 1978, chapter 2.
18. Bekker, G., *Introduction to Terrain-Vehicle Systems*, University of Michigan Press, 1969.

Topological Analysis of Robotic N-Wheeled Ground Vehicles

Michel Lauria¹, Steven Shooter², and Roland Siegwart³

¹ Laborius, Sherbrooke University
Sherbrooke, QC J1K 2R1, Canada
michel.lauria@usherbrooke.ca
<http://www.gel.usherbrooke.ca/laborius/>

² Dept. of Mechanical Engineering, Bucknell University
Lewisburg, PA 17837, USA
shooter@bucknell.edu
<http://www.bucknell.edu/>

³ Autonomous Systems Lab, Swiss Federal Institute of Technology
Lausanne, 1015, Switzerland
roland.siegwart@epfl.ch
<http://asl.epfl.ch/>

Abstract. Robotic ground vehicles are systems that use gravity and contact forces with the ground to perform motion. In this paper we will focus on n-wheeled vehicles able to perform motion with all the wheels maintaining contact at the same time. The main goal of this work is to establish the implication of the topological architecture of the vehicle mechanism on criteria such as climbing skills, robustness, weight, power consumption, and price. Tools will be provided to help the robot designer to understand the implications of important design parameters like the number of wheels, the vehicle mechanism, and the motorisation of joints on the above criteria. Two examples of innovative locomotion concepts for rough terrain are presented and discussed.

1 Introduction

The locomotion subsystem of a mobile robot determines the range of operation of the whole system. Building mobile robots able to autonomously maneuver obstacles in rough terrain is a very complex mechatronic task. The fact is that locomotion can be achieved in a lot of different ways. The most typical classification divides land locomotion into four areas: wheeled, tracked, legged and all others [1]. In this paper we will focus on Wheeled Locomotion Mechanisms (WLM) able to perform ground following motion with all the wheels maintaining contact with the ground at the same time and independently of its profile.

This paper is divided in two main sections. First we will establish a two dimensional (2D) generalized model of WLM including both kinematics and dynamics. Equations will be formulated in a way that they will be ready to be implemented to do simulations using standard linear algebra functions. In the second section we will discuss design issues using two concrete examples of realized WLM.

2 Modelling WLM

2.1 Hypothesis

In this paper, WLM are studied in a 2D world. The ground is modelled as a rigid 2D body with a fixed shape S . The WLM are modelled as a set of b rigid bodies and w rigid wheels interconnected with p pivots (revolute joints). The wheel pivot is coincident with the center of mass of the wheel. Each wheel of the WLM is constrained to touch the ground with exactly one contact point. The other bodies do not touch the ground. In addition all wheels have the same radius r and are constrained to roll along the terrain profile without sliding. Therefore, all the wheel's centers describe the same trajectory T that is entirely defined by S and r .

2.2 Mechanism Topology and Notations

Any WLM can be described by the topological structure of its joint interconnections [2]. Numbering the different entities of the model is necessary to achieve such a description. Our description distinguishes between wheels and other bodies. Our numbering scheme follows these simple rules:

- ground is considered as the body with number zero
- begin the numbering of the pivots with those connected to the wheels
- begin the numbering of the bodies with those connected to the wheels
- give the same numbering to a wheel and its associated pivot

Topology can be described with a set of simple mathematical relationships. For example, Figure 1b represents the complete set of rules associated to the WLM depicted in Figure 1a. The notations used in the figures and equations will now be explained. The center of mass of the i^{th} body b_i will be denoted by G_i . The j^{th} pivot will be denoted by P_j . Contact points, centers and radius of ground curvature, and contact angles for the k^{th} wheel w_k will be denoted by C_k , O_k , $s_k \rho_k$ and α_k respectively with $s_k = 1$ if the ground is convex or -1 if the ground is concave.

2.3 Kinematics of WLM

The velocity state of WLM can be described using spatial vectors $\mathbf{v} [n \times 1]$ [3] containing the linear velocities of pivots \mathbf{v}_P , the linear velocities of the bodies centers of mass \mathbf{v}_G , the angular velocities of the wheels ω_w and the angular velocities of the bodies ω_b (1). The acceleration state can also be described by an analog vector $\mathbf{a} [n \times 1]$ (2). Equation (3) give the dimension n of these vectors.

$$\mathbf{v} = \left[\dots (\mathbf{v}_{P_j})^T \dots (\mathbf{v}_{G_i})^T \dots \omega_{w_k} \dots \omega_{b_i} \dots \right]^T \quad (1)$$

$$\mathbf{a} = \left[\dots (\mathbf{a}_{P_j})^T \dots (\mathbf{a}_{G_i})^T \dots \dot{\omega}_{w_k} \dots \dot{\omega}_{b_i} \dots \right]^T \quad (2)$$

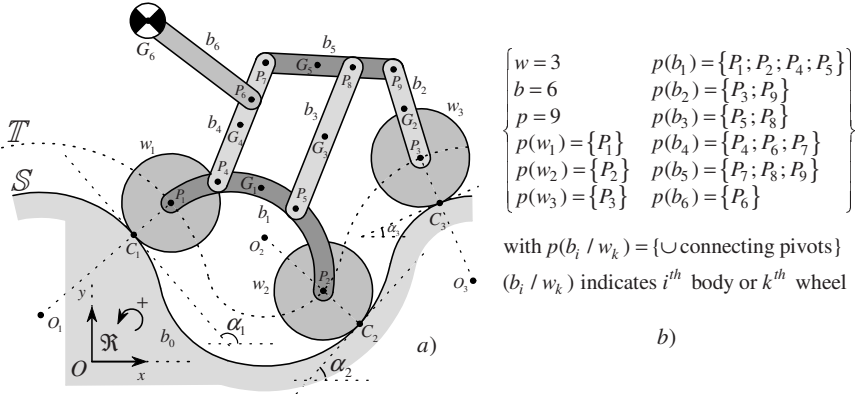


Fig. 1. Representations of a WLM's topological structure (a) Graphical (b) Mathematical

$$n = w + 2p + 3b \quad (3)$$

Kinematic constraints for the k^{th} wheel ($k = 1 \dots w$) are given by (4) and (5) while the rigid body conditions for the i^{th} body and the j^{th} pivot ($i = 1 \dots b$ and $P_j \in p(b_i)$) are given by (6) and (7).

$$\mathbf{v}_{P_k} = r\omega_{w_k} \cdot \cos \alpha_k \sin \alpha_k \quad \mathbf{T} \quad (4)$$

$$\mathbf{a}_{P_k} = r\dot{\omega}_{w_k} \cdot \cos \alpha_k \sin \alpha_k \quad \mathbf{T} - \frac{r^2 (\omega_{w_k})^2}{(r + s_k \rho_k)} \cdot -\sin \alpha_k \cos \alpha_k \quad \mathbf{T} \quad (5)$$

$$\mathbf{v}_{G_i} - \mathbf{v}_{P_j} = \omega_{b_i} \mathbf{e}_\perp \times \overrightarrow{P_j G_i} \quad (6)$$

$$\mathbf{a}_{G_i} - \mathbf{a}_{P_j} = \dot{\omega}_{b_i} \mathbf{e}_\perp \times \overrightarrow{P_j G_i} - (\omega_{b_i})^2 \cdot \overrightarrow{P_j G_i} \quad (7)$$

There are a total of $2e$ equations that can be written in a more compact matrix form (8) (9) using the spatial vectors \mathbf{v} and \mathbf{a} and matrix $\mathbf{G} [e \times n]$ and $\mathbf{c} [e \times 1]$.

$$\mathbf{G} \cdot \mathbf{v} = \mathbf{0} \quad (8)$$

$$\mathbf{G} \cdot \mathbf{a} = \mathbf{c} \quad (9)$$

$$e = 4p \quad (10)$$

These two sets of equations are linear and underconstrained with regard to \mathbf{v} and \mathbf{a} . This reflects the fact that the WLM is free to move and the degree of mobility m is given by (11). We can always decompose the total mobility of a WLM into three different parts:

- The ground-wheel's mobility is always equal to 1
- The wheel's internal mobilities can vary from 0 to $w - 1$

- The others internal mobilities can vary from 0 to infinity

$$m = n - \text{rank}(\mathbf{G}) = n - e = w + 3b - 2p \quad (11)$$

In the example of Figure 1 the total mobility is 3. The vehicle can roll along the terrain. In addition the articulations allow the third wheel to roll independently of the two other wheels. A weight-shifting mechanism given by the movable mass at G_6 provides the third independent mobility to this WLM.

The null space of \mathbf{G} contains the infinite set of possible velocity states for given posture and contact angles (Fig. 3a). Parametrisation is done with an arbitrary velocity vector $\mathbf{p}_v [m \times 1]$. Acceleration state is the sum of two components (Fig. 3b):

- the specific acceleration that depends on contact angles, ground curvatures, WLM's posture and speed state
- the free acceleration. Parametrisation is done with an arbitrary acceleration vector $\mathbf{p}_a [m \times 1]$.

2.4 Dynamics of WLM

We would like to establish the dynamic model of a WLM. Bodies and wheels are subject to different kinds of internal/external applied/kinematically-constrained forces/torques (Fig. 2).

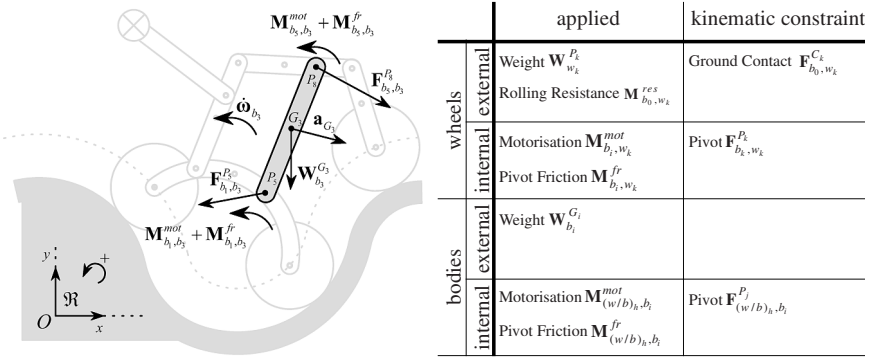


Fig. 2. Graphical representation of forces and torques acting on *body3*

We can represent the wrenches of the pivot forces, contact forces and motorisation torques using three spatial vectors $\mathbf{f}^P [4p \times 1]$, $\mathbf{f}^C [2w \times 1]$ and $\mathbf{m}^{mot} [2p \times 1]$ defined by (12) (13) (14) respectively.

$$\mathbf{f}^P = \left[\dots \left(\mathbf{F}_{(w/b)_i, (w/b)_i}^{P_j} \right)^T \dots \right]^T \quad (12)$$

$$\mathbf{f}^C = \left[\dots \left(\mathbf{F}_{b_0, w_k}^{C_k} \right)^T \dots \right]^T \quad (13)$$

$$\mathbf{m}^{mot} = \left[\dots \left(\mathbf{M}_{(w/b)_h, (w/b)_i}^{mot} \right) \dots \right]^T \quad (14)$$

We write now the $3w$ Newton-Euler equations for the wheels, the $3b$ Newton-Euler equations for the bodies and the $2p$ Action-Reaction equations for the pivot forces in a matrix form (15). Euler equation for body b_i has to be written with respect to center of mass G_i .

$$\mathbf{D}^P \mathbf{D}^C \cdot \left[(\mathbf{f}^P)^T (\mathbf{f}^C)^T \right]^T = \mathbf{M} \cdot \mathbf{a} - \mathbf{f}^0 - \mathbf{D}^{mot} \cdot \mathbf{m}^{mot} \quad (15)$$

with:

\mathbf{M} : WLM masses and inertias matrix $[(3w + 3b + 2p) \times n]$

\mathbf{f}^0 : acceleration-independent applied generalized forces

\mathbf{D}^{mot} : one/zero coefficients matrix which multiply the vector \mathbf{m}^{mot} . It depends of the mechanism topology

\mathbf{D}^C : coefficients matrix which multiply the vector \mathbf{f}^C . It depends of contact angles and posture

\mathbf{D}^P : coefficients matrix which multiply the vector \mathbf{f}^P . It depends of mechanism topology and posture

With the p Action-Reaction equations for the motorisation torques (16), the $2p$ dependent variables of $\mathbf{m}^{mot} [2p \times 1]$ can be substituted by a vector of p independent variables $\mathbf{p}^{mot} [p \times 1]$ (17).

$$\mathbf{R}^{mot} \cdot \mathbf{m}^{mot} = \mathbf{0} \quad (16)$$

$$\mathbf{m}^{mot} = \text{null}(\mathbf{R}^{mot}) \cdot \mathbf{p}^{mot} \quad (17)$$

with

\mathbf{R}^{mot} : coefficients matrix which multiply the vector \mathbf{m}^{mot} . It depends of mechanism topology

After some calculations [12] the forward dynamic model (Fig. 3.c) in which the accelerations are calculated from the motorisation is given by (18). The inverse dynamic model (Fig. 3.d) in which the motorisation is calculated from the free acceleration parameters $\mathbf{p}_a [m \times 1]$ and arbitrary parameters $\mathbf{p}_m [(p - m) \times 1]$ is given by (19).

$$\mathbf{a} = \text{pinv}(\mathbf{G}) \cdot \mathbf{c} + \text{null}(\mathbf{G}) \cdot \text{pinv}(\mathbf{A}) \cdot (\mathbf{C} - \mathbf{B} \cdot \mathbf{p}^{mot}) \quad (18)$$

$$\mathbf{m}^{mot} = \text{null}(\mathbf{R}^{mot}) \cdot (\text{pinv}(\mathbf{B}) \cdot (\mathbf{C} - \mathbf{A} \cdot \mathbf{p}_a) + \text{null}(\mathbf{B}) \cdot \mathbf{p}_m) \quad (19)$$

with

$$\mathbf{A} = (\text{null} \mathbf{D}^P \mathbf{D}^C)^T \cdot \mathbf{M} \cdot \text{null}(\mathbf{G}) \quad (20)$$

$$\mathbf{B} = -(\text{null} \mathbf{D}^P \mathbf{D}^C)^T \cdot \mathbf{D}^{mot} \cdot \text{null}(\mathbf{R}^{mot}) \quad (21)$$

$$\mathbf{C} = (\text{null} \mathbf{D}^P \mathbf{D}^C)^T \cdot (\mathbf{f}^0 - \mathbf{M} \cdot \text{pinv}(\mathbf{G}) \cdot \mathbf{c}) \quad (22)$$

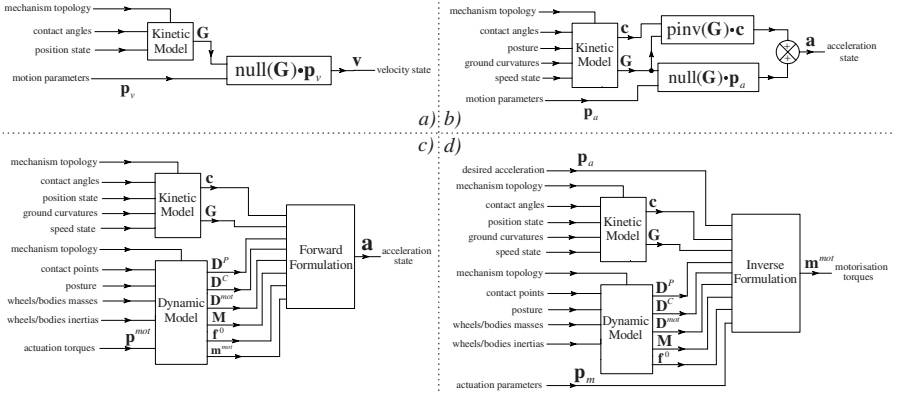


Fig. 3. Models of WLM (a) Speed Kinematic (b) Acceleration Kinematic (c) Forward Dynamic (d) Inverse Dynamic

3 Designing WLM

In this section tools will be provided to help the robot designer to understand the implication of important design parameters like the number of wheels, the climbing strategies and the motorisation of joints. We will focus on quasi-static locomotion and emphasise our argument using our two WLM designs as examples (Fig. 4).

Shrimp [7] [9] is a 6-wheeled ground vehicle based on a 3 DOF passive suspension mechanism. With this design, no sensor based control is necessary to maintain ground contact with all the wheels. The distribution of tangential contact forces is done passively but can be optimised with on board active control and sensors for contact properties estimation (gyro, joint position sensors) [5].

Octopus [8] [15] is a 8-wheeled ground vehicle based on a (6 DOF active + 1 DOF passive) suspension mechanism. The autonomous coordination of the active 14 DOF is based on the on-board integration of inclinometer, joint position sensors and tactile wheels able to sense ground contact properties (angle, force). With this design, active control can distribute the contact forces to minimise tangential forces and increase traction [2]. This decreases the need for friction to climb obstacles.

3.1 Number of Wheels

Statically the forces supporting the weight of a WLM can be provided by the ground in two different ways:

- the normal unilateral contact forces based on the non interpenetration of matter
- the tangential bilateral contact forces based on Coulomb's friction laws

For a n -wheeled WLM with given posture and ground contact angles the solution to the 3 static Newton-Euler equations (23) is given by (24) as described in [14]. The dimension space d of the solution increases with the number of wheels w (25). Thus,

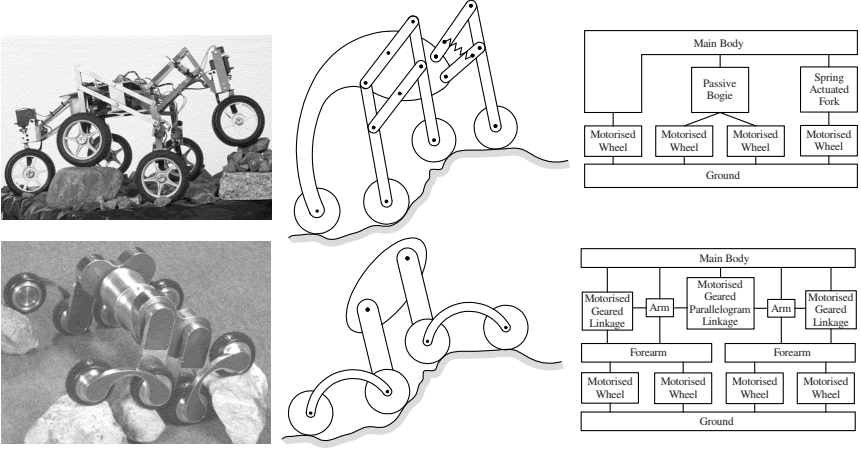


Fig. 4. Shrimp WLM and Octopus WLM

there are theoretically d infinite sets of possible contact wrenches, S^0 , equilibrating the weight and parametred by $\mathbf{p}^S [d \times 1]$.

$$\mathbf{S}^C \cdot \mathbf{f}^C = -m\mathbf{g}^T \mathbf{0}^T \quad (23)$$

$$\mathbf{f}^C = \text{pinv}(\mathbf{S}^C) \cdot -m\mathbf{g}^T \mathbf{0}^T + \text{null}(\mathbf{S}^C) \cdot \mathbf{p}^S \quad (24)$$

$$d = 2w - 3 \quad (25)$$

However, a given n-wheeled WLM with specified ground contact angle configuration is not necessarily able to reach all the theoretically possible equilibrating contact wrenches. There are four overlapping sub-spaces of solutions:

- $S^1 \subset S^0$ given by the physical limits of the contact conditions excluding wheel slip and loss of contact.
- $S^2 \subset S^1$ reachable by a specific mechanism topology.
- $S^3 \subset S^2$ reachable by the motorisation implementation.
- $S^4 \subset S^3$ reachable autonomously and stabilised by a controller which is based on perception of the environment.

It's clear that the chances to find a valid equilibrating contact wrench increase with the number of wheels. In other words, WLM with high number of wheels will be naturally more stable in rough terrain than those with less [10]. Contrarily, each wheel added increases the overall complexity, weight and price.

3.2 Climbing Strategies

Climbing strategy is the second key issue when designing WLM. The three main groups of climbing strategies are (Fig. 5):

- controlled/uncontrolled traction distribution
- controlled location of Center of Gravity (COG)
- controlled/uncontrolled support distribution

Often a WLM combines several different climbing strategies. Implementation of controlled strategies based on perception of the environment increases the climbing skills, but also the complexity, weight and price. Uncontrolled strategies are more robust because climbing is achieved in open loop without perception. However design of efficient uncontrolled WLM like the Shrimp [7][9] require more engineering time for the kinematic linkage optimization. Moreover, particular attention must be given to the mass distribution and the payload placement with this kind of WLM.

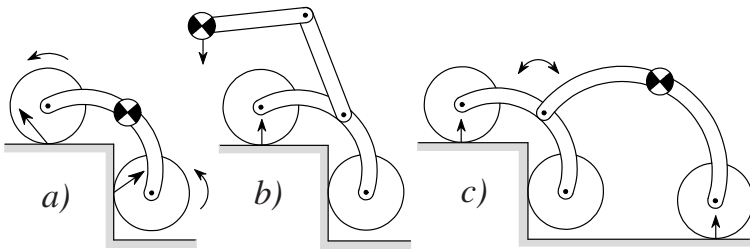


Fig. 5. Basic Climbing Strategies (a) friction based climbing (b) Unbalanced COG climbing (c) Active Support Distribution

3.3 Motorisation

Choosing the appropriate actuators for the pivots is the third key issue. Motorisation is classified in three main groups:

- Active motorisation: a transducer converts a given non mechanical energy source into mechanical power. Output force/torque can be controlled independently of the WLM state.
- Passive motorisation: Springs are introduced in the mechanism. The output force/torque depend of the vehicle's posture.
- No motorisation

Active motorisation is necessary for controlled climbing strategies. Each active motor added increases weight and complexity. For example the 6 DC motors actuating the arms and forearms of Octopus WLM [8][15] are situated inside the main body allowing a better control of the global COG position. Passive motorisation is energetically more efficient but the output force/torque is not controllable and introduces oscillations in internal degrees of freedom.

3.4 Perception and Control

Controlled climbing strategies require sensors and embedded computers. The WLM posture can be measured with joint sensors while the direction of the gravity field is obtained using inclinometers.

Determination of contact points of the wheels to surfaces is more difficult. An estimation can be made from joints sensors in the wheels and articulating body parts under the assumption of no slip and maintained contact. However, this method works only for uncontrolled support distribution WLM like Shrimp. Optimal controlled wheel-traction distribution based on this sensing method was first developed in [4] and applied with success to the Sojourner robot [5]. A three dimensional odometry algorithm has been developed and implemented for the Shrimp by [6]. But to perform actively controlled support distribution, more sophisticated contact sensors are required. We developed an integrated contact sensor based on the wheel's tire deformation measurement (Fig. 6) [8][15]. Eight of these tactile wheels have been integrated in the Octopus WLM. Information about wheel contact points allows an algorithm to find the optimal contact forces distribution and the corresponding necessary actuation.

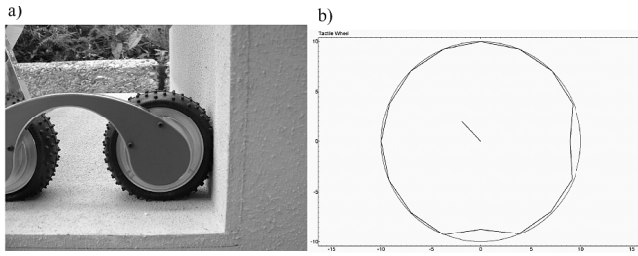


Fig. 6. (a) Tactile wheel of Octopus in contact with a step (b) What sees the sensor

When the motorisation of one articulation of a controlled WLM is not reversible or when there is friction due to a geared linkage transmission mechanism, it is necessary to integrate force sensors in the structure [11] so that control of contact forces can be achieved in closed loop based on chassis internal forces measurement. With our tactile wheel design, normal contact force can be approximated knowing the rigidity of the tire. For Instance, Octopus is able to perform active terrain adaptation autonomously for small obstacles of the size of one wheel. The lack of precision of force measurement with our actual tactile wheel prototype and the friction in the forearm motorisation linkage limit the size of the obstacles that can be overcome autonomously.

4 Conclusion and Outlook

This paper presents a study of climbing Wheeled Locomotion Mechanisms (WLM). A formulation of kinematics and dynamics for fast prototyping 2D simulations

has been proposed. A classification between controlled/uncontrolled active/passive designs has been presented and discussed using two WLM examples Shrimp [7][9] and Octopus [8][15]. The techniques presented in this paper have been used for the design, development and control of these two WLM. Passive WLM solutions are now mature enough for real applications like space exploration [5][13]. However, active WLM solutions demonstrate potential climbing skills that cannot be equalled passively. Enhanced integration of sensors, actuators and advanced embedded control algorithms will lead to greater applications for future field and service robotics applications.

References

1. G. Muscato, G. Nunnari, S. Guccione, G. S. Virk, A. K. M. Azad, A. Semerano, M. Ghriissi, T. White, C. Glazebrook, "Robots for Volcanos: The state of the art", Third International Conference on Climbing and Walking Robots, Madrid, Spain, 2000.
2. M. H. Hung, D. E. Orin, "Efficient Formulation of the Force Distribution Equations for General Tree-Structured Robotic Mechanisms with a Mobile Base", *IEEE Transactions on Systems, man and Cybernetics*, Part B: Cybernetics, Vol 30, No 4, 2000.
3. R. Featherstone, D. Orin, "Robot Dynamics: Equations and Algorithms", *International Conference on Robotics and Automation*, San Francisco, CA, 2000.
4. K. Iagnemma, S. Dubowsky, "Vehicle Wheel Ground Contact Angle Estimation: With application to mobile robot traction control", *7th International Symposium On advances in Robot Kinematics*, Piran-Portoroz, Slovenia, 2000.
5. K. Iagnemma, S. Dubowsky, "Mobile Robot Rough-Terrain Control (RTC) for Planetary Exploration", *26th ASME Biennial Mechanisms and Robotics Conference*, DETC, 2000.
6. P. Lamon, R. Siegwart, "3D-Odometry for rough terrain Towards real 3D navigation", *International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.
7. R. Siegwart, P. Lamon, T. Estier, M. Lauria, R. Piguet, "Innovative Design for Wheeled Locomotion in Rough Terrain", *Journal of Robotics and Autonomous Systems*, vol 40/2-3, pp. 151–162, 2002.
8. M. Lauria, Y. Piguet, R. Siegwart, "Octopus - An Autonomous Wheeled Climbing Robot", *International Conference on Climbing and Walking Robots*, Paris, France, 2002.
9. M. Lauria, T. Estier, R. Siegwart, "An innovative Space Rover with Extended Climbing Abilities", *Video Proceedings of the International Conference on Robotics and Automation*, San Francisco, USA, 2000.
10. S. Hirose, "Considerations on the design of Hyper-Redundant Versatile Robotic System", *Proceedings of TITech COE/Super Mechano Systems Workshop* 98, pp. 12–17, 1998.
11. C. Grand, F. Ben Amar, F. Plumet, Ph. Bidaud, "Stability Control of a Wheel-Legged Mini-Rover", *International Conference on Climbing and Walking Robots*, Paris, France, 2002.
12. M. Lauria, *Nouveaux concepts de locomotion pour véhicules tout-terrain robotisés*, ThLse no 2833, Ecole Polytechnique Fédérale de Lausanne, 2003.
13. S. Michaud, A. Schneider, R. Bertrand, P. Lamon, R. Siegwart, M. Van Winnendael, A. Schiele, "SOLERO: Solar-Powered Exploration Rover," *7th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, The Netherlands, 2002.
14. V. Kumar, K.J. Waldron, "Force Distribution in Walking Vehicles," *Transactions of the ASME*, vol. 112, pp. 90–99, 1990.
15. Octopus Web Page, <http://asl.epfl.ch/research/systems/Octopus/octopus.php>

Development of a Control System of an Omni-directional Vehicle with a Step Climbing Ability

Daisuke Chugo¹, Kuniaki Kawabata², Hayato Kaetsu², Hajime Asama³ and Taketoshi Mishima¹

¹ Saitama University
255, Shimo-Ookubo, Saitama-shi, Saitama 338-8570, Japan
chugo@riken.go.jp / mishima@ics.saitama-u.ac.jp

² RIKEN (The Institute of Physical and Chemical Research)
2-1, Hirosawa, Wako-shi, Saitama 351-0198, Japan
kuniakik@riken.go.jp / kaetsu@riken.go.jp

³ The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8904, Japan
asama@race.u-tokyo.ac.jp

Abstract. We proposed a new holonomic mobile mechanism which is capable of running over the step. This mechanism realizes omni-directional motion on flat floor and passes over non-flat ground in forward or backward direction. The vehicle equips seven omni-directional wheels with cylindrical free rollers and two passive body axis that provide to change the shape of the body on the rough terrain. This paper presents a method to control the wheels for passing over rough terrain with the stable posture. Our vehicle is required to keep synchronization among its wheels for climbing the step without slipping and blocking. Therefore, in this paper, an algorithm of synchronization among all wheels is proposed. The performance of our system is experimented by means of computer simulations and experiments using our prototype vehicle.

1 Introduction

In recent years, mobile robots are expected to perform various task in general environment such as nuclear power plants, large factories, welfare care facilities and hospitals. However, there are a lot of narrow spaces with steps and slopes in such environments, and it is difficult for general car-like vehicles to run around.

Generally for effective task execution, it is required to realize quick and efficient mobile function. The omni-directional mobile capability is useful for the tasks in narrow spaces, because there is no holonomic constraint on its motion [1]. On the other hand, it is required to run over the irregular terrain. Both of these capabilities are required to compose high mobility system for field and service robots.

In related works, various types of omni-directional mobile robots are proposed: legged robots, ball-shaped wheel robots, crawler robots, and so on. The legged robots [2] can move in all directions and pass over rough terrain. However, its mechanism and control system tend to be complicated and energy efficiency is generally not so high. The robot with ball-shaped wheels can run in all directions [3], however, it cannot run on the rough grounds. The special crawler mechanism [?] is also proposed

for the omni-directional mobile robot. It can move on the rough terrain, but it cannot climb over large steps.

For realization of running in narrow spaces and on irregular terrain, we developed a new holonomic omni-directional vehicle with step-climbing ability. [5] Our prototype utilized the rocker-bogie suspension system and also has the redundant actuators, and it is important to keep synchronization among the wheels. However, during the vehicle is climbing steps, it is difficult to synchronize the wheels without slipping and blocking.

For wheel control, we utilized the traction control. Several traction control methods for mobile robots to pass over the rough terrain were already proposed [6,7]. These methods consider to control only single wheel and do not address the problem related to synchronization among plural wheels.

Our proposed method realizes that the mobile robot passes over the irregular terrain with synchronization among the wheel rotations. The key idea of our method is that not only the single wheel state but also the plural wheel states are utilized as feedback value. We verify the performance of the proposed control method through the computer simulations and experiments.

This paper is organized as follows: we discuss the mechanical design, the kinematic model of the robot in section 2; the new control method is proposed in section 3; we show the implementation and experimental results in section 4; section 5 is conclusion of this paper.

2 System Configuration

2.1 Mechanical Design

We already developed a prototype mobile mechanism [5] and also an advanced prototype vehicle system (Figure 1 and 2). The vehicle has seven wheels with DC motors. The size of the vehicle is 750mm(L) x 540mm(W) x 520mm(H) and the total weight is approximately 22 [kg] with the batteries.

The mobile mechanism consists of seven special wheels with free rollers and a rocker-bogie suspension system. The special wheels consist of twelve cylindrical free rollers (Figure 2) [8] and realize to generate the omni-directional motion. Generally, on the viewpoint of energy efficiency of running on the structured terrain [9], the wheeled mobile system is better than the other type of mobile systems (e.g., legged or crawler type). Thus, our mobile mechanism realizes the omni-directional function and high-energy efficiency, compatibly.

Our mechanism utilizes the rocker-bogie suspension system, which consists of passive links, and it can adapt itself to rough terrain. [6,10]. No sensors and no additional actuators are equipped to pass over irregular terrain. In general environment, it is not easy task to estimate terrain condition precisely. Our system realizes such estimate function only using passive body axis.

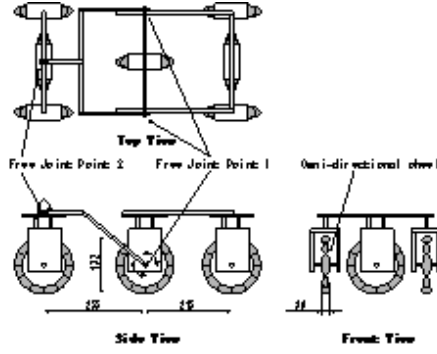


Fig. 1. Overview of the mechanism

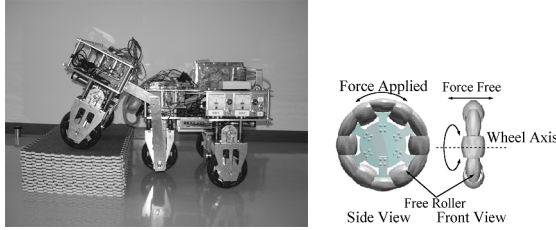


Fig. 2. Prototype Vehicle and Special wheel

2.2 Kinematics

The vehicle's configuration, position and attitude are defined by the body parameters: R_1 , R_2 and wheel rotation velocity values $(\omega_1, \dots, \omega_7)$, in Figure 3.

Here, equation(1) indicates the wheel rotation velocity.

$$\omega_i = kV_i (i = 1, \dots, 7) \quad (1)$$

where,

r : radius of the wheel [mm]

ω_i :rotation velocity of the wheel i [rad/s]

V_i :rotation velocity of the actuator i [rad/s]

k : gear ratio between the actuator and the wheel

Now, $\dot{X} = \dot{x} \ \dot{y} \ \dot{\theta}^T$ and $V = V_1 \ \dots \ V_7^T$ express the motion velocity vector of the vehicle and the rotation velocity vector of the actuators, respectively. V is also derived by using \dot{X} in equation (2).

$$V = J^+ \cdot \dot{X}, \quad (2)$$

where J^+ is pseudo inverse of Jacobian matrix;

$$J^+ = (J^T J)^{-1} J^T = \frac{1}{kr} \cdot \begin{bmatrix} 1 & 0 & R_2 \\ 0 & -1 & R_1 \\ -1 & 0 & R_2 \\ 1 & 0 & 0 \\ 1 & 0 & R_2 \\ 0 & 1 & R_1 \\ -1 & 0 & R_2 \end{bmatrix} \quad (3)$$

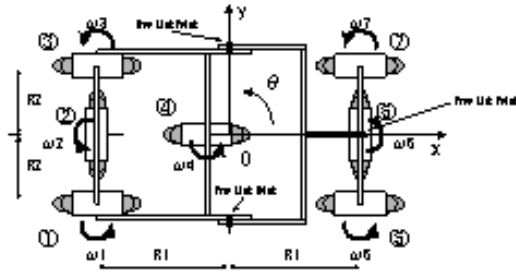


Fig. 3. Coordination and parameters

2.3 Problem Specification

The developed vehicle has redundant actuation system using seven wheels. Thus, our system has to synchronize the wheels with following each control reference, which is calculated by equation (2) using Jacobian. However, during the robot passes over the irregular terrain, the load distribution to each wheel is complex problem. Therefore, it is difficult to synchronize among the wheel. If the system fails to take the synchronization among the wheels, the vehicle will lose the balance of the body posture as shown in Figure 4.

Thus, each wheel has to synchronize with the others when the vehicle runs on the rough terrain. In related works, some traction control methods for single wheel are already proposed. However, they do not discuss synchronization of the wheels for running on rough terrain. For our system, we must consider the synchronization among the wheels. We explain our proposed method in next section.

3 Control System

3.1 Proposed method

In order to synchronize the wheels rotation during the vehicle passes over the step, calculated torque reference value should not over the maximum torque of the

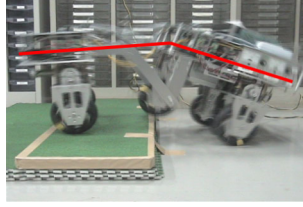


Fig. 4. The vehicle losing the balance

motor. If extraordinary load applies on the wheel(s) or the torque reference exceeds maximum torque of the motor, the system cannot control the wheels, properly.

Our proposed control system is shown in Figure 5. The control reference is calculated by PID-based control system (equation (4)).

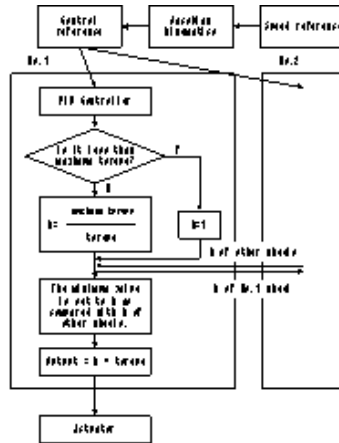


Fig. 5. Flow chart of the control system

The torque reference of i -motor is calculated by equation (4).

$$\tau_i = k_p e + k_i \int e dt + k_d \frac{de}{dt}, \quad (4)$$

where

e :Error value of the motor rotation velocity

k_p :Proportional gain for PID controller

k_i :Integral gain for PID controller

k_d :Derivative gain for PID controller

The coefficient k_i is calculated as:

$$k_i = \begin{cases} \frac{\tau_{\max}}{\tau_i} & \text{if } \tau_i > \tau_{\max}, \\ 1 & \text{if } \tau_i \leq \tau_{\max}, \end{cases} \quad (5)$$

where

τ_{\max} : Maximum torque of the motor
 τ_i : Calculated torque value
 i : 1 ... 7 (number of an actuator)

The reference torque is determined by equation (6):

$$\tau_i^{out} = k \times \tau_i, \quad (6)$$

where $k = \min \{k_1, \dots, k_7\}$.

The controller adjusts the synchronization among the wheel in the case of extraordinary load occurring.

3.2 Simulation

We verify the performance of our method by computer simulations. As initial conditions, three motors are rotating at same fixed velocity speed 100[deg/s] and the load applies to each motor independently. The load is approximated by a dumper model and the dumper coefficients are applied to each wheel as follows.

load A : 0.001[Nm/deg] from 37 to 60[sec]
 load B : 0.004[Nm/deg] from 14 to 52[sec]
 load C : 0.005[Nm/deg] from 22 to 57[sec]

In this case, we assume that the maximum torque of the motor is 30[N].

The results of the simulation are shown in Figure 6. During the load applied to the wheel (from 14 to 60[sec]), rotation velocity of the motor is reduced. Using proposed method, each controller adjusts control command to the wheel and recovers synchronization among the wheel.

3.3 Method of Sensing the Step

We utilized PID based control system, however it is difficult to determine the parameters of the controller when the control target has complex dynamics. Thus, we switch two parameter sets according to the situations. The vehicle has the accurate control mode for the flat floor and the posture stability control mode for the rough terrain. The stability mode utilizes the proposed traction control method, too.

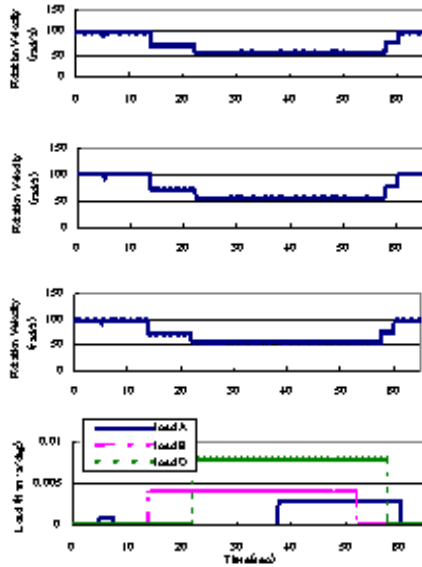


Fig. 6. Simulation result

In order to switch two parameter sets, the terrain estimation function is required. Thus, we proposed the estimation method using the body axes. The angle of two axes of the body is changed passively by the ground surface. The terrain can be measured by using the body kinematics information. Two potentiometers measure the angle of the axes (Figure 7). By this information, the controller can switch two parameter sets according to the terrain condition.

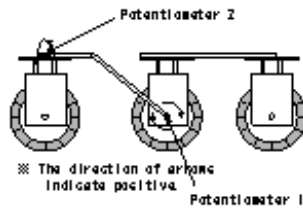


Fig. 7. Two Potentiometers

4 Experiment

Here, we have the following two experiments.

4.1 Measuring the Step

In first experiment, we verify the sensing ability of the vehicle when it passes over the rough ground. The vehicle climbs the step with 30[cm] depth and 1[cm] height.

The experimental result is shown in Figure 8 and it indicates that the height of the step is 0.9[cm] and the depth is 32.5[cm]. Our vehicle need to change the control mode when the step is more than 3[cm] [7], it is enough step detection capability.

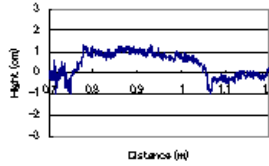


Fig. 8. The result of measuring the step

4.2 Passing Over the Step

Second experiment is for passing over the steps. The vehicle moves forward at 0.3[m/s] and passes over the 5[cm] height step. Furthermore, we compare the result by our proposed method with the one by general PID method.

As the result of this experiment, the vehicle can climb up the step more smoothly by our method (Figure 9). The white points indicate the trajectory of the joint point on the middle wheel and they are plotted at every 0.3 [sec] on Figure 9.

Figure 10 and 11 show the disturbed ratio which means the error ratio of the rotation velocity (a), the slip ratio (b) [11] and the rotation velocity of each wheel (c). The disturbed rotation ratio and the slip ratio are defined by the equation (5) and the equation (6), respectively.

$$\hat{d} = \frac{\omega_{ref} - \omega}{\omega} \quad (7)$$

$$\hat{s} = \frac{r\omega - v_{\omega}}{r\omega} \quad (8)$$

ω :Rotation speed of the actuator.

ω_{ref} :Reference of rotation speed.

r : The radius of the wheel.

v_{ω} :The vehicle speed.

As the result, the rotation velocity of the wheels is synchronized with the proposed control method. Furthermore, the disturbed rotation ratio and the slip ratio are reduced. Thus, this control method is efficiency for step climbing.

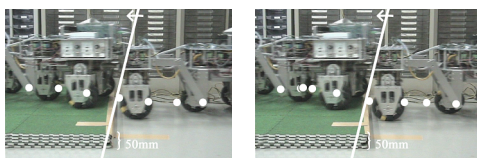


Fig. 9. Step climbing with proposed controlling and general controlling

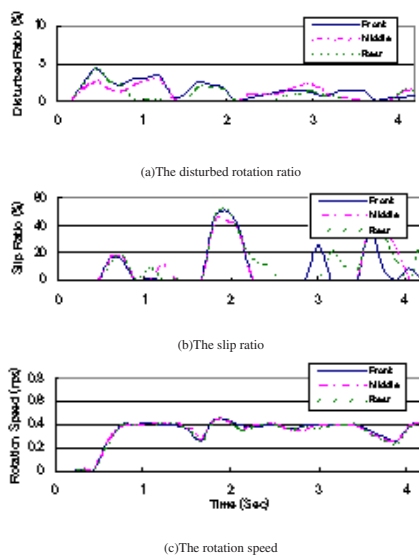


Fig. 10. Experimental Result of proposed method

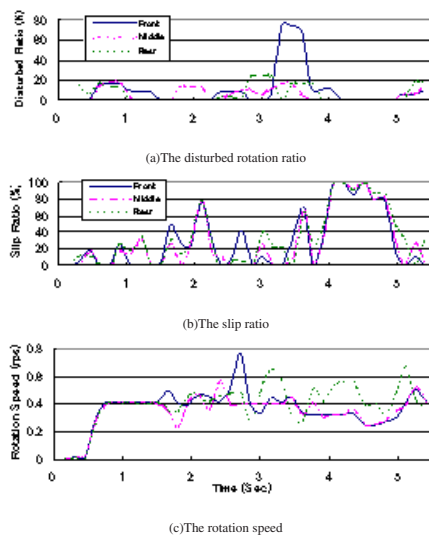


Fig. 11. Experimental Result of old method

5 Conclusions

In this paper, we discuss the control method for omni-directional mobile vehicle with step-climbing ability and the terrain estimation method using its body. We also designed new control system which realized the synchronization among the wheels when the vehicle passed over rough terrain.

We implemented the system and verified its effectiveness by the simulations and experiments. For future works, we will consider the motion planning method based on the environment information.

References

1. G. Campion, G. Bastin and B.D. Andrea-Novel, "Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots," *IEEE Transactions on Robotics and Automation*, vol. 12, No. 1, pp. 47–62, 1996.
2. G. Endo and S. Hirose, "Study on Roller-Walker: System Integration and Basic Experiments," *IEEE Int. Conf on Robotics & Automation*, Detroit, Michigan, USA, pp. 2032–2037, 1999.
3. M. Wada and H. Asada, "Design and Control of a Variable Footpoint Mechanism for Holonomic Omnidirectional Vehicles and its Application to Wheelchairs," *IEEE Transactions on Robotics and Automation*, vol. 15, No. 6, pp. 978–989, 1999.
4. S. Hirose and S. Amano, "The VUTON: High Payload, High Efficiency Holonomic Omni-Directional Vehicle," *6th Int. Symposium on Robotics Research*, Hidden Valley, Pennsylvania, USA, pp. 253–260, 1993.
5. A. Yamashita, *et.al.*, "Development of a step-climbing omni-directional mobile robot," *Int. Conf. on Field and Service Robotics*, Helsinki, Finland, pp. 327–332, 2001.
6. K. Iagnemma, *et.al.*, "Experimental Validation of Physics-Based Planning and Control Algorithms for Planetary Robotic Rovers," *6th Int. Symposium on Experimental Robotics*, Sydney, Australia, pp. 319–328, 1999.
7. K. Yoshida and H. Hamano, "Motion Dynamic of a Rover With Slip-Based Traction Model," *IEEE Int. Conf on Robotics & Automation*, Washington DC, USA, pp. 3155–3160, 2001.
8. H. Asama, *et.al.*, "Development of an Omni-Directional Mobile Robot with 3 DOF Decoupling Drive Mechanism," *IEEE Int. Conf on Robotics & Automation*, Nagoya, Japan, pp. 1925–1930, 1995.
9. T. Estier, *et.al.*, "An Innovative Space Rover with Extended Climbing Abilities," *Video Proc. of Space and Robotics 2000*, Albuquerque, New Mexico, USA, 2000.
10. <http://mars.jpl.nasa.gov/MPF/> (as of Nov. 2003)
11. D. Chugo, *et.al.*, "Development of Omni-Directional Vehicle with Step-Climbing Ability," *IEEE Int. Conf on Robotics & Automation*, Taipei, Taiwan, pp. 3849–3854, 2003.

Sensor-Based Walking on Rough Terrain for Legged Robots

Yasushi Mae¹, Tatsuhi Mure¹, Kenji Inoue¹, Tatsuo Arai¹, and Noriho Koyachi²

¹ Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, JAPAN
mae@sys.es.osaka-u.ac.jp
<http://www-arailab.sys.es.osaka-u.ac.jp>

² Intelligent Systems Institute, National Institute of Advanced Industrial Science and Technology
Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568, JAPAN
n.koyachi@aist.go.jp
<http://staff.aist.go.jp/n.koyachi/koyachi-j.html>

Abstract. A simple sensor-based walking on rough terrains for legged robots using an acceleration sensor attached to the body is described. The algorithm is implemented to a developed proto-type robot with limb mechanism, which has six limbs that can be used for both locomotion and manipulation. The six limbs are arranged on the body radially to have uniform property in all directions. This symmetrical structure allows the robot to generate a gait trajectory for omnidirectional locomotion in a simple manner. The trajectory of the sensor-based walking is obtained by a small conversion of this simple trajectory. The proto-type robot walks on the uneven ground while adjusting the pose of the body to keep high stability margin. Finally, adequate footholds of supporting limbs are examined for manipulation tasks by two neighboring limbs of the robot.

1 Introduction

Outdoor robots require high manipulability and mobility in the application of rescue, construction, agriculture, and space or ocean developments [1–6]. Thus they should have both handling and mobile mechanism. Most of the conventional working robots have been designed in such a manner that a manipulator is simply mounted on a mobile platform [7,8]. Legged robots suit for work in rough terrains because they can select any landing points and keep stability of the pose during manipulation. In considering the advantage of legged robots, there may be many approaches to design locomotion mechanism of robot may be taken.

There are some animals or insects that can use their legs for their hands to manipulate objects dexterously while walking on rough terrains. If manipulation and locomotion functions can be integrated into one linkage called "limb", both high terrain adaptability and manipulability are achieved. Based on this notion, the concept "Limb Mechanism" has been proposed that has integrated locomotion and manipulation functions into one limb [9–14].

We call a robot with limb mechanism "limb mechanism robot". It is required for a limb mechanism robot to walk in any directions quickly and smoothly while

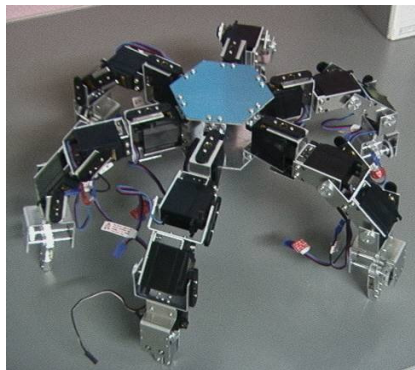


Fig. 1. A limb mechanism robot.

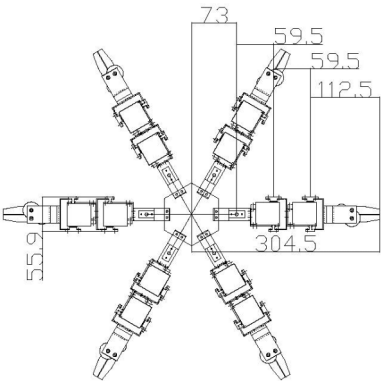


Fig. 2. Radial arrangement of limbs.

keeping its stability. A limb mechanism robot has been designed and developed taking such omnidirectional mobility into account [14].

As one of feasible structures of the limb mechanism a 6-limb mechanism has been analyzed and evaluated in the aspects of omnidirectional mobility [15,16]. In [15,16], two types of structures are compared with respect to their stroke, stability, and error of dead reckoning for six-legged locomotion. The radial leg arrangement model will be proved to have higher omnidirectional mobility than the parallel leg arrangement.

In actual tasks it is essential for a limb mechanism robot to move on rough terrains quickly and smoothly. Furthermore, in manipulation tasks, a limb mechanism robot has to select adequate footholds of supporting limbs not to fall down due to manipulation motions of limbs.

In the present paper, first we introduce a limb mechanism robot, and describe mainly following two topics. one is a simple trajectory generation method in considering gait control strategy on the uneven ground. It can maintain the walking speed of the robot while keeping high stability, even when a transfer limb lands on a bump. The other is adjustment of footholds of four supporting limbs from the point of view of static stability. The footholds should be selected to keep higher stability margin when two limbs are used as manipulation. In the paper, we examine the case two neighboring limbs are used as arms, which makes the limb mechanism robot unstable the most.

2 Limb Mechanism Robot

2.1 Configuration of Limb Mechanism Robot

First, we introduce a limb mechanism robot developed by Takahashi et al.[14,15] (see Fig.1). In designing, the main concern is to fix the number of limbs. A four limb mechanism will be feasible, but its mobility is extremely limited while one of limbs will be employed for arm function. Too many limbs, for example seven or

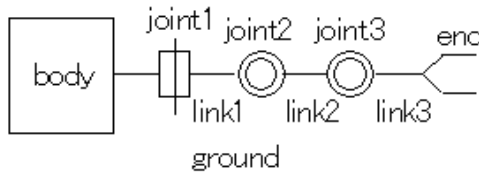


Fig. 3. Configuration of a limb.

more, may cause difficulty in gait control. A six limb mechanism seems the most reasonable in the aspects of achieving high mobility and manipulability, since it enables four-legged locomotion with two arm manipulation in addition to six-legged locomotion.

Arrangement of limbs are important parameter to determine property of a limb mechanism robot. In most insects, two groups of three legs are arranged in parallel. This parallel arrangement has strong directivity in walking and working capabilities. No directivity, or omnidirectional mobility, may be preferable when the robot is applied in a narrow environment where posture and rotational motions are constrained. Especially in rough terrains, omnidirectional mobility is useful to obtain stable and quick change of walking directions. If six legs are arranged equally or equilaterally, then the directivity and the interference among legs can be improved and a large working space can be assured for each limb.

Thus, the limb mechanism robot has been developed to have six limbs which are arranged radially (see Fig.2). Each limb has a 3 d.o.f. serial linkage. The structure is a rotation-pivot-pivot articulation as shown in Fig.3. We call the joints first, second, and third joints in order from the body to the end.

2.2 Overview of Control System

In outdoor working, remote control is desirable because control cables disturb a robot to work smoothly. Since the developed limb mechanism robot has 18 d.o.f., calculation of the amount of control of each actuator in the whole generation of operation becomes very complicated. Although what has a highly efficient computer for control is required, such a computer cannot be carried in a main part from the point of a size and weight, and sufficient calculation cannot be performed by computer which can be conversely put on a main part.

Thus we adopt a radio control system. Figure 4 shows the overview of the whole system. This system consists of a transmitter, a receiver, and servomotors. The transmitter is connected with the control computer. We can drive the servomotors in proportion to the position commands which we input to the computer. The servomotors of the robot are controlled in open loop. The robot is equipped with the sensor module. It is constituted by an acceleration sensor, a PIC(a small CPU with A/D converter), and a transmitter. The PIC processes the sensed gravity to obtain inclina-

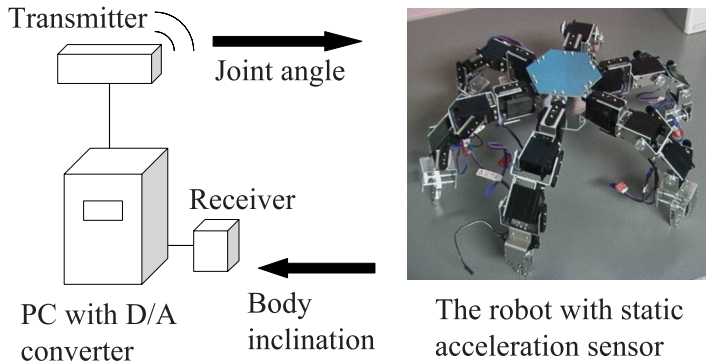


Fig. 4. Overview of the control system.

tion of the body and transmits it to the PC. The PC generates the motion pattern of the robot and transmits the corresponding position commands to the servomotors.

3 Basic Walking Trajectory

3.1 Simplified Trajectory Generation

As indicated in the previous discussion, it is rather complicated and tiresome to find the trajectory of each leg for the generation of the gait pattern. A more general treatment will be considered to find unique trajectory in any directions in each limb. The method is simple and easy to generate trajectories with the same stroke in any directions.

Figure 5 shows one example of the largest size of a circled area in the working space of the limb. Any linear trajectories will be possible in any directions within it, therefore the trajectory generation for each limb might become much easier both in four and six-legged locomotion. A control software is implemented in the controller PC. Actual omnidirectional walking motion has been confirmed in the developed robot.

The diameter of the circle is determined from the workspace of a limb. Figure 6 shows sectional view of the workspace of a limb. The workspace is between outer curved line and inner curved line. We assume a cylinder inscribed in the workspace to generate basic trajectory of a limb (see Fig.6). Then observe two perimeters of circles and draw two perpendiculars from perimeters' end to end. These lines make the basic trajectory.

The method makes it possible to make the stroke of limbs the same length as every direction. The stroke is diameter of a circle. Usually on regular terrains, the end of limb passes on this trajectory. The trajectory consists of four terms, lift up, forward motion, landing, and backward motion. They are shown in Fig.7(a). As shown in Fig.7(b), in case of changing moving direction of the robot, simple rotation of the

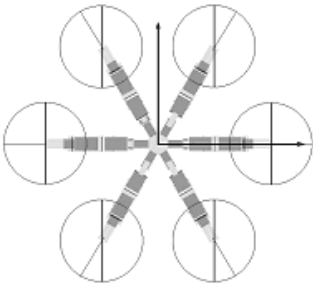


Fig. 5. Stroke in omnidirectional locomotion.

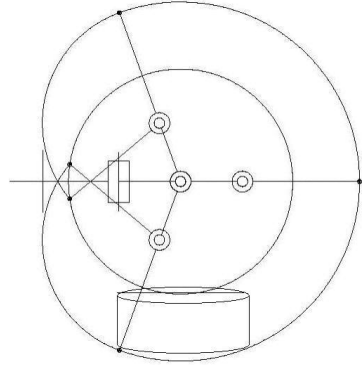
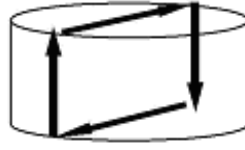


Fig. 6. Cylinder inscribed in workspace.



(a) Regular trajectory.



(b) Trajectory in changing moving direction.

Fig. 7. Basic trajectory of a limb.

trajectory makes omnidirectional locomotion easy, whereby the limb mechanism robot can change the moving direction without re-stepping.

3.2 Simplified Gait Pattern by Phase Shift

Leg mechanisms like insects or animals have symmetry in longitudinal and lateral directions, thus their stability margin and stroke varies in $90[deg.]$ phase, or it may be called four axes symmetry. In the gait control the trajectory of each leg may be generated only for this phase, and it can be repeatedly used in the other directions by taking this symmetry into account. In our limb mechanism robot the trajectories in $60[deg.]$ phase can be repeated due to its six axes symmetry. This may allows simpler control strategy even in four-legged gait as well as in six-legged. The omnidirectional gait may be generated simply by switching the basic gait patterns six times. Furthermore, the patterns has symmetry centred in one limb, thus they are again reduced to a half, that is, the trajectories in $30[deg.]$ phase are only required for the gait control.

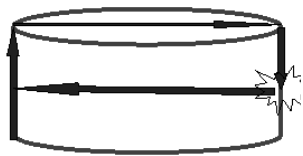


Fig. 8. Trajectory on a bump.



Fig. 9. Pose adjustment to reduce inclination of the body.

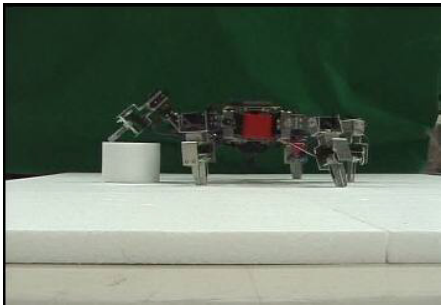


Fig. 10. Limb mechanism robot on a bump with pose adjustment.

4 Sensor-Based Waling on Rough Terrain

On bumpy or rough terrains, it is difficult for the robot to continue walking by the basic trajectory described in the previous section. When some limbs land to bumps or into hollows, the body inclines and it reduces stability margin. In the worst case, the robot falls down. Thus, it is necessary to adjust the pose of the body in walking on the uneven ground.

Easy conversion of the basic trajectory should make the robot possible to walking on the uneven ground while keeping omnidirectional mobility. We describe a trajectory generation method for walking on the uneven ground, which uses an acceleration sensor attached to the body to measure the inclination of the body. In the sensor-based trajectory generation, the following process is added to the basic trajectory generation process.

If the inclination is detected in "landing" term, the landing motion of the limb is stopped and the supporting limbs are moved to reduce the inclination. After adjusting inclination, some ends of limbs may reach the end of working space and cannot move any more. Then, in order to bring the ends of the limb into the working space and keep the height of the body constant, all landing limbs should be folded as shown in Fig.9. After folding landing limbs, the landing limb performs backward motion. Though there are differences of levels between landing positions of limbs, the robot can continue to move by adjusting verical trajectory length of the limbs.

In this way, the robot walks on the uneven ground while keeping high stability margin. Figure 10 shows a scene where the robot moves over a bump in the tripod gait using six limbs. The sensor-based pose adjustment algorithm is implemented

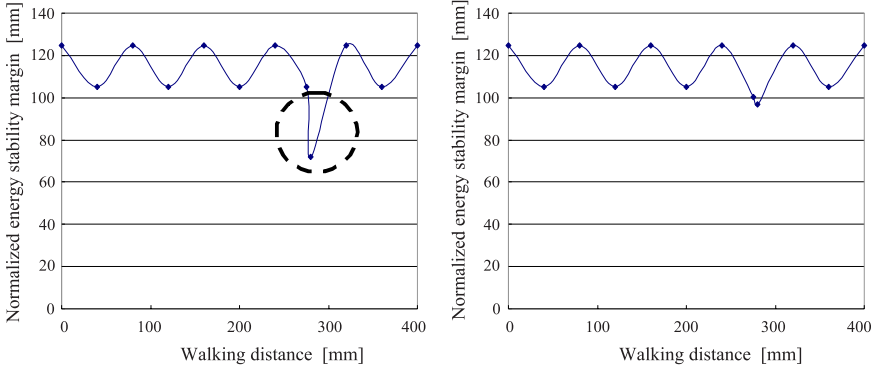


Fig. 11. Change of stability margin while walking on the uneven ground.

in the walking algorithm. We can see the pose of the body is kept horizontally by reducing inclination even a limb is on a bump.

The left and right figures in Fig. 11 show the changes of stability margin without and with the pose adjustment in walking, respectively. The horizontal axes indicate the walking distance of the robot. The vertical axes indicate stability margin. The circle in the left figure indicates the part where the limb lands on a bump and the stability margin decreases. In the right figure, the stability margin at the corresponding part does not decrease. From the figures, we can see the adjustment of the pose of the body makes stability margin almost constant even the robot moves over a bump.

5 Adjustment of Footholds of Supporting Limbs

To keep static stability in manipulation tasks, the robot has to adjust footholds of supporting limbs in accordance with the pose of the manipulation limbs. We discuss adjusting footholds of supporting limbs in the case that the two neighboring limbs are used as arms. This is the case that the mass center of the robot changes the most. The side of the manipulation limbs is called front or forward, temporarily. Figure 12 shows a limb mechanism adjusting footholds of supporting limbs; two neighboring limbs are lifted up in parallel in front of the robot for simulation of manipulation by two neighboring limbs. When the robot moves the two neighboring limbs up, the robot falls down if it does not adjust the footholds of supporting legs.

We examine the change of stability margin depending on the footholds of supporting limbs. In the examination, we fix the height of the body to $149[mm]$ horizontally by fixing the joints of supporting limbs. In that pose, the second joint is at 60 degrees downward and the third joint is at 0 degrees. Only the first joints of supporting limbs are rotated to change the footholds.

Figure 13 shows the change of stability margin when the two first joints of the front side supporting limbs are rotated from 0 degrees to 60 degrees in forward

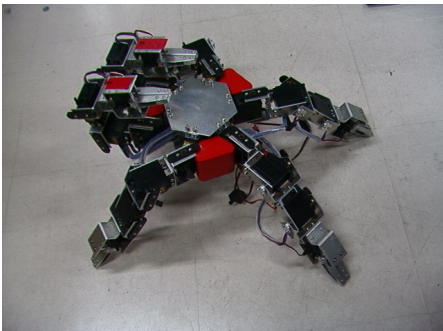


Fig. 12. A limb mechanism robot supported by four limbs while two neighboring limbs are lifted up.

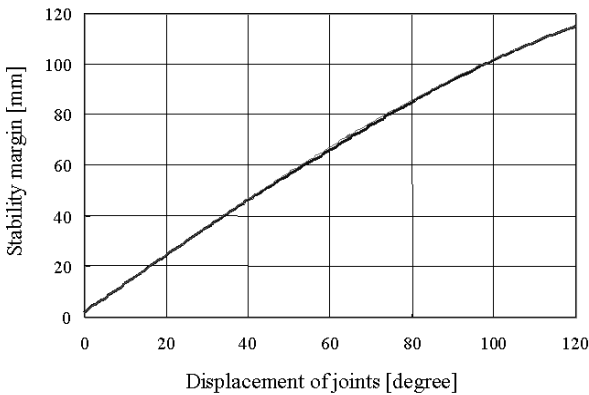


Fig. 13. Change of stability margin for changing footholds by rotating the first joints of the front side limbs.

direction, while the two neighboring limbs are lifted up in parallel. When the first joints are set at 0 degrees, the limbs are set at standard pose where the limbs are spread radially. The horizontal axis indicates the sum of the rotational angles of the two joints. The vertical axis indicates stability margin. As the limbs are moved forward, stability margin is increased monotonously. The maximum stability margin is obtained at the limit of the rotational angles 60 degrees. Thus, the two front side supporting limbs should be moved forward to obtain maximum stability margin before two neighboring limbs are moved up for manipulation.

Figure 14 shows the change of stability margin when the two neighboring limbs are being up in parallel, while the two first joints of the front side supporting limbs are at 60 degrees in forward direction. The two neighboring manipulation limbs are moved up by actuating second and third joints. The horizontal axis indicates the sum

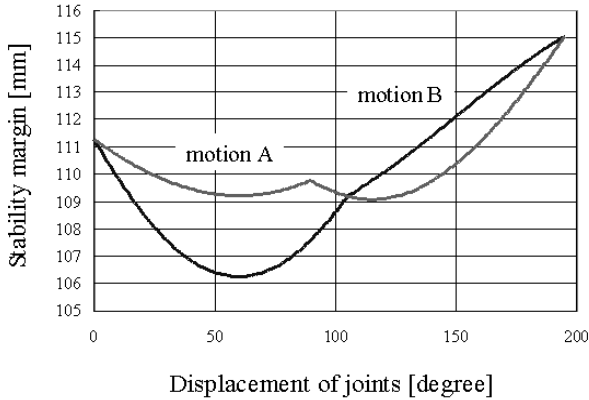


Fig. 14. Change of stability margin for moving two neighboring limbs after adjusting the footholds to maximize stability margin.

of the rotational angles of the two joints. The vertical axis indicates stability margin. We examine two motion pattern of moving limbs. In the first pattern, the third joint is actuated first and the second joint is actuated next (motion A in Fig.14). In the second pattern, the second joint is actuated first and the third joint is actuated next (motion B in Fig.14). The second joint is rotated upward from -60 degrees to 45 degrees. The third joint is rotated upward from 0 degrees to 90 degrees.

From Fig.14 we can see the high stability margin is hold for the two motion patterns. This is because the footholds of the front side supporting limbs are adjusted to increase stability margin, before moving two neighboring manipulation limbs.

6 Conclusions

In the paper, first we introduced a mobile robot with limb mechanism, and a basic trajectory generation for omnidirectional locomotion. Second, we describe a sensor-based walking method on rough terrains using an acceleration sensor attached to the body. The experimental results of walking on the uneven ground show the pose of the body is kept horizontally constant when a limb of the robot is on a bump in walking. Third, we describe adjustment of footholds of supporting limbs to keep high stability margin while two neighboring limbs are used as arm. The change of stability margin is examined in accordance with the change of the footholds.

Acknowledgement

This research was performed as a part of Special Project for Earthquake Disaster Mitigation in Urban Areas in cooperation with International Rescue System Institute (IRS) and National Research Institute for Earth Science and Disaster Prevention (NIED).

References

1. G.Pritschow, et.al., "Configurable Control System of a Mobile Robot for On-site Construction Masonry," Proc. of 10th Inter. Symposium on Robotics and Automation in Construction, pp.85–92, 1993.
2. E.Papadopoulos and S.Dubowsky, "On the Nature of Control Algorithms for Free-Floating Space Manipulators," IEEE Transaction on Robotics and Automation, vol.7, no.6, pp.759–770, 1991.
3. E.Nakano, et al., "First Approach to the Development of the Patient Care Robot," Proc. 11th Inter. Symposium on Industrial Robot, pp.87–94, 1981.
4. S.Skaar, et.al., "Nonholonomic Camera-Space Manipulation", IEEE Transaction on Robotics and Automation, vol.8, no.4, pp.464–479,1992.
5. Y.F.Zheng and Q.Yin, "Coordinating Multi-limbed Robot for Generating Large Cartesian Force," Proc. of IEEE Inter. Conf. on Robotics and Automation, pp.1653–1658,1990.
6. C.Su and Y.F. Zheng, "Task Decomposition for a Multi-limbed Robot to Work in Reachable But Unorientable Space," IEEE Transaction on Robotics and Automation, vol.7, no.6, pp.759–70, 1991.
7. S.Sugiyama, et.al., "Quadrupedal Locomotion Subsystem of Prototype Advanced Robot for Nuclear Power plant Facilities," Proc. of Fifth Inter. Conf. on Advanced Robotics, pp.326–333, 1991.
8. K.Hartikainen, et al., "Control and Software Structures of a Hydraulic Six-Legged Machine Designed for Locomotion in Natural Environments," Proc. of IEEE/RSJ Inter. Workshop on Intelligent Robots and Systems, pp.590–596, 1996.
9. N.Koyachi, et al., "Integrated Limb Mechanism of Manipulation and Locomotion For Dismantling Robot - Basic concept for control and mechanism - ," Proc. of the 1993 IEEE/Tsukuba Inter. Workshop on Advanced Robotics, pp.81–84, 1993.
10. N.Koyachi, et al., "Hexapod with Integrated Mechanism of Leg and Arm," Proc. of IEEE Inter. Conf. on Robotics and Automation, pp.1952–1957, 1995.
11. T.Arai, et al., "Integrated Arm and Leg Mechanism and its Kinematics Analysis," Proc. of IEEE Inter. Conf. on Robotics and Automation, pp.994–999,1995.
12. N.Koyachi, et al., "Design and Control of Hexapod with Integrated Limb Mechanism: MELMANTIS," Proc. of 1996 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems, pp.877–882, 1996.
13. J.Racz, et al., "MELMANTIS - the Walking Manipulator," Proc. of the 5th Inter. Symposium on Intelligent Robotic Systems, pp.23–29, 1997.
14. Y.Takahashi, et al., "Development of Multi-Limb Robot with Omnidirectional Manipulability and Mobility," Proc. of 2000 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems, pp.877–882,2000.
15. T.Arai, et al., "Omni-Directional Mobility of Limb Mechanism Robot," Proc. of 4th Inter. Conf. on Climbing and Walking Robots, pp.635–642,2001.
16. Y.Mae, et al., "Evaluation of omni-directional mobility of multi-legged robots based on error analysis of dead reckoning," Proc. of 5th Inter. Conf. on Climbing and Walking Robots, pp.271–278,2002.

Experiments in Learning Helicopter Control from a Pilot

Gregg Buskey^{1,2}, Jonathan Roberts¹, and Gordon Wyeth²

¹ CSIRO Manufacturing and Infrastructure Technology
P.O. Box 883
KENMORE 4069, Queensland, Australia
Gregg.Buskey@csiro.au

² School of Information Technology and Electrical Engineering, University of Queensland
ST LUCIA, Queensland, Australia

Abstract. This paper details the development of a machine learning system which uses the helicopter state and the actions of an instructing pilot to synthesise helicopter control modules online. Aggressive destabilisation/restabilisation sequences are used for training, such that a wide state-space envelope is covered during training. The performance of heading, roll, pitch, height and lateral velocity control learning is presented using our Xcell 60 experimental platform. The helicopter is demonstrated to be stabilised on all axes using the "learning from a pilot" technique. To our knowledge, this is the first time a "learning from a pilot" technique has been successfully applied to all axes.

1 Introduction

This paper describes the development of an online learning control system for an Xcell size 60 helicopter. Helicopter state, and the corresponding pilot signals for stabilisation are used as training examples for the learning system. In so doing, one can avoid the often tedious and/or mathematically complex process of analytically designing helicopter control systems. To our knowledge, this is the first time that learning from pilot demonstration has been successfully applied to all helicopter axes.

A wide range of classical and non-conventional control methodologies have been applied to scaled size helicopter control, including linear PD/PID [15,1,18], sliding mode control [6], input/output linearization [8,13], gain scheduling [16], fuzzy control [17,11,10,12], neural networks [5,7,13,10,9], markovian policy search [2], and combinations thereof. It should be emphasised that only a few of the non-conventional studies have moved beyond simulations.

While many of the neuro/fuzzy approaches incorporate some form of limited authority, online adaptation, the majority of control authority is learnt offline. This offline controller learning has been performed using actor/critic type reinforcement learning [7], policy search methods based on MDP models built from flight data [2], and back propagation of errors through neural forward models [9]. Only two methods have used pilot control signals to directly train a feedback controller [11,5]. While the Fuzzy Associative Memories (FAM) in [11] showed promise in simulation, the

controllers failed to stabilise the real helicopter. The Multi-layer Neural Networks (MLNN) in [5], on the other hand, managed to stabilise the real helicopter for brief periods, however later investigations showed that the system had many hidden, potential instabilities [3].



Fig. 1. CSIRO's Xcell 60 Helicopter Testbed

2 Approach Outline

In previous work we have used MLNN trained by pilot demonstration to learn the relationship between helicopter state and the corresponding stabilising control demands [5,3]. While the use of these networks has found some success, we have determined that their lack of reliability during training makes them unsuitable for online learning [3]. The method described in this paper uses specific destabilisation/restabilisation sequences to provide the controller with a wide experience envelope during learning. Controller training is performed during the restabilisation phase.

SISO heading (ψ), roll (ϕ) and pitch (θ) controllers, and MISO lateral velocity ($[v, \phi]$), and height ($[Z, w]$) controllers have been developed (Figure 2). Each module has authority over one control input. Lateral modules command the main rotor lateral cyclic pitch (δ_{lat}) which effectively induces a rate of roll, longitudinal modules command the longitudinal cyclic pitch (δ_{lon}) which induces a rate of pitch, the heading module commands the tail rotor pitch (δ_{tail}) which induces a yaw rate, and the height controller sends demands to the main rotor collective pitch which induces a vertical velocity. Rate feedback was not included in any of the loops as it was found in our earlier experiments [5] that the inclusion of rate feedback could introduce destabilising associations in the learning process [4].

3 Platform Description

This section describes the experimental helicopter platform. State variables include linear velocities $[u v w]$, angular velocities $[p q r]$, attitudes $[\phi \theta \psi]$ and position

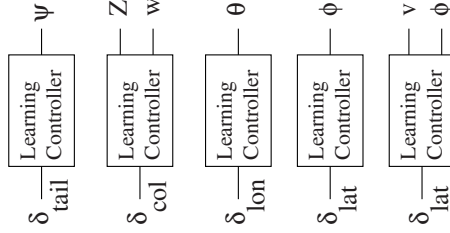


Fig. 2. Velocity control structures

$[X \ Y \ Z]$. Control signals include lateral cyclic δ_{lat} , longitudinal cyclic δ_{lon} , tail rotor collective δ_{tail} , and main rotor collective δ_{col} .

We have recently upgraded our platform from a JR Ergo size 60 model helicopter to an Xcell size 60 (Figure 1). Both have main rotor diameter of ~ 1.5 m, a height of 0.6m with modified landing gear and avionics housing [14], and weigh approximately 4kg without avionics. However, the Xcell has a payload capacity of approximately 10kg, as opposed to the 5kg capacity of the Ergo. This means that the aircraft (with avionics on board) is not constantly operating near its limits, and thus more stable flight performance is achieved. This enhanced flight quality has dramatically increased the performance of both our traditional control and learning control experiments. The new aircraft also brought a revision to our sensor mounting and isolation techniques. The isolation system has been specifically designed to reject rotor pass vibration, while introducing minimal phase lag such that the sensors still maintain a true representation of the aircraft's motion.

The sensors include stereo vision for height and lateral velocity estimation, a cheap, in-house designed IMU comprised of a 3 axis magnetometer, three accelerometers, and three gyros [14], and a low grade DGPS capable of 1Hz updates. Complimentary filtering is used to estimate roll, pitch and yaw from the accelerometers, gyros and magnetometers. Velocity estimation from vision uses the roll and pitch information from the IMU to disambiguate these influences from the optic flow. Further complimentary filtering is used to fuse these velocity estimates with the accelerometer information. No custom processing hardware is used for these estimates. Rather, all vision processing, control algorithms, and communications are maintained by a Celeron III 700MHz PC104 computer [14].

4 Learning Control Structure

Each state is partitioned into several overlapping fuzzy sets. If μ_{Ai} and μ_{Bj} describe the degree of membership of states A in set i and state B in set j , then the degree of membership C_{ij} in each FAM cell M_{ij} is defined as

$$C_{ij} = \mu_{Ai} \mu_{Bj}$$

The output of the FAM controller is defined as

$$o = \frac{\sum_i \sum_j C_{ij} y_{ij}}{\sum_i \sum_j C_{ij}}$$

where y_{ij} is the value associated with FAM cell M_{ij} . Unlike the previous FAM “teach by show” work in [11], we have chosen these cell values to be crisp δ control demands rather than fuzzy Mandini sets. In so doing, the step of carefully designing output sets to facilitate optimal control sensitivity is avoided.

The FAMs are trained using a simple, weighted average approach. For each input/output training example $[x_t, \delta_t]$, the principle FAM cell \bar{M}_{ijt} is found to be that which satisfies

$$\max C_{ij} \quad \forall i, j \mid x_t$$

The cell value running average for the principle cell \bar{y}_{ij} is adjusted according to the corresponding pilot demand δ_t , and the degree \bar{C}_{ijt} to which the current state x_t belongs to \bar{M}_{ijt} according to

$$\bar{y}_{ij} = \frac{\sum_t \delta_t \bar{C}_{ijt}}{\sum_t \bar{C}_{ijt}}$$

This yields an average over all training examples weighted according to their degree of membership within the principle FAM cell. The number of state excursions N into each FAM cell, before training is considered complete, is a design choice. While $N = 1$ is theoretically sufficient to find an exemplar for each cell, larger N results in a controller that represents the general actions of the teacher; i.e. any pilot mistakes will be absorbed by the averaging.

The fuzzy associative memories did not exhibit the sensitivity to pilot variation and error found in feedforward neural networks [4]. This is due to the averaging nature of the FAM controller. Secondly, the FAM cells form local approximators meaning that learning concentrated in one region of state space for a sustained period will not result in previous learning in other regions being lost; this is often a problem with the global approximation nature of feedforward neural networks.

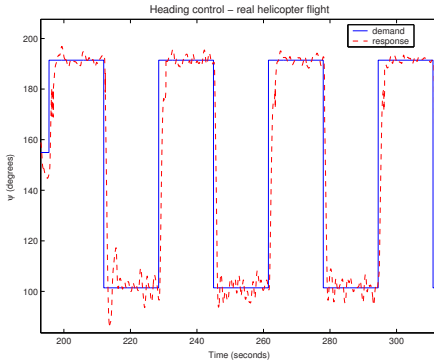
5 Learnt Controller Performance

This discussion details only the results of the learning approach on the experimental platform. All training and testing was performed in moderate wind conditions. A comprehensive discussion of the learning approach in simulation can be found in [4].

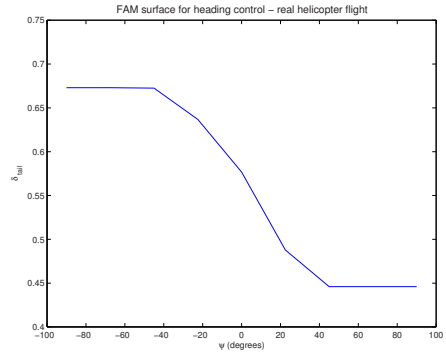
5.1 Heading Control

The learning sequence for heading control involves the pilot first choosing a heading as the stabilisation point. The pilot then takes the aircraft 90 degrees left or right of

this heading. The aircraft is then brought back to the stabilisation point, during which time the learning system trains. The pilot does not need to come back to the exact point, since the desired heading is taken as the average of that which the pilot stops at for several seconds. All heading errors during each sequence are then calculated retrospectively relative to this point. The controller training took only 30 seconds to complete. The resulting heading tracking performance is shown in Figure 3(a). The corresponding 1-D fuzzy surface is shown in Figure 3(b).



(a) Demand vs response



(b) Control surface

Fig. 3. Heading control

5.2 Roll Control

The learning sequence for roll control involves first leaning the aircraft left or right, after which the attitude is brought back to flat. During this flattening of the aircraft attitude, the control system is learning. As with the heading control learning, roll control learning took only 30 seconds. The roll tracking performance, and the corresponding 1-D control surface are shown in Figures 4(a) & 4(b) respectively.

5.3 Pitch Control

The learning sequence is much the same as that for roll control, except of course with motions about the pitch axis (y axis). The tracking performance of the pitch controller is shown in Figure 5. The pitch control surface looks much the same as the roll control surface, and is therefore not included. It is interesting to note however that the range of stick movement for pitch control was around half that for the roll control. During the tuning of our traditional loops we found that the best pitch gain was around $5/8$ of the roll gain.

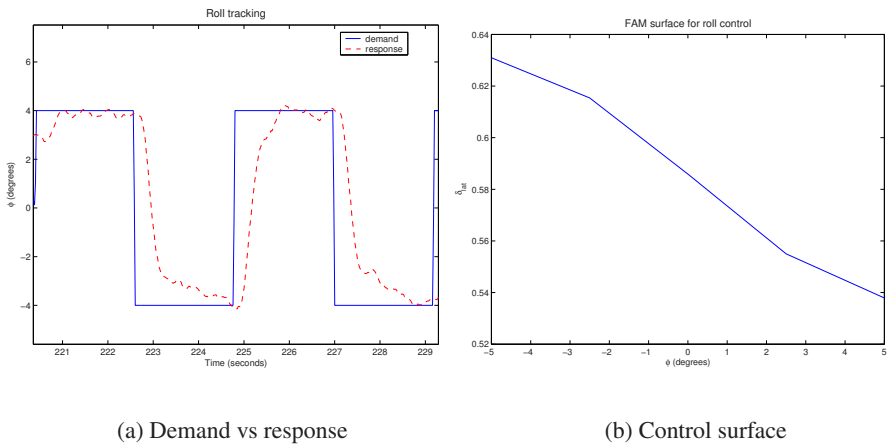


Fig. 4. Roll control

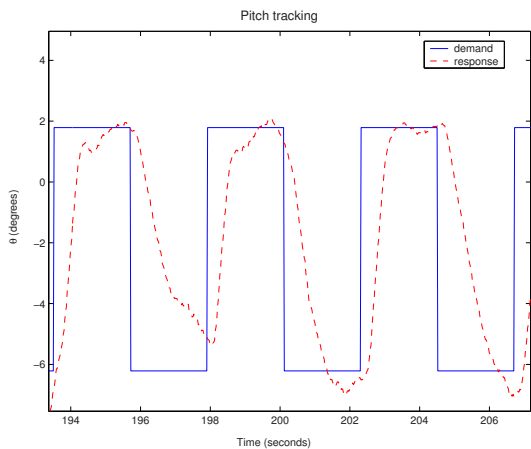
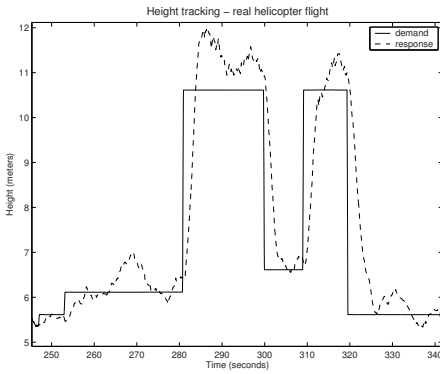


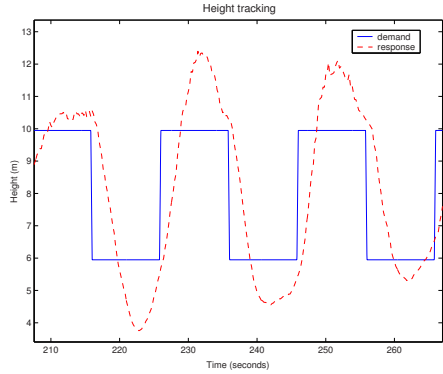
Fig. 5. Demand vs response for pitch control

5.4 Height Control

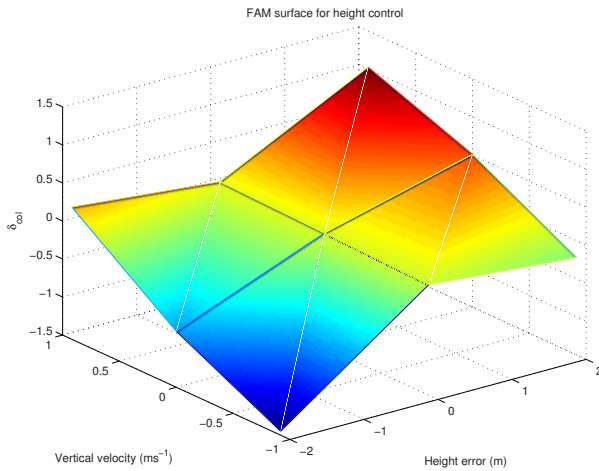
The learning sequence for height control involves the pilot accelerating upward, turning around and descending a couple of meters before abruptly stopping. This process is then mirrored. Since the height estimates are updated only at 5Hz, so to does our controller run at 5Hz. It was found that if during the training phase, the pilots actions were only sent to the servos at a 5 Hz update rate, the pilot's actions were larger. The pilot wasn't informed that this slow update would happen. The pilot complained that the control felt sticky, and that we may wish to check the swash plate movement on the main shaft. However, the pilot altered his control to



(a) Demand vs response - 5 Hz updates



(b) Demand vs response - continuous updates



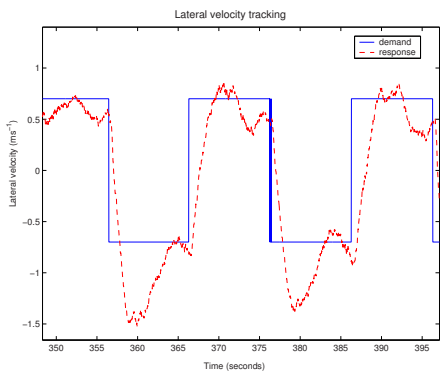
(c) Control surface - continuous updates

Fig. 6. Height control

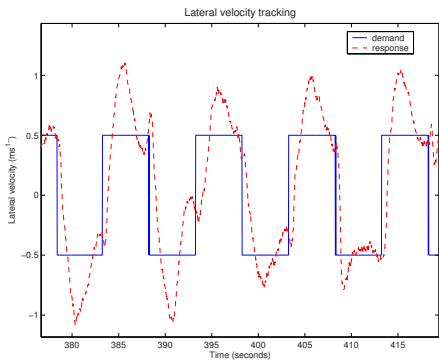
accommodate for this "sticky" behaviour. The corresponding height tracking for the controller generated with 5 Hz pilot updates and uninterrupted updates are shown in Figures 6(a) & 6(b) respectively. Note how the system trained with the 5Hz pilot updates produced less overshoot than that trained with the uninterrupted updates. The control surface generated with 5Hz updates is shown in Figure 6(c).

5.5 Lateral Velocity Control

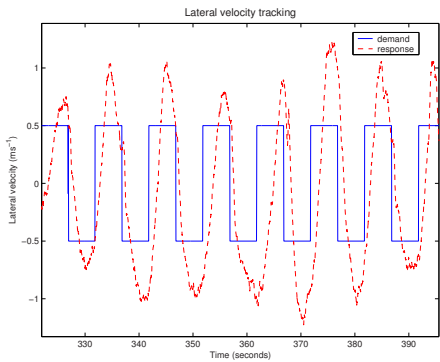
The learning sequence for lateral velocity control involves the pilot first inducing either a positive or negative velocity. The aircraft is then brought back to zero velocity, during which time the learning system is trained. The resulting control performance over different training instances is varied, however all seem to exhibit some considerable overshoot. Nevertheless, from Figures 7(a), 7(b) & 7(c) it can be clearly seen that the aircraft has been stabilised, even if the dynamic response indicates more than 70% overshoot, both biased (Figure 7(a)) & unbiased (Figure 7(b) & 7(c)). The control surface associated with the tracking in Figure 7(a) is shown in Figure 7(d).



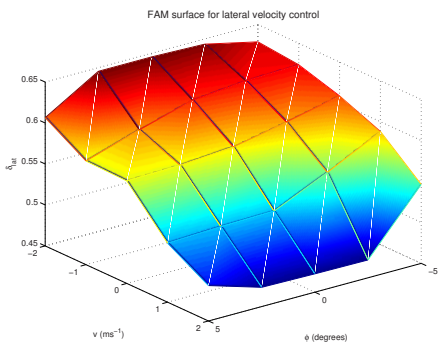
(a) Demand vs response - 1



(b) Demand vs response - 2



(c) Demand vs response - 3



(d) Control surface

Fig. 7. Lateral velocity control

The variation in performance quality is thought to be attributable to two possible factors. One possibility is that during training, the pilot is more aggressive than in other training instances, thus overshooting the zero velocity point. However, while overshoots have been seen in the training data, these have not been to the extent exhibited by the controllers. A second factor could be the partitioning of the state space into fuzzy regions. From the training data, we have noticed that during velocity stabilisation, the pilots actions are very sporadic, meaning that incorrect partitioning may result in the learning system missing these details. Both possibilities need to be examined in more detail in the future.

6 Conclusion

This paper details the progress thus far toward developing a machine learning system which learns to control a helicopter from a pilot's actions. Height control, heading control, roll control, pitch control, and lateral velocity control have all been effectively demonstrated on the real helicopter.

Work in the immediate future is to move the longitudinal motion learning from the simulation environment to the experimental platform. Furthermore, it is necessary to try and understand why overshoots and inconsistencies appear, particularly in the lateral velocity controller. Methods for reducing these overshoots and biases must then be considered, either through alteration of the control structure or the training process.

Acknowledgments

The authors would like to thank the whole automation team for their invaluable assistance and support. In particular, Peter Corke, Craig Worthington, Les Overs, Stuart Wolfe, Steven Brosnan, Pavan Sikka, Graeme Winstanley, Mathew Dunbabin, Elliot Duff, and our pilot Fred Proos.

References

1. O. Amidi. *An Autonomous Vision-Guided Helicopter*. PhD thesis, Dept of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, 1996.
2. J.A. Bagnell and J.G. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *International Conference on Robotics and Automation*, 2001.
3. G. Buskey, J. Roberts, P. Corke, P. Ridley, and G. Wyeth. Sensing and control for a small-size helicopter. In B. Siciliano and P. Dario, editors, *Experimental Robotics*, volume VIII, pages 476–487. Springer-Verlag, 2003.
4. G. Buskey, J. Roberts, and G. Wyeth. Learning autonomous helicopter control from a pilot. In *Proceedings of the International Conference on Advanced Robotics*, 2003.
5. G. Buskey, G. Wyeth, and J. Roberts. Autonomous helicopter hover using an artificial neural network. In *IEEE International Conference on Robotics and Automation*, pages 1635–1640, Seoul Korea, May 2001.

6. T. Chen and T.S. Li. Simplex-type fuzzy sliding-mode control. *Fuzzy Sets and Systems*, 124:249–261, 2001.
7. R. Enns and J. Si. Helicopter tracking control using direct neural dynamic programming. *Proceedings of the International Joint Conference on Neural Networks*, 2:1019–1024, July 2001.
8. T.J. Koo and S. Sastry. Outputtracking control design of a helicopter model based on approximate linearization. *IEEE Conference on Decision and Control*, 4:3635–3640, 1998.
9. K. KrishnaKumar and J. Neidhoefer. Immunised neurocontrol. *Expert Systems With Applications*, 4(3):201–214, 1997.
10. D. McLean and H. Matsuda. Helicopter station-keeping: comparing lqr, fuzzy-logic and neural-net controllers. *Engineering Applications of Artificial Intelligence*, 11:411–418, 1998.
11. J.F Montgomery and A. Bekey. Learning helicopter control through "teaching by showing". *IEEE Conference on Decision and Control*, 4:3647 – 3652, 1998.
12. C. Phillips, C.L. Karr, and G. Walker. Helicopter flight control with fuzzy logic and genetic algorithms. *Engineering Applications of Artificial Intelligence*, 9(2):175–184, 1996.
13. J.V.R. Prasad, A.J. Calise, Y. Pei, and J.E. Corban. Adaptive nonlinear controller synthesis and flight test evaluation. In *Proceedings of the 1999 IEEE International Conference on Control Applications*, pages 137–142, Kohala Coast-Island of Hawaii, August 1999.
14. J.M. Roberts, P.I. Corke, and G. Buskey. Low-cost flight control system for a small autonomous helicopter. In *Australasian Conference on Robotics and Automation*, Auckland, New Zealand, 2002.
15. C.P. Sanders. Hierarchical control of small autonomous helicopters. *IEEE Conference on Decision and Control*, 4:3629 – 3634, 1998.
16. K. Sprague, V. Gavrillets, D. Dugail, B. Mettler, and E. Feron. Design and applications of an avionics system for a miniature acrobatic helicopter. In *Digital Avionics Systems Conference*, pages 3.C.5–1 – 3.C.5–10, Daytona Beach Florida, October 2001.
17. M. Sugeno, H. Winston, and I. Hirano. Intelligent control of an unmanned helicopter based on fuzzy logic. In *51st Annual Forum of the American Helicopter Society*, Frt Worth, May 1995.
18. B. Woodley and H. Jones. A contestant in the 1997 international aerial robotics competition. aerospace robotics laboratory, stanford university. In *AUVSI*, July 1997.

Landing on a Moving Target Using an Autonomous Helicopter

Srikanth Saripalli¹ and Gaurav S. Sukhatme¹

Robotic Embedded Systems Laboratory
Center for Robotics and Embedded Systems
University of Southern California
Los Angeles, California, USA
{srik, gaurav}@robotics.usc.edu

Abstract. We present a vision-based algorithm designed to enable an autonomous helicopter to land on a moving target. The helicopter is required to identify the target, track it, and land on it while the target is in motion. We use Hu's moments of inertia for precise target recognition and a Kalman filter for target tracking. Based on the output of the tracker, a simple trajectory controller is implemented which (within the given constraints) ensures that the helicopter is able to land on the target. We present data collected from manual flights which validate our tracking algorithm.

1 Introduction

In recent years, considerable resources have been devoted to the design, development and operation of unmanned aerial vehicles. The applications of such unmanned aerial vehicles are diverse, ranging from scientific exploration and data collection, to provision of commercial services, military reconnaissance, and intelligence gathering [1]. Other areas include law enforcement, search and rescue, and even entertainment.

Unmanned aerial vehicles, particularly ones with vertical takeoff and landing capabilities (VTOL), enable difficult tasks without endangering the life of human pilots. This potentially results in cost and size savings as well as increased operational capabilities, performance limits, and stealthiness. Currently the capabilities of such unmanned aerial vehicles are limited. A helicopter is a compact VTOL capable platform; it is also extremely maneuverable. Autonomous helicopters equipped with the ability to land on moving targets would be very useful for various tasks as search and rescue, law enforcement, and military scenarios where micro air vehicles (MAVs) may want to land on a convoy of enemy trucks.

We have previously developed a system which was successful in landing a helicopter autonomously on a stationary target using vision and global positioning system [2]. In this paper we focus on the problem of tracking and landing on a moving target using an autonomous helicopter as a platform. We track a moving target using a downward looking camera mounted on a helicopter. Trajectory planning is performed to allow the helicopter to land on the target. Data from flight trials show

that our tracking system performs well. Based on the tracking information, our controller is able to generate appropriate control commands for the helicopter in order to land it on target.

2 Related Work

Classical visual target tracking has concentrated on object recognition using edge detection techniques followed by a velocity field estimator based on optical flow [3]. Edge detection may be bypassed altogether [4] using inter-temporal image deformation for heading prediction through feature matching. [5] discusses the merits of “good” feature selection and offers a motion computation model based on dissimilarity measures of weighted feature windows. The idea of feature selection and point matching [6] has been used to track human motion. In [7] eigenfaces have been used to track human faces. They use a principal component analysis approach to store a set of known patterns in a compact subspace representation of the image space, where the subspace is spanned by the eigenvectors of the training image set. In [8], a single robot tracked multiple targets using a particle filter for object tracking and Joint Probabilistic Data Association Filters were used for measurement assignment. [9] described a Variable Structure Interacting Multiple Model (VS-IMM) estimator combined with an assignment algorithm for tracking multiple ground targets. Although considerable research has been performed on tracking moving targets from both stationary as well as moving platforms for mobile autonomous systems [8], almost no work has been done on using autonomous helicopters for such a task. This paper focuses on tracking a moving target from the air and landing on it using an autonomous aerial vehicle. Several techniques have been implemented



Fig. 1. AVATAR (Autonomous Vehicle Aerial Tracking And Reconnaissance) after landing on a stationary helipad

for vision based landing of an autonomous helicopter on stationary targets. The problem of landing as such is inherently difficult because of the instability of the helicopter near the ground. In [10], a vision-based approach to landing is presented. Their landing pad had a unique shape which made the problem of identification of the landing pad much simpler. In [2], invariant moment descriptors were used for detection and landing on a stationary target.

3 Problem Formulation and Approach

We divide the problem of landing on a moving target into four stages. The first stage consists of detecting the target. We use vision for this purpose. We assume the target shape is known and no distractor targets are present. The second stage is tracking the target. We formulate the tracking problem as a Bayesian estimation problem and under linear system and Gaussian white noise assumptions solve it using a Kalman filter. The third stage is motion planning which plans a desired landing trajectory for the helicopter to land on the moving target. The fourth and last stage is control, which regulates the location of the helicopter over time, in accordance with the planner output.

The general problem is given a mechanical system (whose state is given by x) and initial and final conditions $x_0, x_f \in \mathfrak{X}$ where \mathfrak{X} is the state space of the mechanical system, we have to find a control signal $u : t \rightarrow u(t)$ such that at time t_f the system reaches x_f . The generalized problem is to find control inputs for a model helicopter for the entire range of a family of trajectories. Although such problems have been considered for general cases [11], to our knowledge, this is the first time that such a formalization is being applied to a combination of tracking a moving target and landing on it using an unmanned helicopter.

3.1 Assumptions

Real time target tracking for an autonomous mobile robot in a general dynamic environment is an unsolved problem. However with reasonable assumptions and constraints regarding the environment, the target tracking problem can be tractable. We make several assumptions which simplify the problem for our application but are not restrictive. These assumptions do not change the general nature of the problem but simplify the visual processing part.

The most difficult aspect of tracking is the discrimination of the target from the background. We assume that the shape of the target is known (we use a helipad with the letter H painted on it). Also we ensure that there is a high contrast between the target and the background. It is assumed that the target moves slowly enough for the helicopter to track it. We also assume that the target is not evasive or malicious. The target is smaller than the size of the image plane, in relative terms. Thus the target feature points do not span an area larger than the pixel dimensions of the focal plane. We assume that the helicopter is oriented such that the focal plane of the camera mounted on it is parallel to the target plane and this attitude is maintained throughout the tracking and the landing process. This ensures that we do not have to consider image skew.

4 Target Detection Using Vision

The target detection algorithm is described below in three parts; preprocessing, geometric invariant extraction, object recognition and state estimation. The goal of the

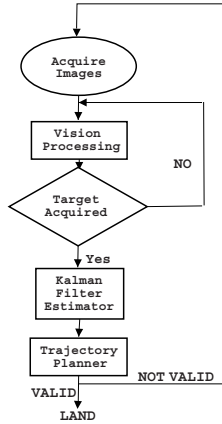


Fig. 2. Flow chart describing the algorithm used for landing on a moving target

preprocessing stage is to locate and extract the target, which is done by thresholding the intensity image and then segmenting and performing a connected component labeling of the image. The image is converted to gray scale and a 7×7 *Median-filter* is applied to the thresholded image for removing noise and to preserve the edge details effectively. The image is scanned row-wise until the first pixel at a boundary is hit. All the pixels which belong to the 8-neighborhood of the current pixel are marked as belonging to the current object. This operation is continued recursively until all pixels belonging to the object are counted. A product of this process is the area of the particular object in pixels. Objects whose area is less than a particular threshold area (≤ 80 pixels) are discarded. Similarly objects whose area is ≥ 700 pixels are discarded. The remaining objects are the regions of interest and are candidates for the target.

The next stage is to extract the target from the image which is performed by using Hu's moments of inertia [12]. These are normalized set of moments of inertia which are invariant to rotation, translation and scaling. The reader is referred to [2] for a complete discussion of the Hu's moments of inertia and their application for landing using an autonomous helicopter on a stationary target. After the extraction of the target from the image, the coordinates of the target are transformed into helicopter coordinates.

The coordinates of the target so obtained are in the image coordinate frame. These coordinates are transformed into state estimates relative to the helicopter, based on the height of the helicopter above the ground. The height of the helicopter is obtained from the on-board differential GPS. The x-coordinate of the target in the helicopter frame of reference is given by

$$x_{heli} = \frac{\text{height} \times \tan \frac{\phi_h}{2} \times x_{image}}{\text{camera res along the x axis}} \quad (1)$$

where ϕ_h is the field of view of the camera in the x direction, x_{image} is the x-coordinate of the target in the image plane. Similarly the y-coordinate of the target



Fig. 3. An typical image obtained from the downward pointing camera on the helicopter during one of the trials.

is given by

$$y_{heli} = \frac{height \times \tan \frac{\phi_v}{2} \times y_{image}}{\text{camera res along the } y \text{ axis}} \quad (2)$$

Since the helicopter is near hover at all instances we assume that the roll, pitch and yaw angles of the helicopter are negligible for computing the world coordinates of the target with respect to the helicopter.

5 Target Tracking Using a Kalman Filter

The output of the target detection (i.e. the measured x and y coordinates of the target on the ground) is the input to the tracker. Based on a second order kinematic model for the tracked object we can model the equation of the target as a linear system described by

$$X_k = \Phi_{k-1} X_{k-1} + w_k \quad (3)$$

where w_k is the random process noise and the subscripts on the vectors represent the time step. X_k is the state vector describing the motion of the target (its position p , velocity v and acceleration a). The measurement vector at time k is given by

$$Z_k = H_k X_k + u_k \quad (4)$$

where H_k is known and u_k is a random measurement noise. The Kalman filter is formulated as follows. Suppose we assume that the process noise w_k is white, zero-mean, Gaussian noise with a covariance matrix Q . Further assume that the measurement noise is white, zero-mean, Gaussian noise with a covariance matrix R , and that it is not correlated with the process noise. The system dynamics are given by

$$\begin{bmatrix} p_k \\ v_k \\ a_k \end{bmatrix} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{k-1} \\ v_{k-1} \\ a_{k-1} \end{bmatrix} + w_k$$

where a_{k-1} is a random time-varying acceleration and T is the time between the steps k and $k-1$. The state propagation and update equation for the discrete Kalman filter are given by

$$\hat{X}_k(-) = \Phi_{k-1} \hat{X}_{k-1}(-) \quad (5)$$

$$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^T + Q_{k-1} \quad (6)$$

$$S_k = H_k P_k(-) H_k^T + R \quad (7)$$

$$K_k = P_k(-) H_k^T S_k^{-1} \quad (8)$$

$$P_k(+) = (I_n - K_k H_k) P_k(-) \quad (9)$$

$$\hat{X}_k(+) = \hat{X}_k(-) + K_k (Z_k - H_k \hat{X}_k(-)) \quad (10)$$

In the above equations, the superscript T indicates matrix transposition. S is the covariance of the innovation, K is the gain matrix, and P is the covariance of the prediction error. Also we distinguish between estimates made before and after the measurements occur; $\hat{X}_k(-)$ is the state estimate that results from the propagation equations alone (i.e., before the measurements are considered) and $\hat{X}_k(+)$ is the corrected state estimate that accounts for measurements. $P_k(-)$ and $P_k(+)$ are defined similarly.

6 Trajectory Planning

We approximate the desired trajectory to be followed by the helicopter for tracking and landing by a cubic polynomial where the altitude z varies with time t as follows:

$$z(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 \quad (11)$$

with the following conditions

$$z(0) = z_o \quad z(t_f) = z_f \quad \dot{z}(0) = 0 \quad \dot{z}(t_f) = 0$$

where t_f is the final time. We restrict the class of trajectories by imposing these additional set of constraints:

$$\begin{aligned} \dot{z} &\leq 1m/s \quad x_h(0) = 0 \quad \dot{x}_h(t_f) = \dot{x}_{target}(t_f) \\ x_h(t_f) &= x_{target}(t_f) \end{aligned}$$

The above constraints provide a lower bound on the time of flight i.e, the time of flight for the helicopter can never be less than t_{min} where t_{min} is given by

$$t_{min} \geq \frac{-4a_2 + \sqrt{4a_2^2 - 12a_3a_1}}{6a_3} \quad (12)$$

If the helicopter has to intercept and land on a target at x meters from the origin. If the target is moving with the same speed as the helicopter, then the target should at least take t_{min} seconds to reach x for the helicopter to actually land on it. Hence the maximum velocity of the target is bounded to $\frac{x}{t_{min}}$. Here we assume that the helicopter is directly above the target at $t = 0$.

7 Behavior-Based Control

Our helicopter is controlled using a hierarchical behavior-based control architecture. The autonomous control has been discussed in detail in [13]. We used this controller for autonomous landing on a stationary target - this was discussed in detail in [2]. The reader is referred to [2] for the control and landing phase details. Here we recapitulate the controller used for landing briefly. The behavior-based control architecture used for the AVATAR is shown in Figure 4.

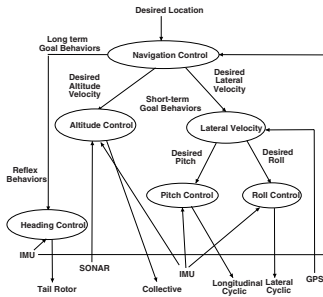


Fig. 4. AVATAR Behavior-Based Controller

The *altitude control* behavior is further split into three sub-behaviors, *hover control*, *velocity control* and *laser control*. The *hover control* sub-behavior is activated when the helicopter is either flying to a goal or is hovering over the target. This sub-behavior is used during the object recognition and object tracking state when the helicopter should move laterally at a constant altitude. The hover controller is implemented as a proportional controller. It reads the desired GPS location and the current location and calculates the collective command to the helicopter.

Once the target has been tracked and a trajectory planned the *velocity control* sub-behavior runs in parallel with the *laser control* sub-behavior. It is implemented as a PI controller. An integral term is added to reduce the steady state error. The helicopter starts to descend (and move laterally) till touchdown. The *laser control* sub-behavior works in conjunction with the *velocity control* sub-behavior. It maintains the helicopter at a commanded altitude as commanded by the trajectory planner until touchdown. The *laser control* sub-behavior is also implemented as a PI controller.

8 Experimental Results

In order to validate our algorithm, we performed experiments with the USC AVATAR shown in Figure 1. The helicopter was hovered manually at a height of 15 meters. The pilot controlled the helicopter to be in the hover position all the time, while the target (which was a ground robot in our experiments) was commanded to move in a straight line. The image data were processed offline. The goal position for the helicopter was 12 meters from the starting point, i.e, the helicopter should track the helipad for 12 meters and then land on it.

Figure 3 shows a typical image obtained from the downward pointing camera on the helicopter. The target is marked with an H and moves in the x-direction for a distance of 12m. Figure 5 shows three curves. The solid line is ground truth. It is the location of the target as measured on the ground by the odometry of the target robot. The dotted line is the location as measured by the camera without filtering. This is obtained by first detecting the target in the image, finding its image coordinates and then transforming them to the helicopter's reference plane as shown in Section 4. The dashed line is the location of the target deduced by the Kalman filter as explained in Section 5. Figure 6 shows the simulated trajectory followed by

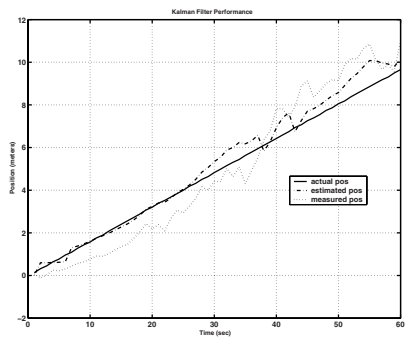


Fig. 5. Kalman filter tracking performance

the helicopter while tracking the target. The plot shows the trajectory of the target (solid) and the trajectory of the helicopter (dashed). As can be seen the helicopter is able to track the target quite well. Figure 7 shows the height of the helicopter

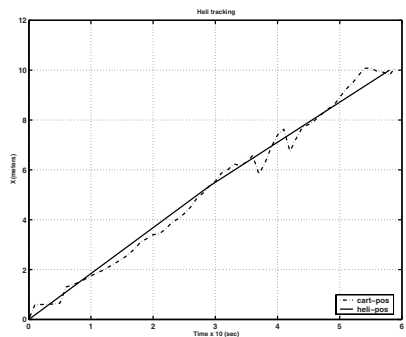


Fig. 6. The helicopter trajectory in the x-direction (dashed) and the target trajectory in the x-direction (solid) while tracking

with respect to time. During simulation it was found that updating the trajectory of the helicopter every time step the Kalman filter made a new prediction was not necessary. Only a discrete number of modifications were made to the trajectory and a cubic spline trajectory as described in Section 6 was used. It may be noted that

during the actual flight tests we will probably use a straight line interpolation of the cubic spline which is also shown in Figure 7

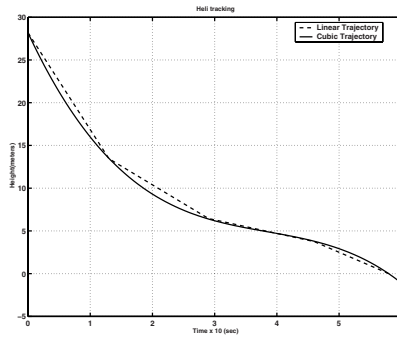


Fig. 7. Height trajectory commanded for tracking and landing

9 Conclusions

This paper describes the design of an algorithm for landing on a moving target using an autonomous helicopter. We use a Kalman filter as an estimator to predict the position of the target and plan the trajectory of the helicopter such that it can land on it. Given conditions about the distance at which the helicopter should land on the target we perform discrete updates of the trajectory so that we can track and land on the target. We have performed experiments in manual control mode where a human pilot was controlling the helicopter and was holding it in hover while we collected data of a ground target moving in the field of view of the helicopter. The estimator and the trajectory planner were run offline to test the validity of the algorithm. Figures 5, 6, 7 show the results obtained by using the Kalman filter in conjunction with a object recognition algorithm.

9.1 Limitations, Discussion and Future Work

In the future we plan to test the algorithm on our autonomous helicopter. Several limitations exist with the current algorithm:

- We assume that the helicopter is in hover (zero roll, pitch and yaw values and zero movement in the northing and easting directions). This is almost impossible to achieve in an aerial vehicle. We plan to integrate the errors in GPS coordinates and attitude estimates into the Kalman filter so that we are able to track the target more precisely.
- The current estimator is able to track the target only in a single dimension. We will extend it so that it is able to track the full pose of the target and verify the validity of the algorithm.

- Currently we use an algorithm based on the intensity of the image for object detection. We also assume that the object is planar. This is quite restrictive in nature. We plan to integrate our algorithm with a better object detector and a camera calibration routine so that the coordinates of the tracked object obtained during the vision processing stage are much more accurate.
- We will extend the algorithm in the future such that we are able to track multiple targets and then land on any one of them. Also we intend to investigate algorithms to be able to pursue and land on evasive targets.

Acknowledgment

This work is supported in part by NASA under JPL/Caltech contract 1231521, by DARPA under grant DABT63-99-1-0015 as part of the Mobile Autonomous Robot Software (MARS) program, and by ONR grant N00014-00-1-0638 under the DURIP program. We thank Doug Wilson for support with flight trials.

References

1. Office of the Under Secretary of Defense, "Unmanned aerial vehicles annual report," Tech. Rep., Defense Airborne Reconnaissance Office, Pentagon, Washington DC, July 1998.
2. S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually-guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Autonomous Systems*, 2003(to appear).
3. B. K. P. Horn and B. G. Sunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
4. C. Tomasi and J. Shi, "Direction of heading from image deformations," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 1993, pp. 4326–4333.
5. J. Shi and C. Tomasi, "Good frames to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 4326–4333.
6. S. Birchfield, "An elliptical head tracker," in *31 Asilomar Conference on Signals Systems and Computers*, November 1997, pp. 4326–4333.
7. M. Turk and A. Pentland, "Eigenfaces for recognition," in *Journal of Cognitive Neuroscience*, 1995, vol. 3, pp. 4326–4333.
8. D. Schultz, W. Burgard, D. Fox, and A. B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *IEEE International Conference on Robotics and Automation*, 2001, pp. 1165–1170.
9. Y. Bar-Shalom and W. D. Blair, *Multitarget-Multisensor Tracking: Applications and Advances*, vol. 3, Artech House, 1992.
10. O. Shakhmurov, Y. Ma, T. J. Koo, and S. S. Sastry, "Landing an unmanned air vehicle: vision based motion estimation and non-linear control," in *Asian Journal of Control*, September 1999, vol. 1, pp. 128–145.
11. K. Ogata, *Modern Control Engineering*, Prentice Hall, 2nd ed., 1990.
12. M. K. Hu, "Visual pattern recognition by moment invariants," in *IRE Transactions on Information Theory*, 1962, vol. IT-8, pp. 179–187.
13. J. F. Montgomery, *Learning Helicopter Control through 'Teaching by Showing'*, Ph.D. thesis, University of Southern California, May 1999.

Scan Alignment and 3-D Surface Modeling with a Helicopter Platform

Sebastian Thrun¹, Mark Diel², and Dirk Hähnel³

¹ Stanford University, Stanford, CA, USA

² WhirledAir, Inc.

³ University of Freiburg, Germany

Abstract. This paper describes initial results for a laser-based aerial mapping system. Our approach applies a real-time laser scan matching algorithm to 2-D range data acquired by a remotely controlled helicopter. Results obtain for urban and natural terrain exhibit an unprecedented level of spatial detail in the resulting 3-D maps.

1 Introduction

In recent years, a number of research teams have developed robotic systems for mapping indoor [10,19] and outdoor environments [8]. Since such models are usually confined to the immediate vicinity of the vehicle, active sensors such as sonars and laser range finders have become the technology of choice [18]—albeit some notable exception using passive cameras [14]. For the problem of acquiring accurate maps of outdoor terrain, ground vehicles are limited in two aspects: First, the ground has to be traversable by the vehicle itself. Many environments are cluttered with obstacles that are difficult to negotiate. Second, all important features in the environment have to be visible from relatively low vantage points. Moreover, the set of vantage points that can be attained usually lie on an approximate 2-D manifold parallel to the ground surface, since most ground vehicle cannot vary the height of their sensors. This is a severe limitation, that is particularly troublesome in complex, natural terrain.

In complimentary research, there exists a huge body of literature on high aerial and satellite-based mapping (see e.g., [2,5]). At high altitude, it is usually impossible to deploy active range sensors; instead, these techniques are commonly based on passive computer vision systems. While traversability is not an issue for high aerial vehicles, the relatively high vantage points makes it impossible to map vertical structures, and it limits the resolution at which maps can be acquired. Furthermore, clouds can cause obstruction of cast shadows in the imagery. And while air vehicles can change altitude and are therefore not subject to the 2-D manifold constraint characteristic of ground vehicles, such changes have next-to-zero effect on the visual appearance of the surface structure.

Low-flying air vehicles, such as helicopters, promise to overcome these limitations: they are much less constraint than ground vehicles with regards to their navigational capabilities, yet they can fly low enough to for acquiring data from vertical structures with high resolution. In particular, helicopters can be equipped with active range sensors. A seminal system by Miller et al. [13,12] has demonstrated



Fig. 1. Instrumented helicopter platform: The system is based on the Bergen Industrial Twin, with a modified SICK LMS laser range finder, a Crossbow IMU, a Honeywell 3-D compass, a Garmin GPS, and a Nikon camera. The system is equipped with onboard data collection and processing capabilities and a wireless digital link to the ground station.



Fig. 2. Some of the electronics onboard the helicopter: An Intel Stayton board with a 400Mhz XScale processor interfaces to the SICK LMS laser via a high speed RS422 serial link, and to all other devices (compass, GPS, IMU) via RS232. The communication to the ground is established via a 802.11b wireless link.

the feasibility of acquiring high-resolution ground models using active laser range sensors on a low-flying helicopter platform.

This paper describes a similar system for acquiring high-resolution 3-D models of urban and ground structures. The system, shown in Figure 1, is based on a Bergen Industrial Twin helicopter, equipped with a 2-D SICK range finder and a suite of other sensors for position estimation. Figure 2 shows some of the computer equipment mounted on the vehicle. The 2-D range finder provides the vehicle with 2-D range slices at a rate of 75Hz, oriented roughly perpendicular to the robot's flight direction. The helicopter is flown under manual control.

Building 3-D maps with centimeter resolution is difficult primarily of two reasons:

1. Using GPS and other proprioceptive sensors, the location and of the sensor platform can only be determined up to several centimeters accuracy. Similar limitations apply to the estimation of its angular orientation. The ground position

error induced by those angular errors are often in the order of several meters. This is illustrated by the ground model shown in Figure 3. Shown there is a 3-D model reconstructed using the vehicle's best estimates of its position and orientation. These limitations mandate the use of the range finder as a means for further reducing the errors in the vehicle pose estimate, and to jointly solve the environment mapping the vehicle localization problem.

2. The rich literature on *simultaneous localization and mapping*, or SLAM [6,10], has developed extensive techniques for simultaneously estimating the structure of the environment (3-D model, or map) and the pose of the vehicle. Range scan alignment techniques (applied in 2-D in robotics[11] and in 3-D in object modeling [16]) overcome this problem by cross-registering multiple scans. To do so, these techniques rely on multiple sightings of the same environmental feature. This is not the case for a 2-D sensor that is moved through the environment in a direction perpendicular to its own perceptive field: Here consecutive measurements always correspond to different things in the world.

We have developed a probabilistic SLAM algorithm that addresses both of these problems. Our approach acquires 3-D models from 2-D scan data, GPS, and compass measurements. The algorithm exploits a local smoothness assumption for the surface that is being modeled, but simultaneously allows for the possibility of large discontinuities. By doing so, it can utilize range scans for vehicle localization, and thereby improve both the pose estimate of the helicopter and the accuracy of the resulting 3-D environment model.

We believe that the maps acquired by our system are significantly more accurate and spatially consistent than previous maps acquired by helicopter systems [13,12]. A key reason of this increase in accuracy comes from the fact that scans are used for the pose estimate of the vehicle's sensor platform.

2 3-D Modeling Approach

2.1 Vehicle Model

Let x_t denote the *pose* of the sensor's local coordinate system at time t , relative to a global coordinate system of the 3-D model. This pose is specified by the three Cartesian coordinates and the three Euler angles (roll, pitch, yaw). In irregular intervals, we receive GPS and compass measurements for the pose, denoted by y_t . The probability of measuring y_t if the correct pose is x_t is Gaussian and denoted $p(y_t | x_t)$:

$$p(y_t | x_t) \propto \exp -\frac{1}{2} (y_t - x_t)^T A^{-1} (y_t - x_t) \quad (1)$$

Here A is the measurement covariance. Since all of these sensors are subject to *systematic error* (e.g., drift), we also employ a differential model $p(\Delta y_t | x_t, x_{t-1})$ where $\Delta y_t = y_t - y_{t-1}$ is the differential measurement (angles are truncated in this subtraction).

$$p(\Delta y_t | x_t, x_{t-1}) \propto \exp -\frac{1}{2} (\Delta y_t - \delta(x_t, x_{t-1}))^T D^{-1} (\Delta y_t - \delta(x_t, x_{t-1})) \quad (2)$$

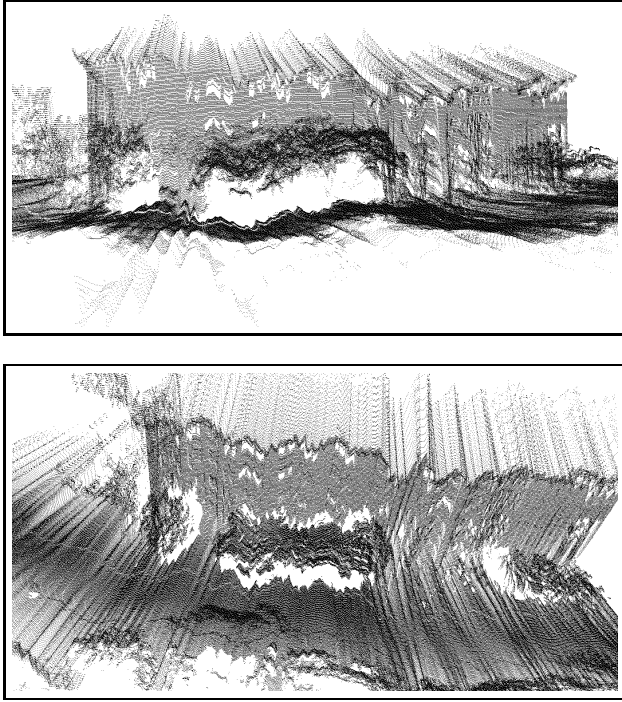


Fig. 3. Raw data of a multi-storey building

Here δ calculates the pose difference. The matrix D is the covariance of the differential measurement noise, whose determinant is much smaller than that of A . This model is implemented by a much narrower Gaussian, accounting for the fact that relative information is more accurate than absolute information. However, measurements y_t alone are insufficient for ground mapping as discussed above.

2.2 Range Sensor Model

To localize the sensor based on range data, we need a model of the range sensor. Most SLAM algorithms model the probability $p(z_t \mid m, x_t)$ of a measurement z_t , given the map m and the pose x_t . Such a generative model is the most general approach to robotic mapping; however, it involves as many variables as there are features in the map m ; thus the resulting likelihood functions would be difficult to optimize in real-time.

For a forward-flying helicopter which never traverses the same location twice, it is sufficient to model a relative probability of subsequent measurements conditioned on the pose: $p(z_t \mid x_t, x_{t-1}, z_{t-1})$. This probability models the spatial consistency of scan z_t relative to the previous scans, z_{t-1} , assuming that those scans are taken at the global poses x_t and x_{t-1} , respectively.



Fig. 4. Helicopter flying by a building under manual remote control. The image also shows the pilot.

In our implementation, this model is defined as follows:

$$p(z_t \mid x_t, x_{t-1}, z_{t-1}) \propto \prod_i \exp -\frac{1}{2} \min \left[\alpha, \min_j \left\{ (z_t^i - f(z_{t-1}^j, x_{t-1}, x_t))^T B^{-1} (z_t^i - f(z_{t-1}^j, x_{t-1}, x_t)) \right\} \right] \quad (3)$$

The index i iterates over individual beams in the scan z_t , and the function f projects measurements in scan j into the local coordinate system of scan i . The argument of the exponential involves two minimizations. The inner minimization identifies the index j of the measurement nearest to z_t^i , when projected into the local coordinate system of the scan z_t . The second minimization upper bounds this distance by α . It is best thought of as an outlier detection mechanism.

The measurement covariance B is degenerate: it possesses infinite covariance in the direction of helicopter flight. This degeneracy accounts for the fact that subsequent scans carry no information about the forward motion of the vehicle. The degeneracy implies that the rank of B^{-1} is five, even though the matrix is six-dimensional.

2.3 Optimization

The resulting probabilistic model is proportional to the product

$$p(y_t \mid x_t) p(\Delta y_t \mid x_t, x_{t-1}) p(z_t \mid x_t, x_{t-1}, z_{t-1}) \quad (4)$$

The negative log-likelihood is now given by the following expression:

$$\text{const.} + \frac{1}{2} \left((y_t - x_t)^T A^{-1} (y_t - x_t) \right)$$

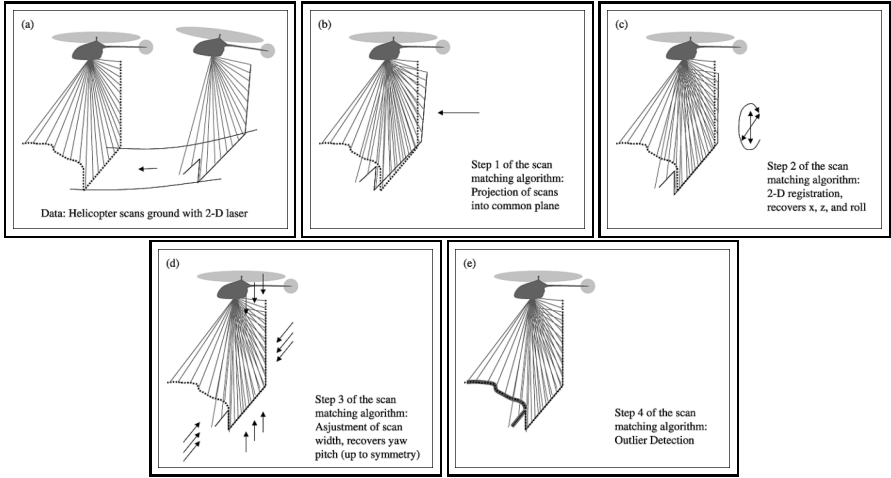


Fig. 5. Illustration of the Helicopter Mapping Process. From top to bottom: .

$$\begin{aligned}
 & + (\Delta y_t - \delta(x_t, x_{t-1}))^T D^{-1} (\Delta y_t - \delta(x_t, x_{t-1})) \\
 & + \sum_i \min \left[\alpha, \min_j \left\{ (z_t^i - f(z_{t-1}^j, x_{t-1}, x_t))^T B^{-1} (z_t^i - f(z_{t-1}^j, x_{t-1}, x_t)) \right\} \right] \Big)
 \end{aligned} \tag{5}$$

The map and the pose is recovered by minimizing this negative log-likelihood. We note that this problem combines a continuous optimization problem that is quadratic in non-linear functions of the pose x_t , with a discrete one that involves finding the corresponding z_{t-1}^j for each measurement beam z_t^i . Following mainstream optimization in the scan matching literature [3], our approach iterates a step in which the minimization is solved for a fix pose x_t , which a step in which the optimal pose is determined for a fixed setting of the minimizing indices. Both steps can be carried out highly efficiently [9]. Figure 5 illustrates this process.

The result is an algorithm that implements the optimization in an incremental fashion, in which the pose at time t is calculated from the pose at time $t - 1$ under incorporation of all scan measurements. While such an implementation is subject to local minima, it can be performed in real-time and works well, as long as the helicopter never traverses the same area twice. The 3-D model is also obtained in real-time, by using the corrected pose estimates to project measurements into 3-D space. The model may simply be represented by a collection of scan points in 3-D, or a list of polygons defined through sets of nearby scan points. Both are computed in real-time.

3 Results

We have tested our approach in a number of different environments, all of which involved significant vertical structure that cannot easily be mapped by high-aerial

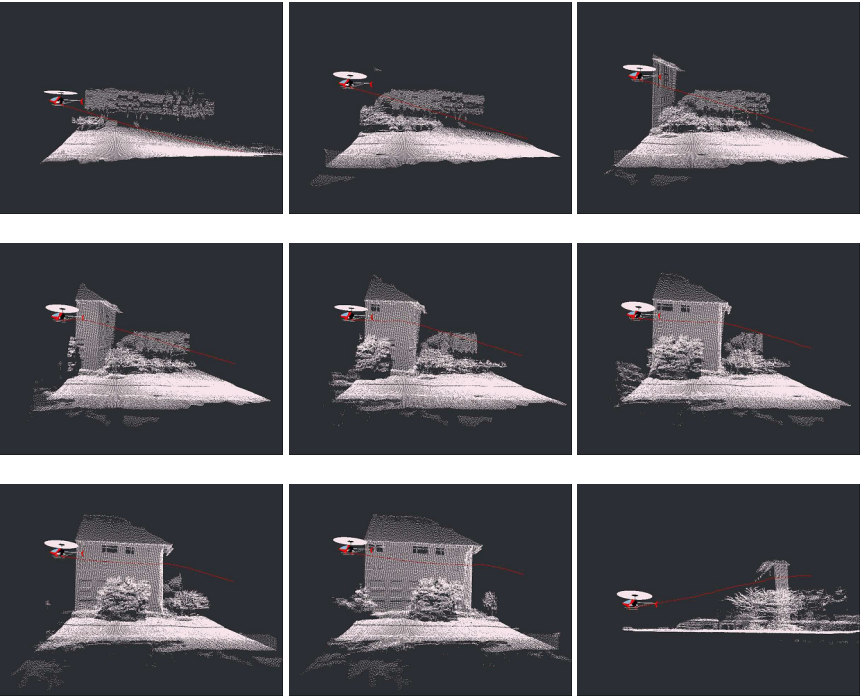


Fig. 6. Visualization of the mapping process, carried out in real-time. This figure shows a sequence of snapshots taken from the interactive ground display, which displays the most recent scans with less than 0.1 seconds latency.

vehicles. Figure 4 depicts the helicopter flying by a multi-storey building under manual control; it also depicts the pilot walking behind the vehicle. The raw data acquired in this flight is shown in Figure 3; this plot uses the helicopter's best estimate of its own pose for generating the map. These plots clearly show significant error, caused by a lack of accurate pose estimation.

Figure 6 depicts a sequence of maps as they are being generated in real-time using our approach. The latency between the data acquisition and the display on the ground is less than a tenth of a second. Snapshots of the final 3-D map are shown in Figure 7.

Figure 8 shows two maps acquired at a different urban site (left image) and at an ocean cliff (right image). These maps are represented by polygons that are defined over nearby points. The search for such polygons is also carried out in real-time, although rendering the final model requires substantially more time than rendering the multi-point model.

Figure 9 shows results obtained in a recent set of experiments involving autonomous flight, using a different helicopter (not shown here). The flight controller has been developed by others [15]. The top map depicts a parking lot, whereas the

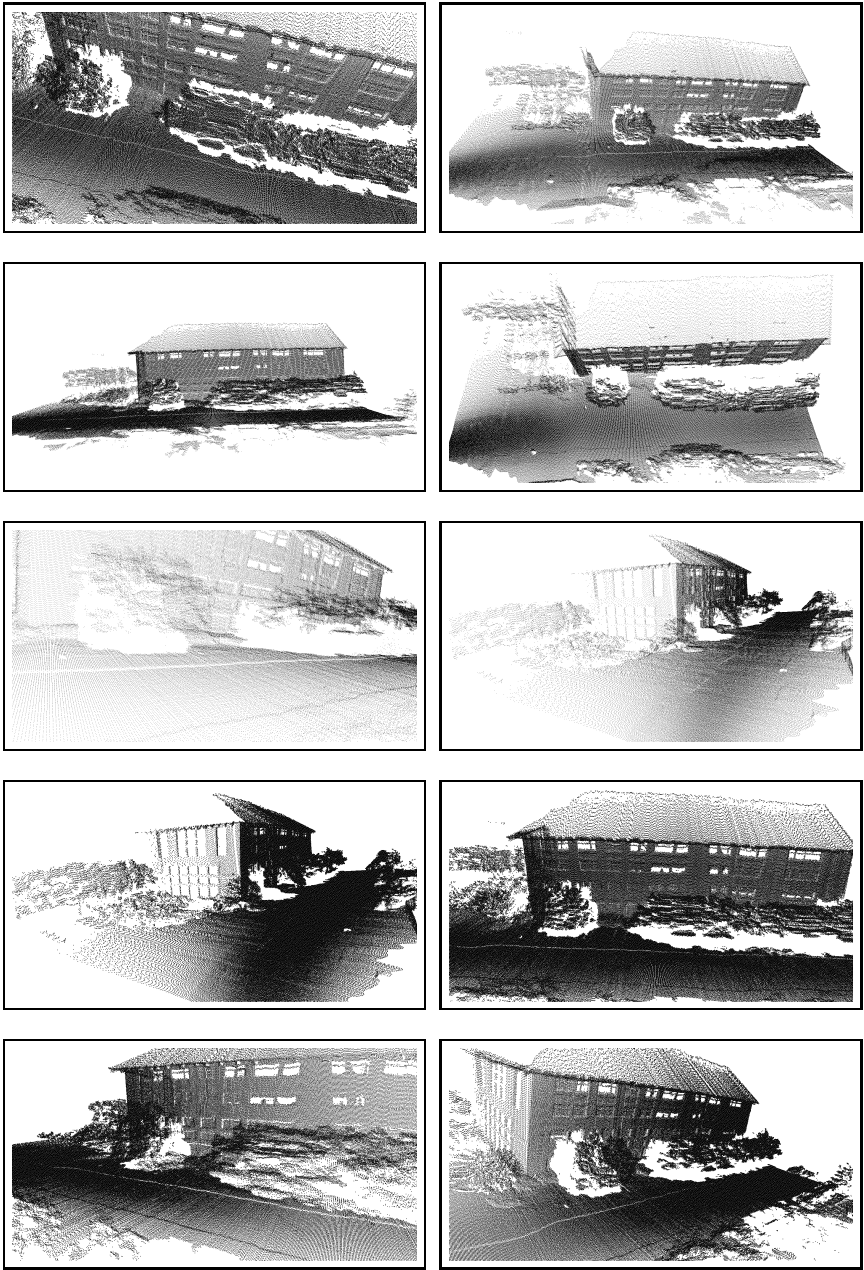


Fig. 7. Snapshots of the 3-D Map of the building shown in Figure 4. The map is represented as a VRML file and can be displayed using standards visualization tools.

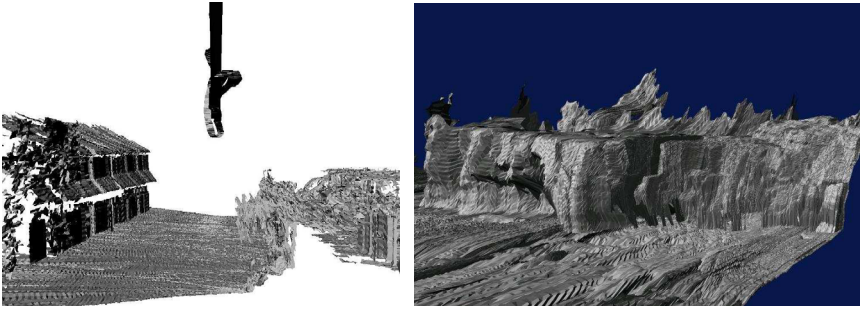


Fig. 8. Multi-polygonal 3-D models of a different urban site with a smaller building, and a cliff at the pacific coast near Davenport, CA. The left diagram also shows the vehicle's estimated path.

bottom map shows a street and a nearby building. The bottom map has also been analyzed for the flatness of the terrain; information that a ground vehicle might utilize to make navigation decisions.

Unfortunately, we do not possess ground truth information for any of the mapped buildings and structures. This makes it impossible to assess the accuracy of the resulting models. However, the models appear to be visually accurate, locally consistent. The spatial resolution of these models in the centimeter range.

4 Conclusion

This paper described initial results for an instrumented helicopter platform for 3-D ground modeling. A real-time algorithm was developed that integrates pose estimates from multiple sensors with range data, acquired by a 2-D laser range finders oriented perpendicular to the vehicle's flight direction. The algorithm uses a fast optimization technique to generate maps in real-time. Relative to prior work in [13,12], our approach utilizes the range data for vehicle localization, which results in maps that are spatially significantly more consistent and—we suspect—more accurate. Experimental results suggest that the maps acquired by our approach are of unprecedented detail and accuracy; however, the exact accuracy is presently not known. Nevertheless, we believe that the findings in this work are highly promising.

It is important to notice that this paper does not address the popular topic of autonomous helicopter control, see [1,17,7,4] for recent work in this area. However, an integration of accurate mapping and autonomous flight would make it possible to operate autonomous helicopters in rugged terrain, such as mountainous areas or caves. It would also open the door to the important problem of selecting safe landing pads in uneven terrain.

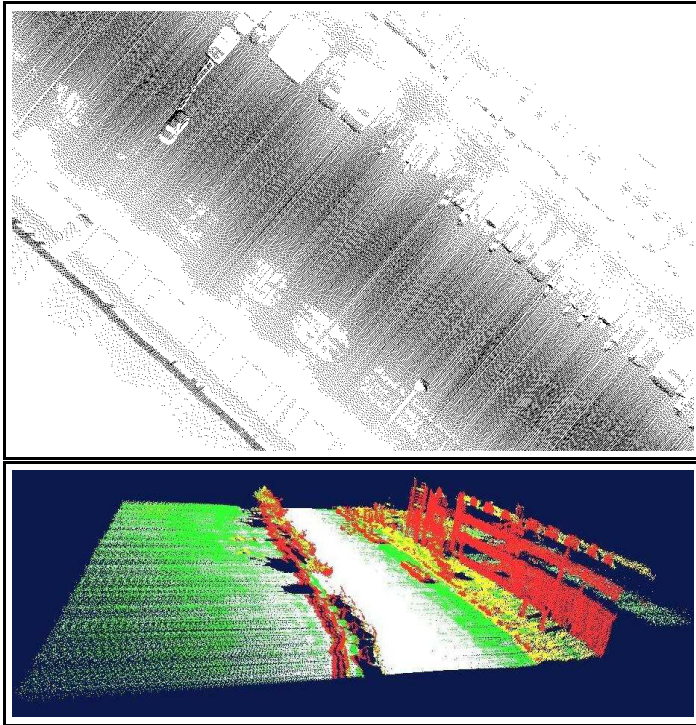


Fig. 9. Two maps acquired through autonomous flight, using a controller developed by others [15]. The bottom map has been analyzed for flatness of the terrain and colored accordingly.

Acknowledgment

The authors gratefully acknowledge the donation of two Stayton boards by Intel.

This research is sponsored by by DARPA's MARS Program (contracts N66001-01-C-6018 and NBCH1020014), which is also gratefully acknowledged. The views and conclusions contained in this document are those of the authors 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.

References

1. J. Bagnell and J. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *Proceedings of the International Conference on Robotics and Automation 2001*. IEEE, May 2001.
2. S. Becker and M. Bove. Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. In *Proc. of the SPIE Symposium on Electronic Imaging, San Jose, 1995*.

3. P. Besl and N. McKay. A method for registration of 3d shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
4. D. Drinka D. Coleman, D. Creel. Implementation of an autonomous aerial reconnaissance system. GeorgiaTech Entry into the 2002 International Aerial Robotics Competition, 2002.
5. P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *Proc. of the 23rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1996.
6. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localisation and map building (SLAM) problem. *IEEE Transactions of Robotics and Automation*, 2001. In Press.
7. J. Dittrich and E.N. Johnson. Multi-sensor navigation system for an autonomous helicopter. In *Proceedings of the 21st Digital Avionics Systems Conference*, 2002.
8. J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions of Robotics and Automation*, May 2001. In press.
9. D. Hähnel, D. Schulz, and W. Burgard. Map building with mobile robots in populated environments. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, 2002.
10. J. Leonard, J.D. Tardós, S. Thrun, and H. Choset, editors. *Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots (W4)*. ICRA Conference, Washington, DC, 2002.
11. F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.
12. R. Miller. *A 3-D color Terrain Modeling System For Small Autonomous Helicopters*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2002. Technical Report CMU-RI-TR-02-07.
13. R. Miller and O. Amidi. 3-D site mapping with the CMU autonomous helicopter. In *Proceedings of the 5th International Conference on Intelligent Autonomous Systems*, Sapporo, Japan, 1998.
14. D. Murray and J. Little. Interpreting stereo vision for a mobile robot. *Autonomous Robots*, 2001. To Appear.
15. A.Y. Ng, J. Kim, M.I. Jordan, and S. Sastry. Autonomous helicopter flight via reinforcement learning. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Proceedings of Conference on Neural Information Processing Systems (NIPS)*. MIT Press, 2003.
16. S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, Quebec City, Canada, 2001. IEEE Computer Society.
17. S. Saripalli, J.F. Montgomery, and G.S. Sukhatme. Visually-guided landing of an autonomous aerial vehicle. *IEEE Transactions on Robotics and Automation*, 2002.
18. S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2002.
19. S. Thrun, C. Martin, Y. Liu, D. Hähnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard. A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics*, 2003. To Appear.

Real-Time Navigation, Guidance, and Control of a UAV Using Low-Cost Sensors

Jong-Hyuk Kim¹, Salah Sukkarieh¹, and Stuart Wishart²

¹ Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
University of Sydney, NSW 2006, Australia
{jhhkim, salah}@acfr.usyd.edu.au

² BAESystems Australia
s.wishart@acfr.usyd.edu.au

Abstract. Applying low-cost sensors for the Guidance, Navigation and Control (GNC) of an autonomous Uninhibited Aerial Vehicle (UAV) is an extremely challenging area. This paper presents the real-time results of applying a low-cost Inertial Measurement Unit (IMU) and Global Positioning System (GPS) receiver for the GNC. The INS/GPS navigation loop provides continuous and reliable navigation solutions to the guidance and flight control loop for autonomous flight. With additional air data and engine thrust data, the guidance loop computes the guidance demands to follow way-point scenarios. The flight control loop generates actuator signals for the control surfaces and thrust vector. The whole GNC algorithm was implemented within an embedded flight control computer. The real-time flight test results show that the vehicle can perform the autonomous flight reliably even under high maneuvering scenarios.

1 Introduction

Over the recent years the use of low cost Uninhibited Aerial Vehicle (UAV) for civilian applications has evolved from imagination to actual implementation. Systems have been designed for fire monitoring, search and rescue, agriculture and mining. In order to become successful the cost of these systems has to be affordable to the civilian market, and although the cost/benefit ratio is still high, there have been significant strides in reducing this, mainly in the form of platform and sensor cost.

However, reduction in sensor cost also generally brings about a reduction in sensor accuracy and reliability. Coupled with the generally high mission dynamics that vehicles undertake within civilian aerospace due to the restricted mission areas, ensures that the design and implementation of these sensors is an extremely challenging area.

More importantly, the implementation of low cost sensors which are used for the Guidance, Navigation and Control (GNC) of the aerial vehicle is where most interest lies although little research is done. When applying a low cost Inertial Measurement Unit (IMU) there are still a number of challenges which the designer has to face. The main restrictions are the stability of the Inertial Navigation System (INS) degraded

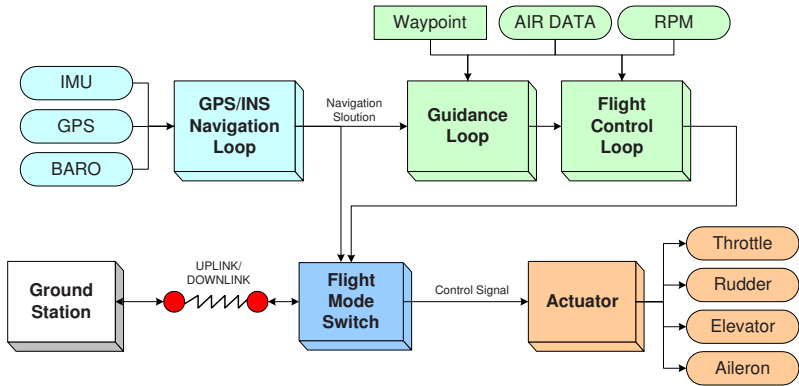


Fig. 1. The overall structure of navigation, guidance and control loop in UAV.

by the inertial sensor drifts. The quality and integrity of aiding sensors is also the crucial factor for the integrated system.

The Global Positioning System (GPS) can provide long-term stability with high accuracy. It also provides worldwide coverage in any weather condition. As a result lots of research have been done to optimally blend the GPS and INS [6][7][8]. Since the performance of the low cost GPS receiver can be easily degraded in high maneuvering environments, the quality and integrity of the GPS system becomes also a crucial factor. In case of GPS outage or fault conditions, the stand-alone INS quality then becomes the dominate factor. If the cost is a prohibitive factor in developing or buying an IMU, then improvements in algorithms, and/or fusing the navigation data with other sensors such as a barometer is required.

In this paper the authors present a low-cost navigation system which is successfully applied to the GNC of a UAV. Figure 1 depicts the overall GNC structure implemented in the Flight Control System (FCS). In remote operation mode, the remote pilot on the ground sends the control signals to the actuator via wireless uplink channel. The INS/GPS/Baro navigation loop downlinks the vehicle states to the ground station for vehicle state monitoring. When the autonomous mode is activated, the navigation solution is fed into the guidance and control loop and the onboard Flight Mode Switch (FMS) redirects the computed control outputs to the actuators.

The INS/GPS/Baro navigation loop makes use of a four-sample quaternion algorithm for the attitude update [2]. A complementary Kalman filter is designed with the errors in position, velocity and attitude being the filter states. It estimates the low-frequency errors of the INS by observing the GPS data with noises. In actual implementation, a U/D factorised filter is used in order to improve the numerical stability and computational efficiency [3][13]. Under high maneuverability, part of the GPS antenna can be blocked from the satellite signals which cause the receiver to operate in 2D height-fixed mode, hence to maximise the satellite visibility under these conditions, a second redundant receiver is installed and used.

The guidance loop generates the guidance commands from the vehicle states and the desired waypoint information. It computes required vehicle speed with respect to the air, height and bank angle. Then the flight control loop (or autopilot) generates actuator control signals to make the vehicle follow the guidance demands as well as to stabilise the vehicle. The control outputs are fed to the control surfaces, or aileron, elevator and rudder, and thrust vector [4][5].

The whole navigation, guidance and control software package was developed using the C++ class methodology. They are implemented and installed within a FCS which has a multitasking capability. The developed flight vehicle, MK3-Brumby, is a delta fixed wing platform with a pusher prop configuration. It can carry an extra sensor payload for different mission objectives. These sensors include radar, vision and laser; and hence the accuracy of the navigation loop is not only for control purposes but also for the accurate registration of landmarks which these mission sensors pick up. The real-time flight tests show that the navigation system can provide accurate and reliable 3D navigation solutions as well as to perform the guidance and control task reliable.

Section 2 will briefly describe the aircraft system including the flight platform, onboard systems, and ground system. Section 3 will provide the details of the low-cost sensors used in this work. Section 4 will describe the navigation loop. Section 5 will detail the structure of the guidance and control loop. Section 6 will present the result of a real-time autonomous flight test. Conclusions are then provided in Section 7.

2 Aircraft Systems

The physical aircraft system comprises of the flight platform, on-board systems, communication links, and ground station.

The flight platform, MK3-Brumby, is shown in figure 2(a). It is a delta fixed wing platform with a pusher prop configuration and is capable of flying at $100kts$ and up to $500m$. The platform can carry up to $11kg$ of additional mission sensors (the payload sensors are part of the flight platform). Figure 2(b) shows four UAVs built for the demonstration of cooperative data fusion amongst multiple vehicles [12].

The on-board systems consist of an FCS, FMS, vision system, scanning radar and/or laser system, and an air data system. The vision and radar systems are mission specific nodes and perform multi-target tracking, target registration and decentralised data fusion. The air-to-air/air-to-ground communication links are established for the decentralised fusion purposes, remote control operation, differential GPS data uplink, and telemetry data. The ground station consists of a DGPS base station, weather station, hand-held controller, and monitoring computer. Additional mission computers performs the monitoring of the mission objectives.

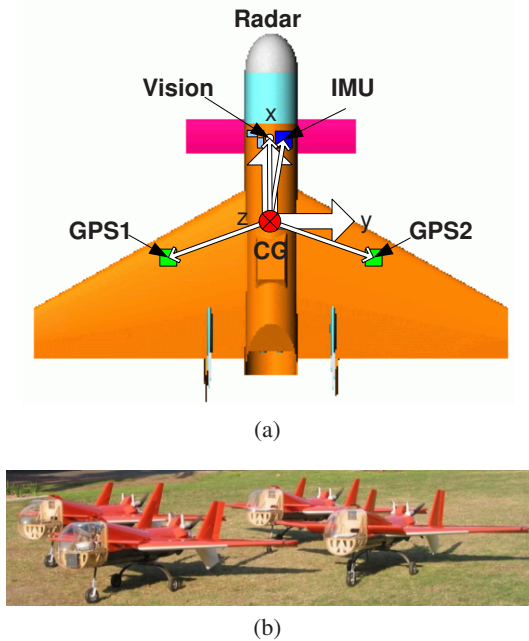


Fig. 2. (a) The location of onboard sensors in the Brumby MK-3 UAV. It is equipped with IMU, GPS, vision camera, radar scanner, air data system, and flight electronics and (b) four flight vehicles: MK3-Brumby. Four sets of identical GNC modules are installed on the vehicles with different mission load.

3 Low-Cost Sensors

The Inertial Measurement Unit (IMU) is an Inertial Science ISIS unit. Figure 3(a) shows the IMU installed in the fuselage of the vehicle. It is a short-range tactical grade IMU with three voltage resonant gyros and three accelerometers. It has a gyro bias repeatability within $0.01^\circ/s$ and accelerometer bias repeatability within $1mg$. The IMU has very small physical dimension and consumes only 5 watts of power. The unit can provide inertial data from $100Hz$ to $400Hz$ through an RS422 serial interface. The resolution of the gyro is too low to detect the earth rotation rate and the self-azimuth alignment can not be performed for initialisation. In this work, the initial heading is uplinked from the ground station. The IMU makes use of an internal temperature lookup table for the temperature compensation. However, in field tests, the turn-on bias was not removed effectively so the calibration process was performed before starting the navigation. This was performed using a tilt sensor as shown in the upper right part on figure 3(a).

The GPS receiver is the BAE Systems *AllstarTM* receiver. Two GPS cards are stacked on the FCC as shown in figure 3(a). Their antennas are installed on each of the wings as shown in figure 3(b). The wing frame was made of fiber glass and the GPS signal loss was negligible. The master GPS receiver provides navigation solu-

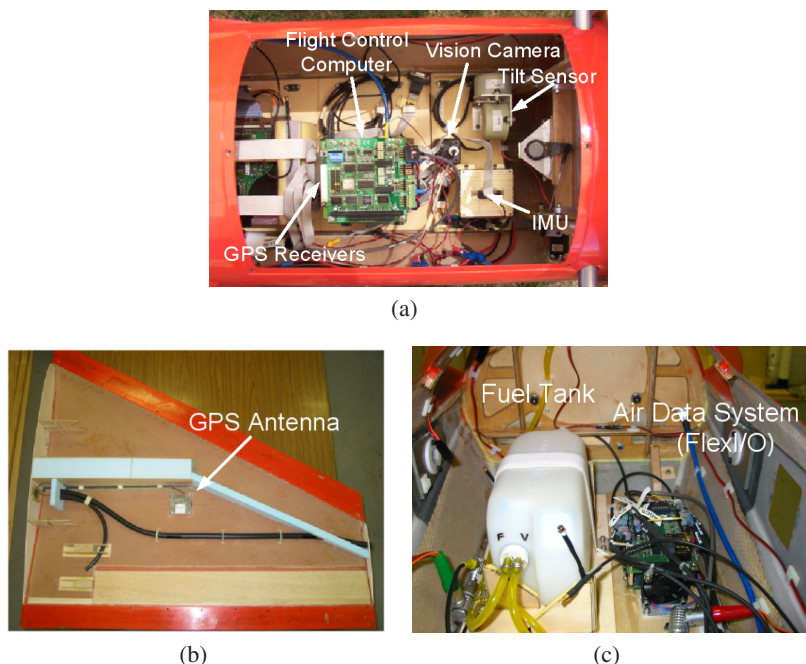


Fig. 3. (a) A view from the front hatch of the vehicle. A low-cost IMU is installed (lower right) next to the vision camera pointing downwards and a tilt sensor (upper right). Two GPS receivers are stacked on the flight control computer. The sensors are connected to the flight control computer via serial and parallel lines. (b) GPS antenna installed inside the wing of the vehicle. Two antennas are installed on both of the wings to provide redundancy. (c) The rear hatch of the vehicle. The air data system can be seen next to the fuel tank. It is implemented using a FlexIO card which can be configured in multi-purpose data sampling.

tions to the Kalman filter. The slave receiver provides redundant navigation solution as well as the raw measurement data for the purpose of GPS attitude determination which is not implemented yet.

The air data system is implemented by using a FlexIOTM card developed by the University of Sydney as shown in figure 3(c). It is a multi-purpose A/D (Analog to Digital) and D/A (Digital to Analog) card which can sample various analog signals such as air pressure, temperature, engine RPM, battery current and voltage. The sampled data are sent to the FCS via RS232 serial interface and FCS downlinks them to the ground station for vehicle monitoring. The measured air data is used in the guidance and control loop and the baro-altimeter data is used to stabilise the vertical axis of the INS.

4 Navigation Loop

The navigation loop plays a key role in aircraft system. Its navigation outputs are used in guidance and control and affect the performance of target registration and picture compilation tasks. It also has to provide precise timing synchronisation to other sensor nodes.

The core of the navigation loop is the strapdown INS and the Kalman filter. The strapdown INS provides continuous and reliable position, velocity and attitude with sufficiently high rates. The Kalman filter estimates the navigation errors by blending the GPS observation or baro-altimeter data running as a background task.

The INS is mechanised in an earth-fixed tangent frame as shown in figure 4(a). It computes position, velocity and attitude of the vehicle with respect to the reference frame by numerical integration of the accelerations and angular rates. In this mechanisation scheme, the reference frame is assumed as a non-rotating inertial frame. The short mission flight time and the frequent GPS corrections makes this assumption valid without significant performance degradations in most of the local terrestrial navigators. If the INS should perform long-range missions without GPS corrections, the INS will require a more precise mechanisation scheme to remove systematic errors like frame rotation effect and coriolis force. In this earth-fixed tangent frame mechanisation, the coriolis and transport rates term are not calculated. The navigation outputs are in MGA (Map Grid Australia) coordinate format instead of WGS-84 coordinate as it is convenient to exchange the vehicle states and relative target observations between multiple UAVs.

The fusion Kalman filter is the heart of the navigation system. In a low-cost system the IMU errors like bias, scale factor error and random walk noise dominate the INS error growth. These INS errors typically show low dynamics and its models have been well developed [1]. The INS error model in earth-fixed tangent frame is used in this paper [11].

Figure 4(b) shows the complementary filter structure that deals with the INS error state instead of the total vehicle states using the INS error model. Since the INS error has low frequency dynamics, the filter can run in relatively low sample rates with lower priority. When the external aiding information from GPS or baro-altimeter is available, a measurement residual is generated by subtracting the actual measurement from the INS predicted measurement. The resulting residual contains the INS error as well as the measurement error. The measurement error typically has high-frequency noise and can be modelled as white (or broadband white) noise process. The Kalman filter suppresses this high-frequency noise and estimates the INS error by the low-pass filtering nature. If the measurement error contains additional dynamic error, this should be properly modelled inside the filter [3].

5 Guidance and Flight Control Loop

The guidance loop forms an outer control loop in autonomous mode. It computes guidance demands to force the vehicle to follow the desired way-point. The flight

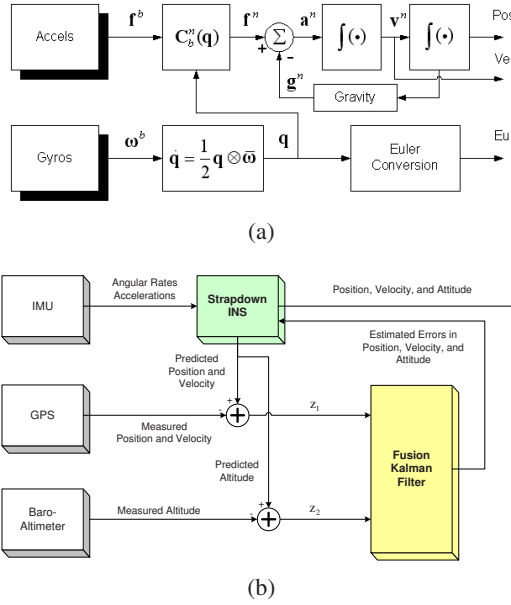


Fig. 4. (a) The INS mechanisation in the earth-fixed tangent frame and (b) the complementary INS/GPS/Baro Kalman filter.

control (or autopilot) loop forms the inner control loop and it generates the actual control signals to follow the guidance objectives as well as to stabilise the vehicle attitude and its rate.

The guidance loop generates the guidance demands from the current vehicle states and the next waypoint information. The guidance demands are desired vehicle speed with respect to the air, desired height and bank angle as shown in figure 5(a). If the autonomous mode is activated, it selects the appropriate next waypoint depending upon the guidance state. Then it decides if the waypoint has been intercepted or missed. If it is not intercepted it determines the Line Of Sight (LOS) angles and LOS rates to the next waypoint. Based on this information it computes the lateral acceleration required to intercept the next waypoint and converts this acceleration to the desired bank angle with a set of additional guidance demands: airspeed and height [4]. The guidance loop implemented in this work updates its guidance demands every 10Hz.

The flight control loop controls the vehicle's attitude and attitude rates as well as the vehicle speed with respect to the air. The block diagram of the control loop is shown in figure 5(b). It performs speed control, height and height rate control, bank angle control, heading control, turn compensation and elevation control by using the guidance demands and measured vehicle states. It generates actuator signals for the engine throttle, rudder, elevator, and aileron. In this work, the control loop generates the control signals every 50Hz which is limited by the bandwidth of the

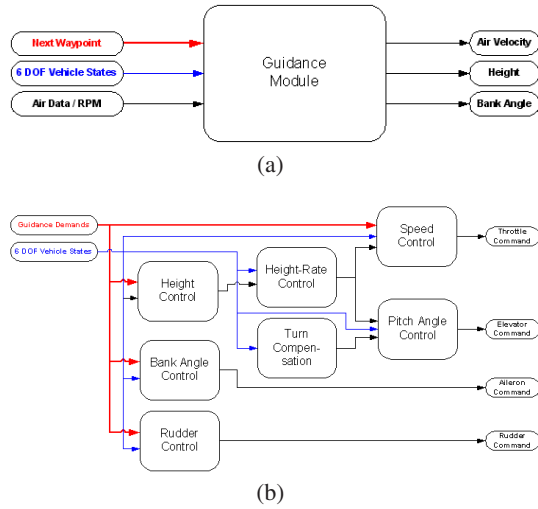


Fig. 5. (a) The structure of guidance module and (b) that of control structure.

electric servo actuators. The control loop is the most time critical task in autonomous flight so it is allocated to the highest priority inside the FCS. Most of the guidance and control algorithms were verified and tested using the Hardware-In-The-Loop (HWIL) simulator which uses the vehicle model and simulated sensor data in the laboratory. However the true vehicle model and control action to the vehicle can only be verified through real-time flight tests.

6 Real-Time Results

Several real-time autonomous flight tests were performed at the test site of the University of Sydney. To evaluate the navigation performance, a more precise navigation system which usually has an order of accuracy is required as a reference system. However, due to the limited payload capacity of the UAV, this was not adopted. Instead, the covariance and innovation data from the fusion filter were used to evaluate the performance and the filter consistency.

A flight test was performed on April 17, 2003. The purpose of this test was to verify the real-time autonomous flight with different sets of waypoint scenarios. The total flight time was 25min and the desired flight height in autonomous mode was set to 328ft above the ground and the desired air speed was set to 80kts.

The full flight trajectory is shown in figure 6(a). The blue solid line is the real-time estimated positions and the red dotted line is the GPS indicated position. Except take-off and landing the vehicle flew in autonomous mode. Figure 6(b) depicts the vehicle trajectory from the take-off in remote control mode until the activation of autonomous mode. Figure 6(c) shows the trajectory in autonomous mode using the first scenario. During the first six rounds, the waypoints are activated in CCW se-

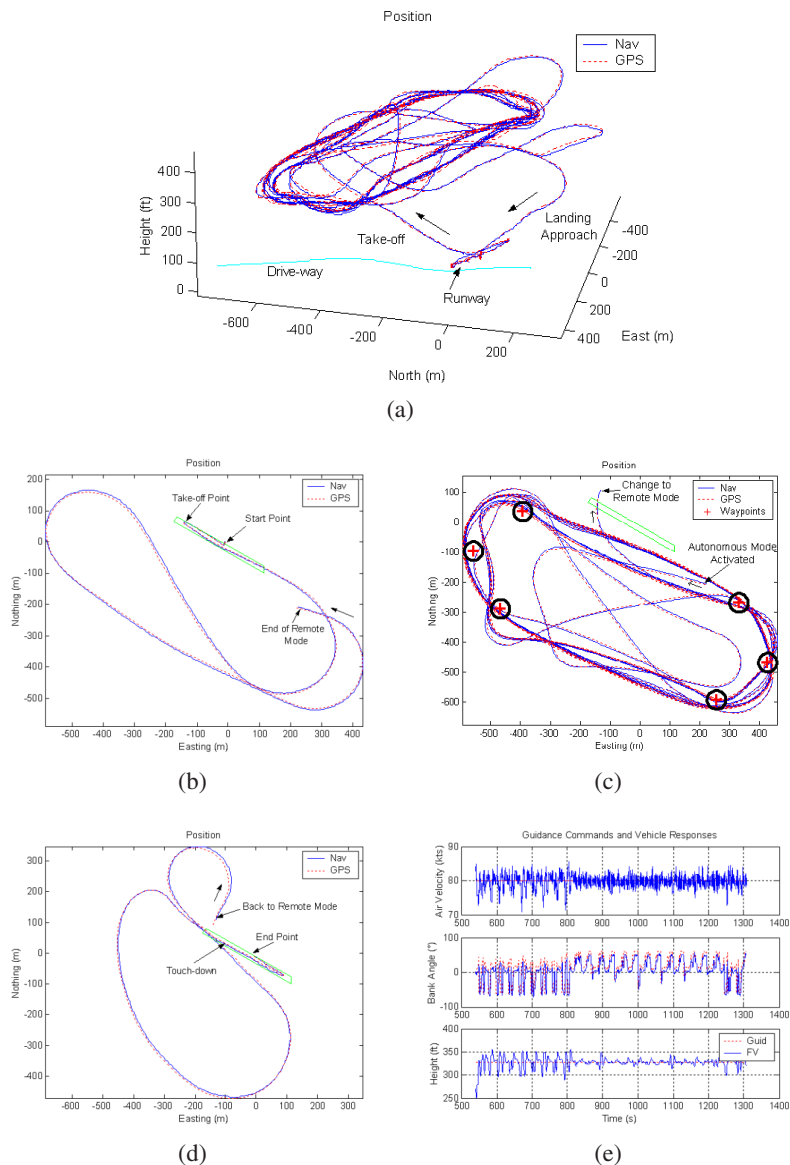


Fig. 6. (a) Real-time estimated 3D trajectory. (b) The vehicle takes off in remote control mode and (c) autonomous mode is activated from the ground station. The guidance loop uses the waypoints to find out the next destination. Several scenarios are applied which require different dynamics on the vehicle. (d) The vehicle is then back to remote control mode to prepare the landing approach. (e) The vehicle responses to the guidance demands in autonomous mode. The desired air velocity was set to 80 knots and desired altitude above the ground was set to 328 ft. It can be seen during the first six rounds the vehicle undergoes more severe dynamics.

quences with some transition sequences. After the seventh round the way points are changed to CW sequences. The former scenario requires higher vehicle maneuvers than the latter. This can be seen in the guidance response in figure 6(e) which show larger overshoots in measured vehicle speed and height as well as in bank angle. After performing the last scenario, the vehicle mode is changed to remote mode to prepare the landing approach and is returned to the ground station as shown in figure 6(d). During the test the number of GPS satellites changed from eight to four during banking. The estimated vehicle 1σ uncertainties in position were maintained within $2m$ in three axes as shown in figure 7(a). The attitude uncertainties are shown in figure 7(b) where the roll and pitch uncertainties are maintained under 0.4° and the heading is under 0.8° .

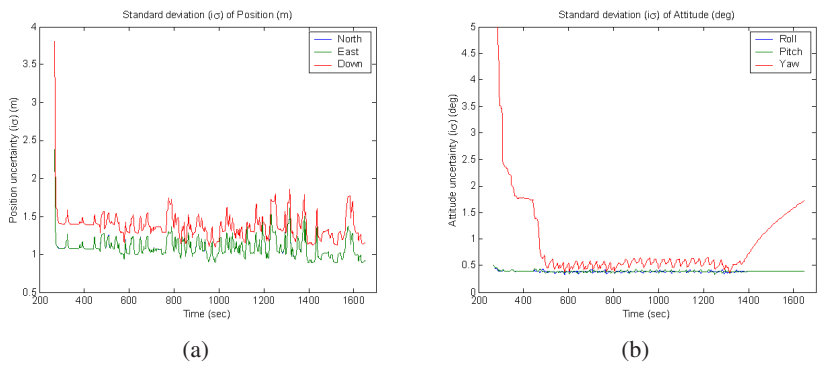


Fig. 7. (a) The evolution of the vehicle 1σ uncertainty in position and (b) that of attitude.

7 Conclusions

This paper presents the real-time implementation and result of the navigation, guidance, and control of UAV using low-cost sensors. A low tactical grade IMU is used for the inertial navigation and two low-cost commercial GPS receivers are used to estimate and correct the INS error. The air data system is built using a customised analog/digital conversion card. It provides air pressures, temperature and engine control signal for the guidance and control loop. The GNC algorithm was developed in C++ methodology and implemented in an embedded computer. Real-time autonomous flight tests shows reliable and accurate GNC performances with several way-point scenarios. With the low-cost aircraft platform and GNC module, the FCS becomes a modular and powerful autonomous system. This system is planned to be used for decentralised data fusion between multiple UAVs. The future work is to make this system to be fully autonomous which does not rely on the external GPS aiding signal by exploring the unknown environments and map building process.

Acknowledgement

The authors wish to thank BAE Systems for their support in providing funding and systems engineering towards this project.

References

1. D.G. Meskin and I.Y. Bar-Itzhack, "Unified Approach to Inertial Navigation System Error Modeling," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 3, pp. 648–653, May-June 1992.
2. P. G. Savage, "Strapdown Inertial Navigation Integration Algorithm Desogn Part I: Attitude Algorithms," *Journal of Dynamic Systems, Measurement and Control*, vol. 21, no. 1, pp. 19–28, January-February 1998.
3. M. Grewal and A. Andrew, *Kalman Filtering - Theory and Practice*. Prentice-Hall, Inc., 1993.
4. B.L. Stevens, F.L. Lewis, *Aircraft Control and Simulation*, John Wiley and Sons, Inc., 1992.
5. Skogestad, Postlethwaite, *Multivariable Feedback Design*, John Wiley and Sons, Inc., 1997.
6. A. H. Lewantowicz. Architectures and GPS/INS Integration: Impact on Mission Accomplishment. In *IEEE Position, Location and Navigation Symposium*, pages 284–289, 1992.
7. S. Snyder, B. Schipper, L. Vallot, N. Parker, and C. Spitzer. Differential GPS/Inertial Navigation Approach/Landing: Flight Test Results. In *IEEE Transactions on Aerospace and Electronic Systems Magazine*, pages 3–11, 1992.
8. R.L. Greenspan. GPS and Inertial Navigation. In *American Institute of Aeronautics and Astronautics*, 1996.
9. R.E. Phillips and G.T. Schmidt. GPS/INS Integration. In *Advisory Group for Aerospace Research and Development (AGARD)*, Lecture series 207:9.1–9.10, 1996.
10. S. Sukkarieh, E.M. Nebot, and H. Durrant-Whyte. A High Integrity INS/GPS Navigation Loop for Autonomous Land Vehicle Applications. In *IEEE Transactions on Automatic Control*, 15:572–578, June 1999.
11. S. Sukkarieh, "Aided Inertial Navigation Systems for Autonomous Land Vehicles," PhD thesis, Australian Centre for Field Robotics, The University of Sydney, 1999.
12. S. Sukkarieh, A. Goktogan, J. Kim, E. Nettleton, J. Randle, M. Ridley, S. Wishart, and H. Durrant-Whyte, "Cooperative Data Fusion Amongst Multiple Uninhabited Air Vehicles," in *International Conference on Service and Experimental Robotics*, Italy, 2002.
13. J.H. Kim and S. Sukkarieh. Flight Test Results of a GPS/INS Navigation Loop for an Autonomous Unmanned Aerial Vehicle (UAV). In *Proceedings of the 15th International Technical Meeting of the Satellite Division of the Institute of Navigation*, pages 510–517, September, OR, USA, 2002.

A Compact Millimeter Wave Radar Sensor for Unmanned Air Vehicles

Ali Haydar Göktoğan, Graham Brooker, and Salah Sukkarieh

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The Rose Street Building J04
The University of Sydney 2006 NSW Australia
{agoktogan, gbrooker, salah}@acfr.usyd.edu.au
<http://www.acfr.usyd.edu.au>

Abstract. This paper presents a compact Millimeter Wave (MMW) radar unit that has been developed to be used as a Range, Bearing and Elevation (RBE) sensor on the Brumby Mk III Unmanned Air Vehicles (UAV). The Brumby MkIII is the flight platform used in the Autonomous Navigation and Sensing Experimental Research (ANSER) project which is focused on the development and demonstration of the Decentralised Data Fusion (DDF) and Simultaneous Localisation and Map Building (SLAM) algorithms on multiple UAVs. In the airborne DDF and SLAM demonstrations with UAVs, it is essential to have a terrain sensor on board for the RBE measurements of the ground targets. The focus of this paper is the hardware and software components of the MMW radar as an RBE sensor in the context of the ANSER project.

1 Introduction

There are a number of groups researching the concepts of the Decentralised Data Fusion (DDF) and Simultaneous Localisation and Map Building (SLAM) algorithms in field robotics applications, mainly using ground vehicles, experimental mobile robots or submersibles, all of which have slow dynamics [1][3][5]. In most of these experiments, sensors, like cameras and underwater sonars have fixed view frusta. Other sensors such as scanning lasers have limited control over where sensor can be pointed [5]. Ground vehicles offer flexibility on the selection of the sensors since in most cases, power requirements, weight, size and shape of the sensor are not very critical [1].

Using a UAV as the test platform for the DDF and SLAM techniques poses many problems that are not experienced on the ground vehicles. Commercially available sensors for UAVs are very limited or too expensive for most of the academic researches. The sensor development for UAVs is a challenging problem. The high speed, coupled with the random motion that the vehicle experiences due to wind gusts and vibration make the control of the sensor an extremely difficult problem. The weight and power limitations combined with the volumetric size and shape constraints imposed by the flight dynamics characteristics of the UAV make the sensor development even more of a difficult task.




The Autonomous Navigation and Sensing Experimental Research (ANSER) project is one of the flagship projects being conducted at the Australian Centre for Field Robotics (ACFR). It is focused on the development and demonstration of DDF and SLAM algorithms on multiple sensing nodes, including the millimeter wave (MMW) radar, existing on multiple UAV platforms [4][6][7][8].

This paper will present the MMW radar sensor designed and developed for use in the ANSER project. The next section will present Brumby Mk III UAV, the flight platform used in the flight experiments. In Section 3 the MMW radar sensor is introduced as a payload for Mk-III. The basic principles and the design parameters of the MMW radar are explored in Section 4. Details of the MMW radar hardware are presented in Section 5. The transformations of the target coordinate are given in Section 6. Section 7 will discuss the feature extraction method from the power spectrum. Finally a conclusion and further work will be provided in section 8.

2 The UAV

ANSER experiments are conducted with multiple Brumby Mk III UAVs. Mk III offers a speed range of 55 to 100 Knots, and roll rate of up to 300 deg/sec. A payload mass of over 13kg with space for over one hour of fuel and a larger payload volume than its predecessors. The airframe is modular in construction to enable the replacement or upgrade of each component. The specifications of the Brumby series UAVs are summarized in Table 1.

Table 1. Brumby Mk I-II-III UAV Specifications

	Mk I	Mk II	Mk III
			
MTOW	25kg	30kg	45kg
Wing Span	2.3m	2.8m	2.9m
Engine	74cc	80cc	150cc
Power	5Hp	5.5Hp	16Hp
Max Speed	100 Knots	100 Knots	100 Knots

The Brumby UAVs are delta wing, pusher aircraft. The delta wing design requires no tail, is compact for a given wing area and has a minimal component count. The pusher design leaves a clear space in the nose for the MMW radar and other sensors. The engine is located at the back and sensors in the front section of the aircraft stay free from exhaust contamination. The MMW radar is designed to be carried in the nose cone of the UAV. The larger and transparent nose cone of the Mk III has a rear fairing to blend the nose cone to the fuselage. The Mk III has another distinguishing

feature of fixed canards to the front of the aircraft that was chosen as the quickest, simplest, lowest risk, most cost effective way to reduce the landing speed.

The payload attachments in the Brumby Mk III are modular to enable easy interchange of the payloads. Each payload position is provided with 24v DC power and an Ethernet and/or Controller Area Network (CAN) connection to the other onboard systems. As a payload sensor, the MMW radar is connected to these busses.

3 MMW Radar Payload

It is important for decentralized data fusion and control amongst multiple UAVs that at least one of the platform carries a sensor which returns both Range, Bearing and Elevation (RBE) of the ground targets [6][7].

For the RBE measurements, an ultrasonic sensor is not feasible in the UAV applications because of the limited range of the sensor and high speed of the UAV. Scanning laser and radar sensors are the only practical alternatives. They offer different qualities; the laser sensor provides better tracking accuracy and it is less expensive than radar, but laser sensor suffers reduced performance in fog, rain and dust.

The radar has a wider beam-width and therefore it has less accurate angular resolution, but multiple targets may be detected with radar in a single observation. Radar has comparable range accuracy and it can operate in adverse weather conditions. Most of the airborne military applications use the atmospheric window around 94GHz. 77GHz which is designated for automotive applications also exhibits a low atmospheric absorption [9].

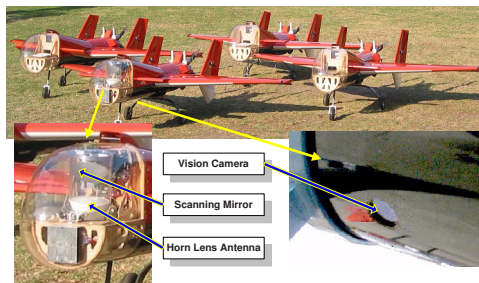


Fig. 1. 77GHz MMW radar on attitude-stabilizer gimbal mechanism with a scanning mirror is mounted in the nose cone of Brumby Mk III UAV. The UAV also carries a vision camera located under the fuselage.

The ACFR has built and used the 77GHz ($\lambda \approx 4\text{mm}$, E-Band) MMW radars on various innovative field robotics applications for a number of years [1][2][10][11]. The MMW radar discussed in this paper has also been built in house and operates

at 77GHz. The radar front end is placed on a specially designed attitude stabilizing gimbal with a scanning mirror. The whole radar unit is compact and light enough to be mounted into the nose cone of the Brumby Mk III UAV as shown in Figure 1.

4 MMW Radar Principles and Design Parameters

The MMW radar is built to operate using the Frequency Modulated Continuous-Wave (FMCW) principle for range measurements [1][9]. A linear frequency chirp f is transmitted from the radar for a duration of T_d . The transmitted signal propagates along the beam axis with a divergence mainly determined by the radius of the horn lens antenna. The radiated electromagnetic wave reflect back from the target and is received by the receiver. The transmitted and received signals are mixed and the result of the mixing is the beat frequency of f_b . The value of the beat frequency is determined by the round trip time of T_p . It is the time required for the transmitted signal to return to the receiver after being reflected back from the target.

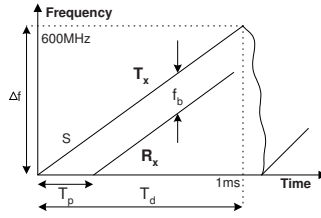


Fig. 2. In the Frequency Modulated Continuous-Wave (FMCW) radars, the target range is calculated from the frequency difference (f_b) between transmitted (T_x) and received (R_x) signals. f_b defined by round trip time T_p of the transmitted signal to reflect back from the target.

The sweep bandwidth of $f=600\text{MHz}$ and sweep duration of $T_d=1\text{ms}$ are used in our MMW radar. As shown in Figure 2, the range resolution is a function of the sweep bandwidth f . With these design parameters, and the Equation 1, where c is the speed of electromagnetic radiation, it is possible to achieve a range resolution of $\delta R_{chirp}=0.25\text{m}$.

$$\delta R_{chirp} = \frac{c}{2f} \quad (1)$$

The range accuracy is characterized by the linearity Lin of the sweep as defined in Equation 2 where $S=\delta f/\delta t$ is the chirp slope.

$$Lin = \frac{S_{max} - S_{min}}{S_{min}} \quad (2)$$

The nominal linearity of the radar is about 0.1%. From Equation 3, the resolution for a target at $R=500\text{m}$ is obtained as $\delta R_{lin}=0.5\text{m}$.

$$\delta R_{lin} = R \text{ Lin} \quad (3)$$

The final system range resolution δR , can be described by Equation 4 which concludes as $\delta R=0.56\text{m}$ at the maximum range, and improving as the range decreases.

$$\delta R = \sqrt{\delta R_{chrip}^2 + \delta R_{lin}^2} \quad (4)$$

The beat frequency f_b for the target at $R=500\text{m}$ can be calculated from Equation 5 as $f_b=2\text{MHz}$.

$$f_b = T_p \frac{\delta f}{\delta t} \approx T_p \frac{f}{T_d} \approx \frac{2R}{c} \frac{f}{T_d} \quad (5)$$

Target range R to beat frequency f_b mapping is summarized in Equation 6.

$$\frac{R}{f_b} = \frac{T_d c}{2 f} = 250\text{m}/\text{MHz} \quad (6)$$

Spectrum analysis of the received signal reveals the range of the targets in the radar beam. Details of the signal processing hardware are presented in the next section.

5 MMW Radar Hardware

Building blocks of the MMW radar sensor payload are shown in Figure 3. The initial real time, hardware-in-the-loop integration tests of the MMW radar and the attitude stabilizing gimbal are performed in the Real time Multi-UAV Simulator (RMUS) environment [4]. The RMUS provides a testing and validation mechanism for simulation and the real demonstration of multiple UAVs conducting both decentralised data fusion and control. Figure 4 shows a bench-top test setup for the integration tests of the MMW radar unit with the navigation computer.

Radar Front End and Horn-Lens Antenna : The transmitter and receiver in the MMW radar front end unit are configured to run at 77GHz. It is connected to a horn-lens antenna for target acquisition and to small low-gain horn for altimetry reading.

Equation 7 is an empirical formula that relates 3dB beamwidth to antenna diameter D for a typical antenna [9]. From this equation we can conclude that, for a given frequency (77GHz, $\lambda \approx 4\text{mm}$), the antenna aperture is the only parameter we can modify to constrain the beamwidth. As pictured in Figure 5, beamwidth defines the angular resolution.

$$\theta_{3dB} = \frac{70\lambda}{D} \text{deg} \quad (7)$$

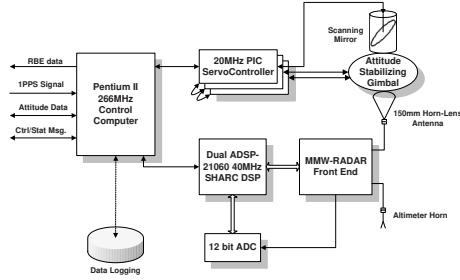


Fig. 3. The main building blocks of the MMW radar sensor payload designed to be used on Brumby Mark-III UAV.

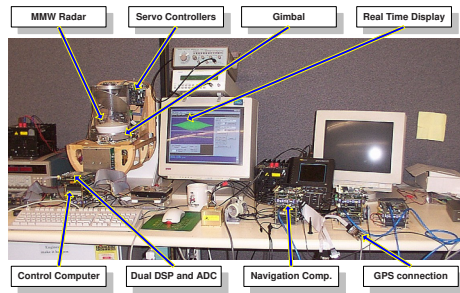


Fig. 4. The MMW radar unit on attitude-stabilizer gimbal mechanism is connected to radar control computer. The navigation computer with attached IMU and GPS units sends attitude data to radar control computer in RMUS environment.

The physical size of the UAV is the major constraint for the determination of the antenna diameter and hence the angular resolution of the MMW radar sensor. The 150mm horn-lens antenna used on the Brumby Mk III UAV generates a 1.8° 3dB beamwidth.

Analog to Digital Converter : The MMW radar front end is connected to a high speed 12 bit Analog to Digital Converter (ADC) unit. Digitized data is stored in 4096 word buffer and than transferred to DSP board memory through the high speed SHARC link ports.

A Field Programmable Gate Array (FPGA) controls the operation of the ADC and selection of either internal or external clock and trigger source. Although the ADC unit supports up to 10MHz conversion rate, it is operated at 5MHz. The effect of the sampling rate can be explored from Equation 6.

Digital Signal Processing Unit: The Digital Signal Processing (DSP) unit consist of dual ADSP-21060 processors from Analog Devices running at 40MHz on a single PC104 board. The Super Harvard Architecture (SHARC) gives the ADSP-21060 fast data transfer and processing power. ADSP-21060 feature link ports

that allow point-to-point interprocessor communication in multiprocessing systems. Through the link ports, data can be transferred between the SHARC DSPs and peripheral devices without any processor overload. The ADC board is connected to the DSP board via link ports which allows new data to be transferred from ADC to DSP whilst the previous data is being processed.

The signal processing software running on the DSP unit performs the Hamming window on the digitized signal to limit the effects of discontinuities at the start and end of the sampling period and hence improve the spectral integrity. Researches have proposed various window functions for similar effect [1][2]. After the window operation, the digitised data is subject to the Fast Fourier Transform (FFT). The optimized FFT algorithm implemented to run on ADSP-21060 calculates the FFT of 4K word data in 1.8ms. Analysis of the FFT output for feature extraction is detailed in Section 7.

Attitude-Stabilizing Gimbal and Scanning Mirror: As the UAV manoeuvres during the flight, it's attitude (ie. roll and pitch angles) changes continuously. The MMW radar sensor is mounted on the attitude-stabilizing gimbal to keep the grazing angle of the MMW radar fixed. The gimbal has two degrees of freedom; one on the roll and the other on the pitch axis. It is controlled in real time to counteract the UAV's motion on these axes.

The scanning mirror on top of the gimbal reflects the millimeter waves radiated from the horn-lens antenna. Received echo signals follow the same path from target to receiver module via scanning mirror and horn-lens antenna while the scanning mirror rotates five revolutions every two seconds. A simplified scan pattern is shown in Figure 5.

Servo Controller and Motor Drivers : The gimbal axes and scanning mirror are driven by separate brushless servo motors. Each motor is controlled by a 20MHz PIC microprocessor based servo controller and motor driver board. These boards are daisy chained on a RS485 serial bus and commanded by a 266MHz Pentium II control computer as shown in Figure 3.

Control Computer : The control computer receives aircraft attitude data from the navigation computer of the UAV. This data is used to command the servo motor controller boards to generate attitude stabilization of the gimbal. The control computer also performs the initialization, and health monitoring of the DSP and servo controllers. It communicates with higher level controllers via control and status messages.

The DSP unit passes the result of the FFT analysis ie. target range to the control computer, than the control computer reads the current gimbal state (gimbal axes and scanning mirror angles) from the servo controllers. The UAV attitude data received from the navigation computer is combined with the target range and gimbal state to form a RBE data package which is later converted into information form for DDF by higher level programs.

The RBE data packages need to be time stamped with highest the possible resolution practically achievable. The internal high resolution counter data of the Pentium processor is used for the time stamp. The One Pulse Per Second (1PPS)

signal received from the Global Positioning System (GPS) is connected to the control computer to generate hardware interrupts. The interrupt service routine synchronizes the high resolution counter data to the global time which is shared amongst the other DDF nodes.

6 Coordinate Transforms

In the context of DDF, sensors should output the information-form estimate of the target location in the earth frame. Equation 8 is the transformation relating the target t to the radar sensor sr , the sensor to the UAV body b , and then the body to the earth frame e where \mathbf{P}_{f1}^{f2} is position of the frame $f1$ in the frame $f2$ and \mathbf{C}_{f1}^{f2} is the direction cosine matrix which relates how the frame $f1$ is rotated with respect to the frame $f2$. The coordinate transformations are illustrated in Figure 5.

$$\mathbf{P}_t^e = \mathbf{P}_b^e + \mathbf{C}_b^e \mathbf{P}_{sr}^b + \mathbf{C}_{sr}^b \mathbf{P}_t^{sr} \quad (8)$$

The position of the target in the radar sensor frame \mathbf{P}_t^{sr} , is defined by two angles θ_t^{sr} , ψ_t^{sr} and the vector \mathbf{r}_t^{sr} . \mathbf{r}_t^{sr} is the vector which points to target from the radar sensor frame.

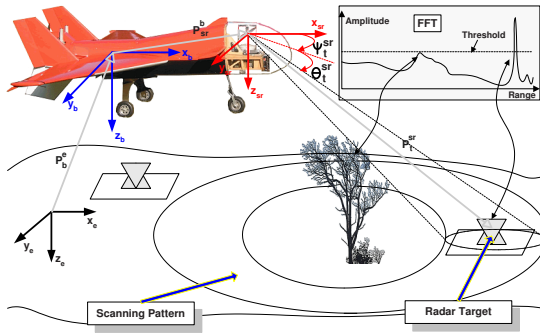


Fig. 5. Scanning mirror rotates around z_{sr} axis. The look-down angle θ_t^{sr} , of the radar beam is fixed. Analysis of the FFT power spectrum reveals the target range.

The magnitude of the \mathbf{r}_t^{sr} vector is the range of the target found from the feature extraction process on the FFT power spectrum. θ_t^{sr} is the grazing angle of the target from the sensor frame on the $x_{sr}y_{sr}$ plane. It is defined by the mechanical structure of the scanning mirror and it is fixed, ie. θ_t^{sr} is constant. The azimuth of the target ψ_t^{sr} is obtained from the attitude-stabilizing servo controllers as the position of the scanning mirror at the time when the radar signal is acquired. Equation 9 shows the relation between \mathbf{P}_t^{sr} and \mathbf{r}_t^{sr} .

$$\mathbf{P}_t^{sr} = \begin{bmatrix} r_t^{sr} \cos \theta_t^{sr} \cos \psi_t^{sr} \\ r_t^{sr} \cos \theta_t^{sr} \sin \psi_t^{sr} \\ r_t^{sr} \sin \theta_t^{sr} \end{bmatrix} \quad (9)$$

The position of the UAV body in the earth coordinate \mathbf{P}_b^e , is provided by the Inertial Navigation System (INS). The position of the MMW radar sensor in the UAV body \mathbf{P}_{sr}^b is determined by UAV design and sensor calibration.

7 Feature Extraction

Various methods for target detection exist, as described in [2][12][13]. Polarisation information can be used to discriminate radar targets from the clutter of the terrain. The use of MMW radar polarisation data for autonomous land vehicle navigation is addressed in [2]. [13] addresses the use of a "down-range interpreter" function to form an evidence map for three dimensional map building with MMW radars.

Constant False Alarm Rate (CFAR) is one well established and reliable method. A simple thresholding has also been demonstrated with satisfactory results particularly in controlled environments [2].

In this application, two independent returns are expected; one from the ground directly below the aircraft produced by the altimeter horn shown in Figure 3, and one from the scanned beam. Detection of the altimeter signal is easily achieved using a standard CFAR process based on the noise statistics of the radar receiver as this echo will be isolated in range.

Detection of radar target features received via the scanned beam will be more complicated as they will be embedded within the ground echo return which is statistically quite complex. However, because of the radars good range resolution, the magnitude of the clutter return (reflections from the ground) into each frequency bin output by the FFT process will be quite small. A CFAR process that analyses the statistics of this ground echo will produce a suitable threshold to maximise the detection probability and minimise the false alarm probability of a moderately sized corner reflector from a single observation.

Multiple, sequential observations of the same area can be used to enhance the contrast between the corner reflector and that of the surrounding clutter if the observation period is sufficiently long to ensure complete (or even partial) decorrelation of the clutter returns.

The large elevation footprint coupled with uncertainties in the radar height and the slope of the terrain result in a coupling between the uncertainty in the range and elevation measurements from the radar. This is mitigated by measuring the radar height using an altimeter (as discussed) and measuring the extent of the footprint of the scanned beam to determine the slope.

8 Conclusion and Future Work

This paper has presented the compact MMW radar sensor developed for the Brumby Mk III UAV. The system integration tests of the MMW radar sensor, DSP, attitude stabilizing gimbal, and the radar control computer with the navigation computer of the UAV are performed in RMUS environment.

The ground targets consisting of low-cost corner reflectors are successfully detected in the ground tests. Target range data measured with the MMW radar sensor matched with the range data read from the laser range sensor. Flight tests with the MMW radar sensor are scheduled for late 2003.

Acknowledgements

The authors wish to thank BAE Systems for their support in providing funding this project.

References

1. Clark Stephen, "Autonomous Land Vehicle Navigation Using Millimeter Wave Radar," *University of Sydney*, PhD Thesis, 1999.
2. S.Clark and Whyte, H.Durrant, "The Design of a High Performance MMW Radar System for Autonomous Land Vehicle Navigation," *International Conference on Field and Service Robotics*, Canberra, Australia, pp. 292–299, 1997.
3. H. Durrant-Whyte, S Majumder, M D Battista, and S Scheduling, "A Bayesian Algorithm for Simultaneous Localisation and Map Building," *10th International Symposium of Robotics Research - ISRR 2001*, Lorne, Victoria, Australia, 2001.
4. Göktoğan, Ali Haydar and Nettleton, Eric and Ridley, Matthew and Sukkarieh, Salah, "Real Time Multi-UAV Simulator", *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.
5. G. Dissanayake, H. Durrant-Whyte, and T. Bailey, "A Computationally Efficient Solution to the Simultaneous Localisation and Map Building (SLAM) Problem," *IEEE International Conference on Robotics and Automation 2000*, San Francisco CA, 2000.
6. Sukkarieh, Salah and Göktoğan, Ali Haydar and Kim, Jong-Hyuk and Nettleton, Eric and Randle, Jeremy and Ridley, Matthew and Wishart, Stuart and Durrant-Whyte, Hugh, "Cooperative Data Fusion and Control Amongst Multiple Uninhabited Air Vehicles," *ISER 2002 Seventh International Symposium on Experimental Robotics*, 20002.
7. Nettleton, Eric W. and Durrant-Whyte, Hugh F. and Gibbens, Peter W. and Göktoğan, Ali H., "Multiple Platform Localisation and Map Building," *Sensor Fusion and Decentralised Control in Robotic Systems III*, Boston, USA, vol. 4196, pp. 337–347, 2000,
8. Sukkarieh, Salah and Nettleton, Eric and Kim, Jong-Hyuk and Ridley, Matthew and Göktoğan, Ali and Durrant-Whyte, Hugh, "The ANSER Project-Multi UAV Data Fusion," *International Journal of Robotics Research*, 2002.
9. Brooker, G. and Bishop, M. and Scheduling, Steve, "Millimetre Waves for Robotics," *Australasian Conference on Robotics and Automation*, Sydney, 2001.
10. Scheduling, S. and Brooker, G. and Hennessey, R. and Bishop, M. and Maclean, A., "Terrain Imaging and Perception using Millimetre Wave Radar," *Australasian Conference on Robotics and Automation*, Auckland, 2002.
11. Scheduling, S. and Brooker, G. and Bishop, M. and Hennessey, R. and Maclean, A., "Terrain Imaging Millimetre Wave Radar," *International Conference on Control, Automation, Robotics & Vision*, Singapore, 2002.
12. Currie, Nicholas C. and Brown, Charles E., "Principles and Applications of Millimeter-Wave Radar," *Artech House, Inc.*, Norwood, Massachusetts, 1987.
13. Alex Foessel and John Bares and William Red L. Whittaker, "Three-Dimensional Map Building with MMW Radar," *Proceedings of the 3rd International Conference on Field and Service Robotics*, Helsinki, Finland, June 2001.

Motion Analysis of a Parallel Mobile Robot

Shraga Shoval¹ and Moshe Shoham²

¹ Department of Industrial Engineering & Management,
Academic College of Judea and Samaria, Ariel, Israel
shruga@yosh.ac.il

² Faculty of Mechanical Engineering,
Technion, Haifa, Israel,
shoham@tx.technion.ac.il

Abstract. This paper presents a kinematic and force analysis of a mobile robot built on the principle of parallel mechanisms. The robot consists of an upper plate connected to 3 legs, each equipped with an asynchronous driving unit. A kinematic model for the robot provides data for accurate position estimate, even in rough and slippery terrains where conventional odometry fails. The paper presents an analysis of the forces acting on the robot under various surface conditions and robot configurations. This analysis provides useful data to determine whether a specific motion can be completed given the limitations on stability, the geometry and friction of the surface, and the required motion direction. The paper presents simulation results that are verified by experiments using our prototype model.

1 Introduction

Parallel mechanisms consist of an upper platform that is maneuvered by several (3-6) legs connected to a lower stationary platform. The maneuverability, rigidity and accuracy are functions of the number of legs and the type of joints between the plates and the legs. The basic conceptual mechanics is known as the Stewart platform [Stewart, 1965] even though earlier versions are known, and since then many manipulators were developed based on this mechanism [Hunt 1983, Tsai and Tahmasebi 1983 and others].

Ben Horin and Shoham [1997] have suggested using mobile joints between the legs and the stationary platform, turning the mechanism into a semi-mobile robot. The mechanism consists of the following components: three links of fixed length, having a spherical joint on one end and a revolute joint on the other end, three actuators which move plenary on a stationary platform and an output platform having six degrees-of-freedom (DOF). To further increase mobility, Ben Horin and Shoham [1999, 2000] suggest a more flexible design. This mechanism, shown in Fig. 1, is based on 3 inflatable legs, an upper platform and 3 asynchronous driving units for the three legs. The upper joint of each leg is a revolute joint, while the lower joint, which connects the leg to the driving unit, is a spherical joint. This configuration offers six DOF for the upper plate where the control parameters are the positions (X,Y,Z) of the three driving units. The mechanism is designed for applications that require a light weight and easy

deployable robot. Given the required trajectory for the end effector (6 parameters) the inverse kinematics model can generate the required path of each driving unit, subject to its non-holonomic constraints.



Fig.1. The inflatable mobile robot [Ben Horin and Shoham, 2000]

In this paper we present a kinematic and force analysis for the parallel mobile robot. The kinematic analysis, shown in section 2, enables an improved position estimate based on additional encoders attached to the upper platform. Section 3 details the force analysis for the robot, and section 4 shows simulation results for motions in various configurations and constraints. Section 5 verifies the simulation results with experiments conducted with our prototype platform and section 6 provides concluding remarks.

2 Kinematic Analysis

Figure 2 is a schematic description of the parallel mobile mechanism. The upper plate is connected to each of the three legs with revolute joints. The three legs are driven by three asynchronous units that are connected to the legs with spherical joints. Controlled motion of the three driving units determines the pose (position and orientation) of the upper plate.

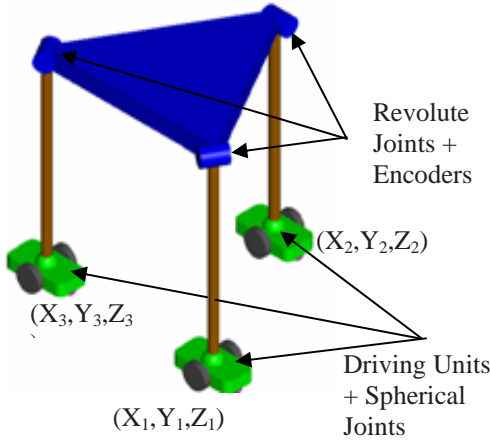


Fig.2. Schematic description of the mobile parallel mechanism

To determine the accurate configuration of the upper plate, both the absolute position of each driving unit (X_i, Y_i, Z_i) and the direct kinematics model are required. The position of the driving units is determined by Odometry using encoders attached to the driving wheels. Tahmasebi and Tsai [1994] show that for the above parallel mechanism there are 16 possible direct kinematics solutions, which require extensive computational effort. Furthermore, determining the absolute position of each driving unit is subjected to odometric errors and cannot provide a reliable position estimate. To improve the accuracy and to simplify the direct kinematics model Shoval and Shoham [2001] use additional measurements taken from on board encoders attached to the upper revolute joints. The additional encoders measure the rotation angle between the upper plate and the legs. Based on these measurements, the positions of the three driving units are derived in the U-V-W coordinate system (attached to the center of the upper plate). Based on the position of the drive units as determined in the U-V-W system, the Euclidean distances between the drive units l_1 , l_2 and l_3 are given by Eq. 1:

$$l_i = \sqrt{(U_{i-1} - U_{i+1})^2 + (V_{i-1} - V_{i+1})^2 + (W_{i-1} - W_{i+1})^2} \quad (1)$$

i – in cyclic order

These Euclidean distances can also be derived according to the position of the driving units as determined by the Odometric model according to Eq. 2.

$$m_i = \sqrt{(X_{i-1} - X_{i+1})^2 + (Y_{i-1} - Y_{i+1})^2 + (Z_{i-1} - Z_{i+1})^2} \quad (2)$$

If the odometric system is accurate, the distances derived in the upper plate coordinate system (l_i) are equal to the distances derived in the world coordinate system (m_i). If, however, these distances are different, the odometric calculation is faulty. An odometric error in one drive unit affects two distances according to Eq. 2. Assuming a single odometric error at each interval time, the accurate

position of three driving units is continuously updated. In the unlikely event that two (or three) driving units are subjected to odometric errors simultaneously, the proposed procedure cannot be implemented and additional measures must be taken (i.e. re-calibration of the robot's position).

3 Force Analyses

Fig. 3 shows the contact forces applied on the robot by the ground. Do and Yang [1988] suggest a solution based on the Newton-Euler method, resulting in 36 linear equations. Other researchers use the principle of virtual work to reduce the complexity of the solution. Ben Horin [1999] uses the Kane method to determine the dynamic equations, solving it with numerical procedures. To simplify the solution we assume that the robot's mass is concentrated at the upper plate, while the legs have negligible mass. We also assume that internal changes in the robot internal configuration are quasistatic (dynamic forces associated with changes of internal configuration are negligible compared with other).

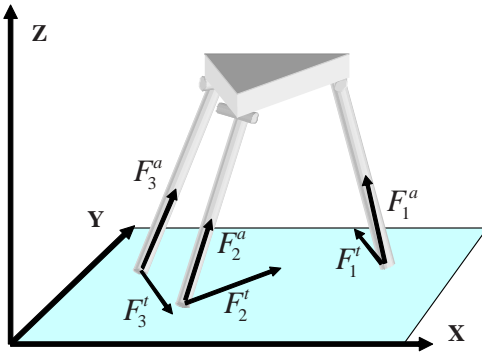


Fig.3. Contact forces applied on the robot

The revolute joints between the legs and the upper plate sustain only axial force - F^a acting along the leg, and tangential force - F^t acting parallel to the axis of the revolute joint. Reducing the problem to 6 unknown forces simplifies the solution to the following equations:

$$\sum F_x = \sum_{i=1}^3 F_i^a \cos \theta_i \cos \mu_i + \sum_{i=1}^3 F_i^t \cos \delta_i \cos \gamma_i = m a_x \quad (3)$$

$$\sum F_y = \sum_{i=1}^3 F_i^a \cos \theta_i \sin \mu_i + \sum_{i=1}^3 F_i^t \cos \delta_i \sin \gamma_i = m a_y \quad (4)$$

$$\sum F_z = \sum_{i=1}^3 F_i^a \sin \theta_i + \sum_{i=1}^3 F_i^t \sin \delta_i = m a_z \quad (5)$$

$$\sum M_x = \sum_{i=1}^3 \{ (F_i^a \cos \theta_i \sin \mu_i + F_i^t \cos \delta_i \sin \gamma_i) (r_i^b - r^c)_z - (F_i^a \sin \theta_i + F_i^t \sin \delta_i) (r_i^b - r^c)_y \} = j \ddot{\theta} \quad (6)$$

$$\sum M_y = -\sum_{i=1}^3 \{ (F_i^a \cos \theta_i \cos \mu_i + F_i^t \cos \delta_i \cos \gamma_i) (r_i^b - r^c)_z + (F_i^a \sin \theta_i + F_i^t \sin \delta_i) (r_i^b - r^c)_x \} = j \ddot{\xi} \quad (7)$$

$$\sum M_z = -\sum_{i=1}^3 \{ (F_i^a \cos \theta_i \sin \mu_i + F_i^t \cos \delta_i \sin \gamma_i) (r_i^b - r^c)_x + (F_i^a \cos \theta_i \cos \mu_i + F_i^t \cos \delta_i \cos \gamma_i) (r_i^b - r^c)_y \} = j \ddot{\omega} \quad (8)$$

where

F_i^a - the axial force in leg i ,

F_i^t - the tangential force in leg i .

θ_i - the angle between leg i and the X-Y plane (therefore the angle between the axial force and the X-Y plane).

μ_i - the angle between leg i and the positive X axes.

δ_i - the angle between the tangential force of leg i and the X-Y.

γ_i - the angle between the tangential forces of leg i and the positive X axes.

r_i^b - the position vector of the bottom of leg i .

r^c - the position vector of center of the upper plate.

a_x, a_y, a_z - accelerations along the X, Y and Z

θ, ω, ξ - Euler orientation angles

J - Moment of inertia of the upper plate

m - Mass of the upper plate

Let us assume a friction coefficient μ between the driving wheels and the surface. We transform the axial - F_i^a and tangential - F_i^t forces (derived by Eq. 3-8) to a new coordinate set defined by: \hat{u}_d - in the direction of the required motion, \hat{u}_l - in the lateral direction, and \hat{u}_n - normal to the surface. The lateral force F_l acting on the driving unit is given by:

$$F_l = (F_i^a \bullet \hat{u}_l + F_i^t \bullet \hat{u}_l) \quad (9)$$

To prevent lateral slippage, F_l is bounded by:

$$F_l < \mu (F_i^a \bullet \hat{u}_n + F_i^t \bullet \hat{u}_n) \quad (10)$$

The two expressions associated with F_l provide the first condition for preventing lateral slippage:

$$\frac{(F_i^a \bullet \hat{u}_n + F_i^t \bullet \hat{u}_n)}{(F_i^a \bullet \hat{u}_l + F_i^t \bullet \hat{u}_l)} < \mu \quad (11)$$

Given a specific moment M_d generated by the driving motor and r - the radius of the driving wheel, the friction driving force - F_d generating the motion of the driving unit in the required direction is limited by:

$$F_d = M_d r < \mu (F_i^a \bullet \hat{u}_n + F_i^t \bullet \hat{u}_n) \quad (12)$$

Given a specific terrain topography and friction, the **actual** driving forces applied on each leg can be determined both along the longitudinal and lateral directions. Furthermore, based on these forces the **accurate** dynamic reaction of the robot can be calculated.

4 Simulation Results

We first examine the forces applied on the robot when traveling at a constant speed on a horizontal surface, with an identical inclination angles (η) between the legs and the upper plate. The robot is traveling in the positive Y direction with a configuration is shown in Fig. 5a. We start the experiment with $\eta=90^\circ$ (legs are perpendicular to the surface and the upper plate), while gradually and simultaneously reducing the inclination angles for all legs. The weight of the upper plate is 100N. As expected, the axial forces - F_i^a on all three legs are identical, starting with 33.33N when the legs are perpendicular ($\eta=90^\circ$), and increasing as η decreases (Figure 5b). Since all driving units are in the travel direction, the tangential forces - F_i^t are close to zero in all legs, and therefore are not shown in the graphs.

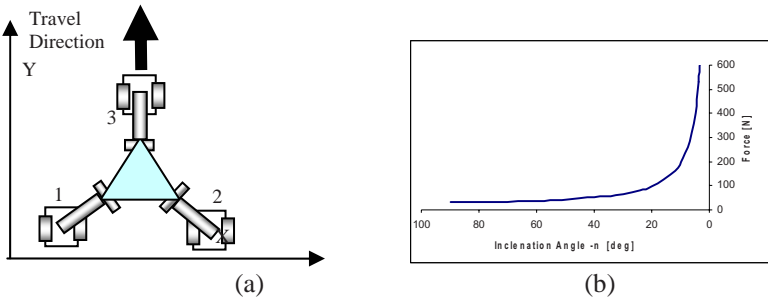


Fig.5. Axial forces for constant motion over a horizontal surface.

Next, we transform the axial and tangential forces to the corresponding components in the longitudinal (F_d), lateral (F_l), and normal to the surface directions (F_n). Due to the symmetry, the normal components in all legs remain constant (33.33N). The friction coefficient is 0.7, generating a maximal friction force of 23.33N for all legs. Fig. 6a shows the lateral and longitudinal forces for legs 1 and 2. These forces are identical due to the symmetry of the two legs relative to the driving direction. Fig. 6b shows the same forces for leg 3. As shown, the lateral forces on legs 1 and 2 pass the maximal friction force at $\eta=53^\circ$. The longitudinal force of leg 3 passes the friction force limit at $\eta=55^\circ$. The robot can therefore travel at a constant speed on a horizontal surface with a friction coefficient of 0.7 as long as the inclination angle of the legs is larger than 55° . At that angle, longitudinal slippage occurs at leg 3. Further decrease of the inclination angle to 53° causes additional lateral slippage in legs 1 and 2.

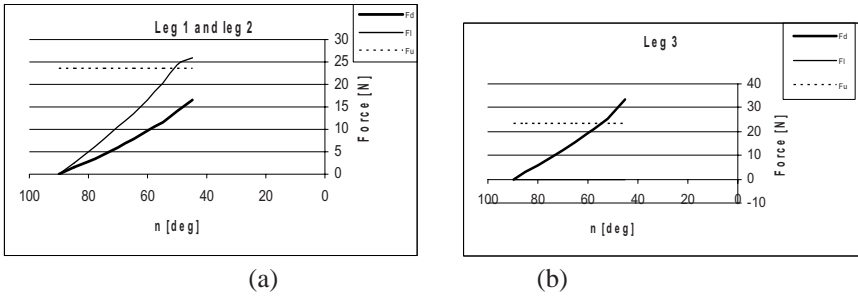


Fig.6. Forces on legs 1 and 2 (a) and 3 (b) for constant speed.

In the next set of simulations, the robot travels along a horizontal surface in a straight line with constant acceleration along the $Y+$ axis (equivalent to traveling at a constant speed on an inclined surface). In addition to friction constraints, external stability must also be considered. Fig. 7 shows the forces on legs 1 and 2 as a function of the inclination angle η during a 5m/sec^2 (equivalent to traveling on an inclined surface of 30°). The results indicate that legs 1 and 2 lose contact at inclination angles larger than 78° (shown as a negative friction force). However, the lateral force is larger than the friction limit for all inclination angles, resulting in lateral slippage for any internal configuration. In order to complete a stable motion at a 5m/sec^2 the inclination of the “front” leg (leg3) must be reduced, as shown in Fig. 7b. This change adjusts the force distribution, similar to humans climbing a steep hill. The new internal configuration enables stable motion as long as inclination angle for leg 3 is in the range of 68° - 53° , and legs 1 and 2 are larger than 78° .

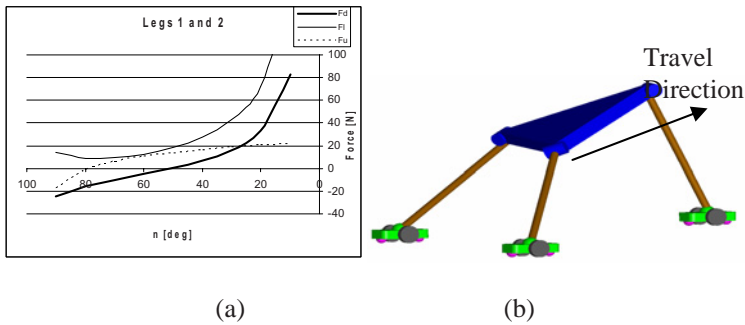


Fig.7. Accelerated motion of 5m/sec^2 .

Similar results are obtained for circular motion. Again, symmetric internal configuration results either in tipover or immediate slippage in one or more legs. Asymmetric internal configuration enables the robot to safely complete the required motion even for sharp curvatures with relatively high speeds.

5 Field Experiments

To verify the simulation results we have conducted field experiments using our inflatable mobile platform shown in Fig. 1. In these experiments we measured the stability limits for various internal. The results indicate close match between the theoretical simulations and the field experiments. For example, stability limit for horizontal surface with symmetric configuration (shown in Figs. 5-6) is obtained for inclination angle of 60° , compared with 53° determined in the simulation. Motion over inclined surface of 30° is stable when inclination angle for leg3 is in the range of 65° - 55° compared with 68° - 53° determined in the simulation.

6 Conclusions

A new design for a parallel mobile robot is presented. The robot consists of three legs, each driven by an asynchronous mechanism connected to the legs with a spherical joint. Each leg is connected to an upper platform with revolute joint and additional encoders, measuring the revolute angle of the upper joints. These encoders provide data used by the kinematic model for early detection and correction of positioning errors generated by odometry. Early detection and correction of odometric errors in each leg prevent accumulation of significant errors of the upper plate, and can identify irregularities on the surface.

A simplified dynamic model provides a solution for the forces applied on the robot. The model determines whether a specific task can be reliably performed, given a specific surface topography and friction. The model can also detect instabilities either by losing contact with the ground (tipover), or by slippage (longitudinal or lateral). An unstable configuration can be avoided by changing the inclination angles between the legs and the upper plate. This feature allows the robot to complete motions in complex terrains where conventional robots cannot maintain stability due to inertial forces, surface topography, or friction constraints.

References

1. Ben Horin (Dombiak) P., 1999, Analysis and Synthesis of an Inflatable Parallel Robot", *M.Sc. Thesis, Technion*, Haifa.
2. Ben Horin (Dombiak) P., Shoham, M., and Grossman, G., "A Parallel Six Degrees of-Freedom Inflatable Robot," *ASME 2000 Mechanism and Robotics Conference*, Washington, 2000.
3. Ben Horin R., "Criteria for Analysis of Parallel Robots", *D.Sc. Thesis, Technion*, Haifa, 1997.
4. Do W. Q. D., Yang D. C. H., "Inverse Dynamics Analysis and simulation of a Platform Type of Robot", *Journal of Robotic Systems*, Vol. 5, No. 3, pp. 209-227, 1998.
5. Hunt K. H., 1983, "Structural Kinematics of In-Parallel-Actuated Robot Arm", *ASME Journal of Mechanisms Transmissions and automation in Design*, Vol. 105, pp. 705-712.

6. Shoval S. and Shoham M. "A Redundant Parallel Mobile Mechanism", *Proceedings of IEEE International Conference for Robotics and Automation*, May 2001, Seoul, Korea, pp. 2273-2279.
7. Stewart D., 1965, "A platform with six Degrees of Freedom", *Proceedings of Institute of Mechanical Engineering, London England*, Vol. 180, pp. 371-386.
8. Tahmasebi F., Tsai L. W., 1994, "Closed-Form Direct Kinematics Solution of a New Parallel Minimanipulator", *Transactions of the ASME*, Vol. 116, pp. 1141-1147.
9. Tsai L. W., Tahmasebi F., 1983, "Synthesis and Analysis of a New Class of Six Degree-of-Freedom Parallel Minimanipulators", *Journal of Robotic Research*, Vol. 10, pp. 561-580.

Teleoperation System for Two Tracked Mobile Robots Transporting a Single Object in Coordination Based on Function Allocation Concept

Hiroki Takeda¹, Zhi-Dong Wang², and Kazuhiro Kosuge²

¹ Dept. of Machine Intelligence and Systems Engineering,
Tohoku University
takeda@irs.mech.tohoku.ac.jp

² Dept. of Bioengineering and Robotics, Tohoku University
wang.kosuge@irs.mech.tohoku.ac.jp
<http://www.irs.mech.tohoku.ac.jp>

Abstract. In this paper, we propose a collision avoidance algorithm for two nonholonomic tracked mobile robots transporting a single object based on a function-allocation concept. In this algorithm, a leader robot is controlled manually. A follower robot estimates the desired trajectory of the leader along its own heading direction and generates the motion for transporting the object and for avoiding obstacles by using an omni-directional vision sensor. We experimentally implement the proposed algorithm in a tracked mobile robots system, and illustrate the validity of the proposed control algorithm.

1 Introduction

A trailer, which is transporting a large size container, has steering mechanisms in both the front wheel and the rear wheel to advance the motility of the trailer. Usually two operators drive this type of the trailer and each operator controls each steering. However, long experiences are needed to drive such a trailer efficiently especially when the trailer circles. In this paper, we consider this trailer system as the two mobile-robot system that transports a single object in coordination. We will discuss the motion control method of this system. In general, we control directly one of the robots and the other robots are controlled autonomously to maneuver a multiple robot system by an operator. If we control this system by this way, the operator has to determine the control input not only for the controlling robot itself but also for the whole system. This makes the controllability be declined. In the proposed system, the effective task could be realized by allocating the functions which are necessary to realize the task. Based on different concept on function distributed and information management, many researchers have proposed various motion control algorithms for the multiple mobile robots to handle a single object in coordination [1]-[6] etc. We briefly review some of the multi-robot systems which execute the coordinate task.

N.Miyata et al. have proposed the control algorithm transporting a single object by multiple nonholonomic car-like robots based on the function allocation concept [4]. In this algorithm, several functions for achieving tasks are allocated to

each robot and multiple mobile robots realize the tasks effectively. To realize this system, however, each robot is controlled in the centralized control system to share the information of other robots.

Kosuge et al. proposed a leader-follower type of motion control algorithm for nonholonomic tracked mobile robots to transport a single object[8]. In this algorithm, a motion command of the object is given to one of the robots referred to as a leader. The rest of the robots referred to as followers estimate the motion of the leader through the motion of the object and transport it together with the leader in coordination. Because the robots do not use explicit communication, the execution of more reliable transport task is expected. In addition, authors proposed a collision avoidance algorithm for two-tracked mobile robots[9]. In this system, the motion command of the leader is given in advance in this control algorithm. Then we have to design the whole trajectory in advance when we apply this control algorithm. In this paper, we expand the method of [9] and apply to dynamic environment such as the slip of wheels could not be neglected.

2 Basic Leader-Follower Algorithm

In this section, we briefly explain the decentralized control algorithm proposed in [8]. In case of the mobile robots under a nonholonomic constraint, we assume that each robot holds the object through a free rotational joint, which is located on the axis of both wheels, as shown in Fig. 1. In this case, the motion of the robot is characterized by two kinds of motion. One is the translational motion along the heading direction of the robot, and the other is the rotational motion around the free rotational joint.

In the algorithm proposed in [8], each robot was controlled to have a following dynamics along the heading direction.

$$D {}^l\dot{x}_l + K {}^l x_l = {}^l F_{lx} \quad (1)$$

$$D {}^1\dot{x}_1 + K {}^1 x_1 = {}^1 F_{1x} \quad (2)$$

where D is a positive definite damping coefficient and K is a positive definite stiffness coefficient. ${}^l x_l$ and ${}^1 x_1$ are the motion deviations of the leader and the follower according to the forces applied to each robot ${}^l F_{lx}$, ${}^1 F_{1x}$, respectively. A desired trajectory of the object is given to the leader and the follower estimates its

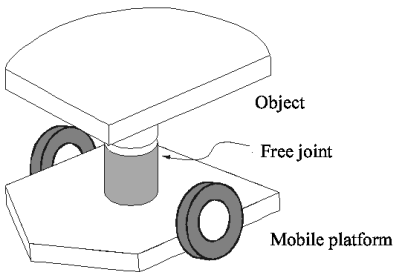


Fig. 1. Holding mechanism applied to both leader and follower robot, which has nonholonomic constraints, to transport a single object in coordination.

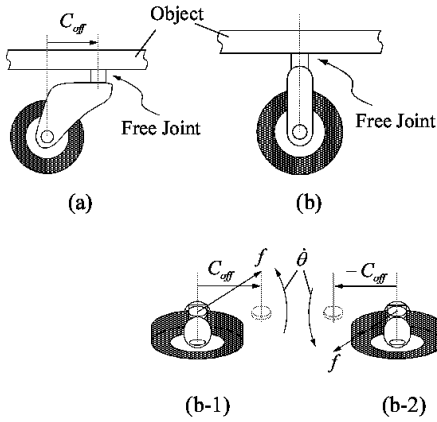


Fig. 2. (a) General view of the real caster model which has fixed C_{off} , (b) Proposed dual-caster model. The direction of the offset of the caster C_{off} is changed to minimize the rotational angle ((b-1)(b-2)).

motion along the heading direction of the follower by using the estimation algorithm proposed in [8]. For the rotational motion of the robot, each robot is controlled as if it has a caster-like dynamics as shown in Fig. 2. The follower makes its orientation rotate to the heading direction of the object based on the caster-like dynamics. Let us briefly review this dynamics.

A real caster always has an offset C_{off} between the axis of the wheel and the free rotational joint as shown in Fig. 2(a). The caster turns to the direction of the force applied to the caster by this offset. If the robot is controlled to imitate the motion of the real caster directly, the robot will rotate more than 90 degrees, when the robot is pulled backward. In the cooperative object transportation, the excessive force will be generated between robots during rotational motion more than 90 degrees of the follower. To avoid this problem, we propose that the follower is controlled to have two different caster dynamics as shown in Fig. 2(b). That is, we consider changing the position of the free rotational joint according to the force direction applied to the robot. When the robot is pushed forward, the offset is set equal to $C_{off} (> 0)$ (Fig. 2(b-1)), and when the robot is pulled backward, the offset is set equal to $-C_{off}$ (Fig. 2(b-2)). We refer to this caster-like dynamics as the dual-caster action.

3 Function Allocation

In this research, the task is to control two tracked mobile robots which is transporting a relative large object in an environment with obstacles from a remote site. The robots system should have two functions, transporting the object in coordination and avoiding collision between obstacles and the transportation system including robots and the object. In this research, we allocate the two types of function to each robot. One is to control the position of the object and the other is to control the orientation of the object and to transport the object in coordination. We allocate the position control function of the object to the leader, so that the leader transport the object

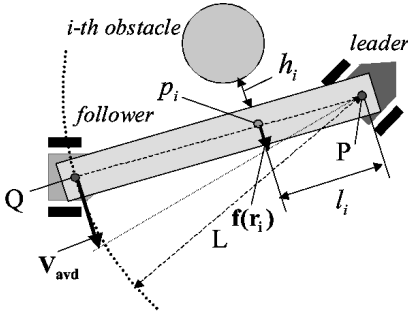


Fig. 3. Avoidance velocity vector of the follower V_{avd} is calculated based on h_i , the minimal distance between the obstacle and the object to avoid the obstacle i .

based on the joystick control of the leader. We also allocate the orientation control function of the object and the transport function with the other robot to the follower, so that the follower transports a single object with the leader.

The collision avoidance problem here is similar with the case of a multi-trailers system. In a relatively simple environment, it may be possible to avoid collision between an obstacle and transported object by only controlling path of the leader if the operator has enough knowledge of both the obstacle and characteristics of the follower's motion. However, it becomes very hard if the environment is complicated and the operator does not have enough information of the follower's motion. Actually, in the proposed leader-follower system, the orientation of the follower depends on the force applied to the follower and it is difficult to estimate the motion of the follower in advance.

In this research, we solve the problem by controlling the leader on its collision free path and allocating the obstacle avoidance function of both the transported object and the follower to the follower. When the leader is moving on a collision free path, the collision between the object and obstacles could be avoided by controlling the orientation of the object, which is depending on the follower's motion. This can be realized by introducing an avoidance velocity vector to the follower's motion control (Fig. 3). In [9], we proposed a control system for achieving the cooperative transporting task in a static environment based on this function allocation concept. In the proposed system, a fixed trajectory is given to the leader in advance and the follower's collision motion is based on a static map of the environment, which involves the position and the geometrical information of the obstacles. However, when we apply this system to more general environment, we have to construct the map each time. Furthermore, static maps information is not enough if there has large slips of the wheels of the robots or the position of the obstacles are changed during the execution of the transportation task. In this paper, we use the camera installed on the leader to provide environment information in front of the leader to the operator. We also utilize an omni-directional vision sensor to get the information of the environment not only around the robot but also around the object for the avoidance motion of both the object and the follower in the dynamic environment. The omni-directional vision sensor is installed on the top of the follower's rotational center.

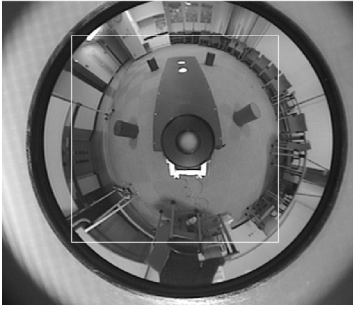


Fig. 4. Omni-directional image of the follower during object transportation. The object is projected as a dark trapezoid sharp area from the image center. The leader is supporting the front part of the object which is indicated by two white marks on the object.

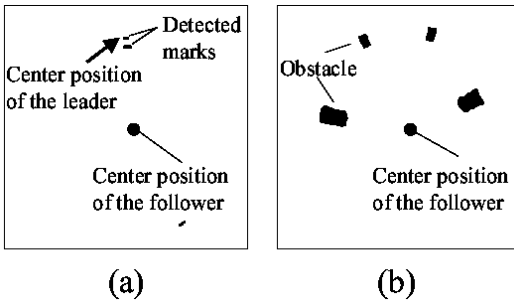


Fig. 5. Processed omni-directional image shown in Fig. 4: (a) detected marks of the leader. (b) detected obstacles.

Processed data of the omni-directional image, which are including the position of the obstacles and the orientation of the object, are utilized for the motion control of the follower. In this system, the operator only needs to take care of the leader's obstacle avoidance, which is a reasonable easy task for the operator from the remote site.

4 Control of Robots

4.1 Control of Leader

The operator controls the leader's motion based on the information from a camera installed on the front side of the leader. The leader is controlled by a 2-DOF joystick. This joystick has the translational velocity input and the angular velocity input. The control commands of the joystick are sent through the wireless Ethernet in our laboratory.

4.2 Control of Follower

Detecting of object In [9], the object is modeled as the line $|PQ|$, which consists of the free rotational joint of the robots, as shown in Fig. 3. In addition, the orientation of the object is decided by utilizing the data of the force sensor which is mounted on the follower. The follower calculates the avoidance motion V_{avd} based on the

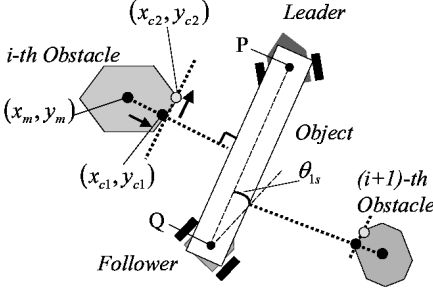


Fig. 6. determination of the nearest point on each obstacle by referring the pixel data of the obstacle's area.

minimum distance between the i -th obstacle and the object which is described as h_i . In this case, the follower knows the position of a point P and the position of the obstacles. In this paper, we model the object and get the orientation of the object by processing the two white marks on the object, one of which is on the position of the free joint of the leader as shown in Fig. 4. This image is digitized to distinguish the marks as shown in Fig. 5(a). This makes the sensing more stable comparing to the previous method based on the output of the force sensor.

Detecting obstacles In the digitized image from the omni-directional vision sensor, the area of the obstacle is extracted by utilizing the area filter(Fig. 5(b)). Each obstacle is labeled and the center position of the obstacle is calculated as (x_m, y_m) . Next the value of the pixels on perpendicular line of the object from the point (x_m, y_m) are referred. Then the point (x_{c1}, y_{c1}) which is a point of the edge of the area of the obstacle can be decided. The follower refers the value of the pixel along to the orientation of the object from (x_{c1}, y_{c1}) and decides the nearest position of the obstacle (x_{c2}, y_{c2}) as shown in Fig. 6. We generate the minimum distance between the modeled object(line $|PQ|$) and the point (x_{c2}, y_{c2}) .

Avoidance motion of follower The avoidance velocity vector V_{avd} is generated as follows[9].

$$V_{avd} = \sum_{i=1}^n v(r_i) \cdot \frac{L}{l_i} \quad (3)$$

$$v(r_i) = \frac{k}{|r_i|^2} r_i \quad (4)$$

where l_i is the distance from the grasping point of the leader to the point p_i , and k is a constant. From eqs.(3) and (4), the follower could avoid obstacles by specifying the avoidance velocity vector V_{avd} perpendicular to the heading direction of the object. For the rotational motion around the free rotational joint of the follower, we consider that how the follower generates the angular velocity of the follower around its free rotational joint by using the velocity along the heading direction of the follower and

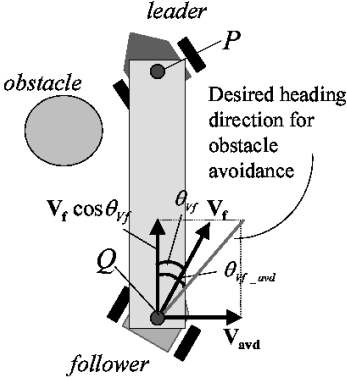


Fig. 7. The motion direction of the follower is derived by the velocity vector of the object, which is calculated by estimation of the leader's motion V_f , and the avoidance velocity vector V_{avd} , which is generated based on the minimum distance among the obstacles and the object.

the avoidance velocity V_{avd} . Let P and Q be the positions of the rotational joints of the leader and the follower respectively as shown in Fig. 7. Let V_l be the velocity of the leader at the point P , and V_f be the velocity of the follower at the point Q . Let θ_{V_f} be the angle between the vector from P to Q and V_f . From Fig. 7, the velocity along the heading direction of the object is $V_f \cos \theta_{V_f}$. The desired orientation of the follower is also calculated as follows.

$$\theta_{V_f-avd} = \tan^{-1} \frac{V_{avd}}{|V_f \cdot \cos \theta_{V_f}|} \quad (5)$$

Let us consider how the follower aligns its orientation to the direction calculated in eqs.(5). We calculate the force \tilde{f}_{1x} along the direction of θ_{V_f-avd} and the force \tilde{f}_{1y} perpendicular to the direction of θ_{V_f-avd} as following equation based on the magnitude of the force f applied to the follower as shown in Fig. 2.

$$\tilde{f}_{1x} = f \cos(\theta_{V_f-avd}) \quad (6)$$

$$\tilde{f}_{1y} = f \sin(\theta_{V_f-avd}) \quad (7)$$

Then we apply \tilde{f}_{1x} and \tilde{f}_{1y} to the dual-caster action as follows;

$$D_{cast} \dot{\theta}_1 = C \tilde{f}_{1y} \quad (8)$$

$$C = \begin{cases} C_{off}, & \text{for } \tilde{f}_{1x} \geq 0 \\ -C_{off}, & \text{for } \tilde{f}_{1x} < 0 \end{cases} \quad (9)$$

where D_{cast} is the damping coefficient of the dual caster and C is the offset of the dual caster as shown in Fig. 2. By specifying the velocity along the heading direction of the follower calculated by the estimation algorithm expressed in section 3 and the angular velocity of the follower around its free rotational joint expressed in this section, the follower could transport a single object in coordination without colliding with obstacles.

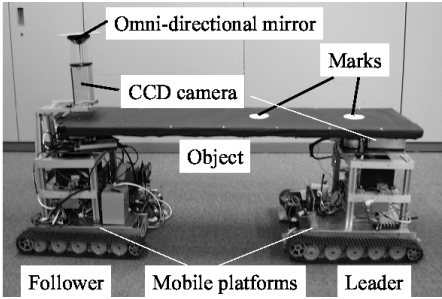


Fig. 8. Experimental system: View of the robots.

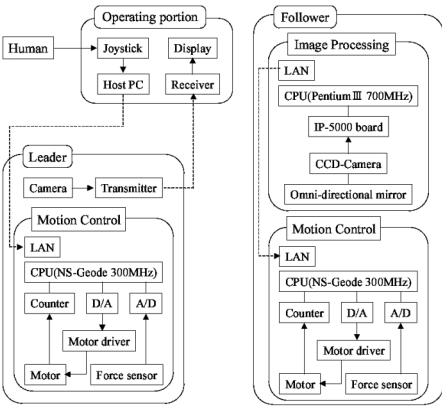


Fig. 9. Experimental system: Hardware profile.

5 Experiments

The proposed control algorithm, which is applied to the transporting task in coordination in the dynamic environment, was implemented in the experimental system, which consists of two tracked mobile robots as shown in Fig. 8. Each robot has a force sensor at its rotational joint and it is controlled by its on-board controller. Each robot is powered by rechargeable battery and connected to the network system of our laboratory through a wireless Ethernet. The detail of the robot system is described as Fig. 9. The leader is controlled manually by the 2-DOF joystick and control signal is sent by the wireless Ethernet. The follower was controlled by using the proposed algorithm in the previous section.

The results are shown in Figs. 10– 11. Fig. 10 shows the pathes of the leader controlled by the human operator and the follower using the proposed algorithm respectively and the snap shot of the experiments. Fig. 11 shows the translational and angular velocity of the leader. It is evident that the leader realizes the motion instructed by the joystick input. As shown in Figs. 10– 11, the transportation task was successfully done without colliding with the obstacles. In this experiment, the follower detects the obstacles which are coming close to the object or itself and generates the avoidance motion. Then we can see that the leader and the follower could execute the transportation task successfully.

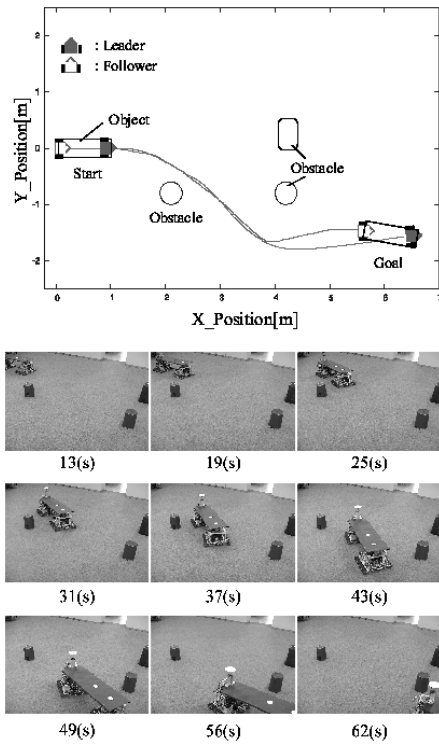


Fig. 10. Experimental results: real trajectory of the leader controlled by the human operator and the follower using the proposed algorithm(above) and the view of the experiments(below).

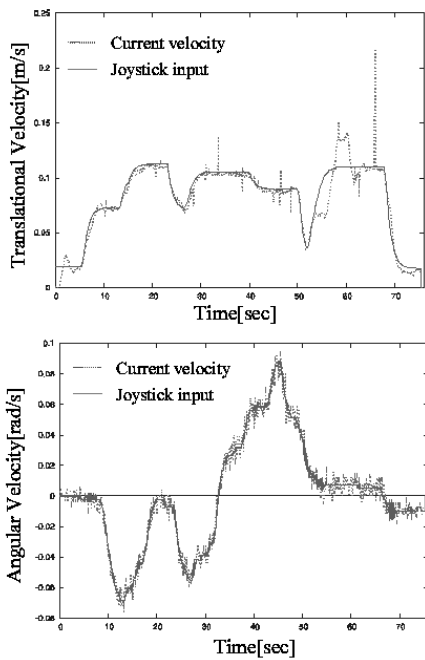


Fig. 11. Experimental results: translational velocity and angular velocity of the leader.

6 Conclusion

In this paper, we consider the problem of handling a large size of container as transporting a single object in coordination by two tracked mobile robots. In addition, we proposed a collision avoidance algorithm for two tracked mobile robots transporting a single object in coordination based on the function allocation concept. The follower transports the object with the leader without colliding with obstacles by using the visual sensor. We implemented the proposed algorithm in the mobile robots experimented system and illustrated the validity of the proposed algorithm. This algorithm has the advantages such that this system could be applied not only to the dynamic environment such that the position of obstacles are changing during the task is executed but also to cope with the slip of the wheels of the follower during the object transportation.

References

1. D.J.Stilwell, J.S.Bay, "Toward the Development of a Material Transport System using Swarms of Ant-like Robots," *Proc. of 1993 IEEE Int. Conf. Robotics and Automation*, Atlanta, USA, pp. 766–771, 1993.
2. L.Chaimowicz, T.Sugar,V.Kumar, et al, "An architecture for Tightly Coupled Multi-Robot Cooperation," *Proc.of 2001 IEEE Int. Conf. Robotics and Automation*, Seoul, Korea, pp. 2992–2997, 2001.
3. K.Inoue, T.Nakajima, "Cooperative Object Transportation by Multiple Robots with Their Own Objective Tasks," *Journal of the Robots Society of Japan*, Vol.19, pp. 888–896, 2001.(In Japanese).
4. N.Miyata, J.Ota, Y.Aiyama, et al, "Cooperative Transport System with Regrasping Car-like Mobile Robots," *Proc.of 1997 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Grenoble, France, pp. 1754–1761, 1997.
5. M.N.Ahmadabadi, E.Nakano, "Task allocation and distributed cooperation strategies in a group of object transferring robots," *Proc. of 1997 IEEE Int. Conf. Intelligent Robots and Systems*, Grenoble, France, pp. 435–440, 1997.
6. Z.Wang, M.N.Ahmadabadi, E.Nakano, T.Takahashi, "A Multiple Robots System for Cooperative Object Transportation with Various Requirements on Task Performing," *Proc.of the 1999 IEEE Int.Conf.on Robotics and Automation*, Detroit, USA, pp. 1226–1233, 1999.
7. T.G.Sugar, V.Kumar, "Control of Cooperating Mobile Manipulators," *IEEE Transactions on Robotics and Automation*, VOL.18, NO.1, pp. 94–103, 2002.
8. K.Kosuge, T.Oosumi, M.Sato, et al, "Transportation of a Single Object by Two Decentralized-Controlled Nonholonomic Mobile Robots," *Proc. of 1998 IEEE Int. Conf. Robotics and Automation*, Leuven, Belgium, pp. 2989–2994, 1998.
9. H.Takeda, Y.Hirata, Z.Wang, K.Kosuge, "Collision Avoidance Algorithm for Tow Tracked Mobile Robots Transporting a Single Object in Coordination Based on Function Allocation Concept," *Proc. of Distributed Autonomous Robotic Systems*, Fukuoka, Japan, pp. 155–164, 2002.

Development of a Terrain Adaptive Stability Prediction System for Mass Articulating Mobile Robots

Antonio Diaz-Calderon¹ and Alonzo Kelly²

¹ Jet Propulsion Laboratory
California Institute of Technology
adiaz@jpl.nasa.gov

² The Robotics Institute
Carnegie Mellon University
alonzo@ri.cmu.edu
<http://www.frc.ri.cmu.edu/alonzo>

Abstract. Dynamic stability is an important issue for vehicles which move heavy loads, turn at speed, or operate on sloped terrain. In many cases, vehicles face more than one of these challenges simultaneously. This paper presents a methodology for deriving proximity to tipover for autonomous field robots, which must be productive, effective, and self reliant under such challenging circumstances. The technique is based on explicit modeling of mass articulations and determining the motion of the center of gravity, as well as the attitude, in an optimal estimation framework. Inertial sensing, articulation sensing, and terrain relative motion sensing are employed. The implementation of the approach on a commercial industrial lift truck is presented.

1 Introduction

This paper addresses the issue of detecting proximity to a tipover event for an articulating mobile robot subject to large inertial accelerations. Articulating mobile robots can alter their configuration in response to commands issued to move the payload. As a result, the location of the center of mass (cg) of the vehicle is not fixed; a characteristic that may adversely affect the maneuverability of the vehicle. In this work, the general idea of computing the instantaneous motion of the cg of the vehicle relative to an inertial reference frame is addressed first. Following this, an algorithm to quantify the stability regime of the vehicle based on the inertial forces acting on the vehicle as a result of induced accelerations is presented.

1.1 Related Work

Study of rollover phenomena for heavy trucks abound in the automotive literature e.g., [3,4,9,11,12]. These authors study lateral rollover propensity under different handling conditions. While the proposed static characterizations of vehicle lateral stability are not appropriate as instantaneous measures of vehicle stability they do highlight the importance of considering the height of the cg and the vehicle

weight. Another important property that limits the applicability of these results to our problem is the fact that heavy trucks do not articulate any mass.

Articulating mobile robots on the other hand can change their configuration affecting the location of their cg. For these systems researchers have proposed a number of approaches [2,6–8,10] that can be broadly classified as quasi-static stability approaches. In a quasi-static approach it is assumed that the vehicle motion relative to an inertial reference frame is slow (e.g., 1 m/s or less), therefore inertial accelerations can be neglected with the exception of gravitational acceleration. As a result, the vehicle does not experience inertial forces that might trigger the tipover event. Approaches based on the support polygon are presented in [6,8,10]. These approaches suffer from the common problem of not measuring true cg motion while still assuming slow velocities. For vehicles that can articulate significant mass over long distances, and experience significant inertial accelerations, the quasi-static stability assumption is not applicable.

1.2 Approach

In this paper, the instantaneous motion of the cg is computed together with the inertial forces acting on the cg as a result of this motion. Sensor indications are mapped onto the instantaneous location of the cg, providing true cg motion.

To evaluate the stability regime of the vehicle, two derived vehicle states are defined: motion state and configuration state. The former provides information as to what the vehicle is doing (i.e., how fast it is turning, is it driving on a non-level road, how heavy is the load, etc.) and the latter provides information as to the configuration of the articulated parts at any point in time. The configuration state of the vehicle determines the limits on the D'Alembert forces that can be applied to the vehicle without actually causing a tipover event. It is also used to compute the instantaneous location of the vehicle cg relative to its polygon of support. The location of the cg (in 3D space) along with the inertial forces acting on the vehicle enable the computation of the *tipover stability regime*, which is a measure of the proximity of the vehicle to a tipover event as a function of motion and configuration states.

The proposed stability algorithm consists of two parts: 1) an estimation and prediction system and 2) a dynamic module. The estimation and prediction system estimates the current and future states of the vehicle using on-board sensors to measure relevant dynamic quantities. The dynamic module computes the stability regime using the vehicle's state estimate.

The utility of a tipover proximity indicator is that it can be used to drive a number of mechanisms, which can take corrective action. For man-driven vehicles, a console indication or audible warning could be produced. For autonomous systems, an exception can be raised for resolution at higher levels of the autonomous hierarchy or various governing mechanisms can be engaged to actively reduce the severity of the situation.

This paper is organized as follows: Section 2 presents the optimal estimation framework, Section 3 presents our approach to assessing the stability regime of the

vehicle and Section 4 presents some results from the application of the algorithms presented in the paper.

2 Optimal Estimation Framework

The estimation and prediction system provides the instantaneous motion of the vehicle cg using on-board sensors to measure relevant dynamic quantities. This framework is based on an extended Kalman filter (EKF) [5].

2.1 System Dynamics

The system's state vector describes the motion of the cg of the vehicle as well as the vehicle attitude:

$$\mathbf{x} = [\mathbf{v} \ \mathbf{a} \ \omega_z \ \alpha_z \ \phi \ \theta]^T \quad (1)$$

where \mathbf{v} and \mathbf{a} are the linear velocities and accelerations of the cg, and ω_z and α_z are the angular velocities and accelerations of the cg about the vehicle's vertical, and ϕ and θ are the Euler angles that describe the vehicle attitude. The system dynamics will be described by the following differential equation:

$$f(\mathbf{x}) = \left[\mathbf{a} \ \mathbf{0}_{3 \times 1} \ \alpha_z \ 0 \ \frac{\sin(\theta) \cos(\phi) \omega_z}{\cos(\theta)} \ -\sin(\phi) \omega_z \right]^T \quad (2)$$

This system model is nonlinear, therefore given the initial condition $\mathbf{x} = \mathbf{x}_k$ at time t_k we use the formula for the Euler method (Eq. 3) to advance the solution of the nonlinear differential equation $\dot{\mathbf{x}} = f(\mathbf{x})$ from t_k to $t_{k+1} = t_k + dt$.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + f(\mathbf{x}_k) dt \quad (3)$$

2.2 Measurements

We are interested in the true motion of the cg because that motion determines the instantaneous stability regime (e.g., stable, marginally stable) of the vehicle. However, sensor indications describe the sensor's own motion. To find the true cg motion, sensor indications are mapped onto sensor indications at the cg taking into account the motion of the cg relative to the sensor.

Transformation of inertial sensor measurement. With reference to Fig. 1, let frame $\{i\}$ be the inertial reference frame attached to the earth, frame $\{e\}$ be a frame attached to the encoder sensor, frame $\{a\}$ be attached to the accelerometer, and frame $\{c\}$ be attached to the cg of the vehicle. The relative motion between these frame is constrained as follows: $\vec{\omega}_a^c = \vec{\omega}_a^e = \vec{\omega}_c^e = 0$, $\vec{v}_a^e = 0$, $\vec{v}_c^e \neq 0$, and $\vec{v}_c^a \neq 0$.

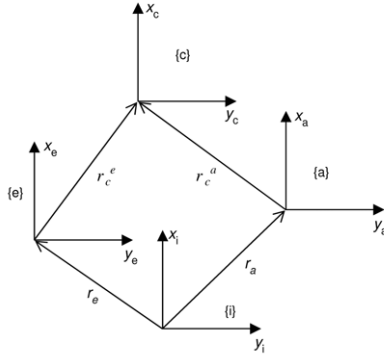


Fig. 1. Frames used in the transformation of sensor data.

With this frame definitions, the speed of the cg (i.e., \vec{v}_c) relative to the inertial frame is computed from Eq. 4 as follows.

$$\vec{v}_c = \left(\frac{d\vec{r}_c}{dt} \right)_i = \vec{v}_e + \vec{v}_c^e + \vec{\omega}_e \times \vec{r}_c^e \tag{4}$$

We use the symbols \vec{r}_c^n and \vec{v}_c^n to describe the position and velocity of the cg relative to the n th sensor.

The specific force, \vec{t}_c , measured at the cg is computed from $\vec{t}_c = \vec{t}_a + \Delta \vec{t}$, where \vec{t}_a is the specific force measured by the accelerometer and $\Delta \vec{t}$ is an increment in specific force due to the offset of the cg relative to the accelerometer. This increment can be computed from Eq. 5 as follows:

$$\Delta \vec{t} = \vec{a}_c^a + 2\vec{\omega}_a \times \vec{v}_c^a + \vec{\alpha}_a \times \vec{r}_c^a + \vec{\omega}_a \times \vec{\omega}_a \times \vec{r}_c^a \tag{5}$$

The location and motion of the cg relative to the sensor induce inertial accelerations that must be accounted for when measuring true cg motion. Induced inertial accelerations are the following:

Acceleration seen in frame {a}	\vec{a}_c^a
Euler acceleration	$\vec{\alpha}_a \times \vec{r}_c^a$
Coriolis acceleration	$2\vec{\omega}_a \times \vec{v}_c^a$
Centripetal acceleration	$\vec{\omega}_a \times \vec{\omega}_a \times \vec{r}_c^a$

where $\vec{\omega}_a$ and $\vec{\alpha}_a$ are the angular velocity and acceleration (about the local vertical) of the sensor.

Kinematics of the CG. As the robot is articulated, the cg experiences additional velocities and accelerations relative to the inertial frame. To account for these velocities and accelerations, we compute the instantaneous motion of the cg relative to the body frame. The sensor transformations derived earlier are then applied to obtain the true cg motion.

$${}^B\mathbf{r}_c = {}^B\mathbf{T} \left({}^0\mathbf{r}_{c_0} m_0 + \sum_{i=1}^n \left(\prod_{j=1}^i {}^j\mathbf{T}(\Theta_j) \right) {}^i\mathbf{r}_{c_i} m_i \right) \frac{1}{m} \quad (6)$$

Eq. 6 defines the instantaneous location of the cg (${}^B\mathbf{r}_c$) as a function of joint variables Θ_j .

Measurement model. The measurement vector \mathbf{z} includes: yaw rate gyro (ω), steer encoder (δ), speed (v), accelerometers specific force (t), and roll (Φ) and pitch (Θ) inclinometers.

$$\mathbf{z} = [z_\omega \ z_\delta \ z_{v_x} \ z_{v_z} \ \mathbf{z}_{t_a} \ z_\Phi \ z_\Theta]^T \quad (7)$$

Eq. 8 gives the relationship between the state (\mathbf{x}) and measurement (\mathbf{z}) vectors, which is non-linear. The measurement Jacobian is defined as $\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ and the matrix ${}^i_B\mathbf{R} = \mathbf{R}_z(0) \mathbf{R}_y(\theta) \mathbf{R}_x(\phi)$ is the ZYX-Euler angles rotation matrix that describes the orientation of the truck relative to the earth, and ${}^B_i\mathbf{R} = {}^i_B\mathbf{R}^{-1}$.

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \arctan \left(\frac{\omega_z}{\frac{v_y - v_{c_y}^\delta - \omega_z r_{c_x}^\delta}{v_x - v_{c_x}^\delta + \omega_z r_{c_y}^\delta}} \right) \\ v_x - v_{c_x}^e + \omega_z r_{c_x}^e \\ v_z - v_{c_z}^e \\ \mathbf{a} + {}^B_i\mathbf{R} \ {}^i\mathbf{g} - \Delta \mathbf{t}_a \\ \arctan \left(\frac{-(\mathbf{a} + {}^B_i\mathbf{R} \ {}^i\mathbf{g} - \Delta \mathbf{t}_{inc})_y}{(\mathbf{a} + {}^B_i\mathbf{R} \ {}^i\mathbf{g} - \Delta \mathbf{t}_{inc})_z} \right) \\ \arctan \left(\frac{(\mathbf{a} + {}^B_i\mathbf{R} \ {}^i\mathbf{g} - \Delta \mathbf{t}_{inc})_x}{(\mathbf{a} + {}^B_i\mathbf{R} \ {}^i\mathbf{g} - \Delta \mathbf{t}_{inc})_z} \right) \end{bmatrix} \quad (8)$$

3 Assessment of Stability Regime

Assessing stability regime is accomplished in two steps. First, the geometric footprint of the vehicle is computed. This is used to determine the limits on the motion of the cg; i.e., stability envelope. Second, the forces acting on the cg are determined. These forces are used to formulate a single stability measure of the vehicle's stability regime, which is expressed as an angle measure.

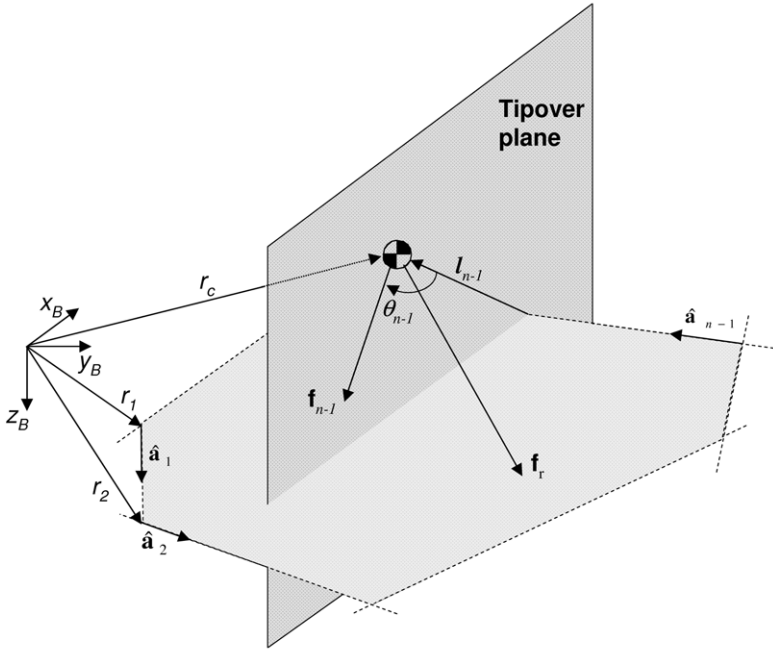


Fig. 2. Assessment of dynamic stability

3.1 Definition of the Support Polygon

The polygon of support is defined by the robot contact points with the ground, which form a *convex polygon* when projected onto the horizontal plane. For example, in a wheeled vehicle these contact points may be defined by the center of the contact patch of the tire and the road. Let \vec{r}_i represent the location of the i th ground contact point and let \vec{r}_c represent the location of the vehicle center of mass. As illustrated in Fig. 2 these vectors can be expressed in the vehicle frame $\{B\}$ and numbered such that the unit normal of the support polygon is directed upward and out of the ground. The boundary of the support polygon is defined by the lines joining the ground contact points. These lines are the candidate tipover axes, $\vec{a}_i, i = 1 \dots n$. The i th tipover axis is the axis about which the vehicle will physically rotate during a tipover event; it defines the normal to the *tipover plane* (Fig. 2). The tipover plane serves as a simplification in which the vehicle mass properties and the forces acting on the vehicle are projected onto to analyze the tipover propensity about the plane normal.

3.2 Formulation of the Stability Measure

The sum of forces acting on the vehicle must be balanced when the vehicle is in a stable configuration. These forces include the inertial forces ($\vec{f}_{\text{inertial}}$), gravitational

loads (\vec{f}_g), ground reaction forces at the vehicle's wheels (\vec{f}_s), and any other external disturbances acting on the vehicle (\vec{f}_d). The force equilibrium equation can be written as in Eq. 9.

$$\sum \vec{f}_{\text{inertial}} = \sum (\vec{f}_g + \vec{f}_s + \vec{f}_d) \quad (9)$$

The net force acting on the cg that would contribute to tipover instability about any tipover axis, \vec{f}_r , is defined in Eq. 10.

$$\vec{f}_r \triangleq \sum (\vec{f}_g + \vec{f}_d - \vec{f}_{\text{inertial}}) = - \sum \vec{f}_s \quad (10)$$

For a given tipover axis \vec{a}_i , we are only concerned with those components of \vec{f}_r which act *about* the *ith* tipover axis. The projection of the resultant force onto the *ith* tipover plane and the moment of the resultant force about the plane normal are used to compute the stability measure as follows.

Let \vec{f}_i be the projection (e.g., the component of \vec{f}_r acting along the *ith* tipover axis) of the resultant force onto the tipover plane. Then

$$\vec{f}_i = \vec{f}_r - (\vec{f}_r \cdot \hat{a}_i) \hat{a}_i \quad (11)$$

and $\hat{a}_i = \vec{a}_i / |\vec{a}_i|$. Similarly, the torque acting about the *ith* tipover axis can be computed from Eq. 12.

$$\vec{n}_i = ((\vec{l}_i \times \vec{f}_r) \cdot \hat{a}_i) \hat{a}_i \quad (12)$$

In Eq. 12, \vec{l}_i is the *ith* tipover axis normal given by

$$\vec{l}_i = (\vec{r}_i - \vec{r}_c) - ((\vec{r}_i - \vec{r}_c) \cdot \hat{a}_i) \hat{a}_i \quad (13)$$

The stability measure for each tipover axis is defined as the subtended angle between the *ith* resultant force, \vec{f}_i , and the *ith* tipover axis normal, \vec{l}_i , as illustrated in Fig. 2. This measure is denoted by Θ_i and can be computed from Eq. 14 as follows:

$$\theta_i = - \arcsin \left(\frac{(\vec{l}_i \times \vec{f}_r) \cdot \hat{a}_i}{|\vec{l}_i| |\vec{f}_i|} \right) \quad (14)$$

The stability regime with respect to a given tipover axis is determined from the value of the corresponding resultant angle as follows. If the resultant angle $\Theta_i > 0$ the vehicle is stable about the *ith* tipover axis. If $\Theta_i = 0$ the vehicle is marginally stable about that tipover axis. This means that the ground reaction forces at the inside support points relative to the *ith* tipover axis are zero. If $\Theta_i < 0$ the vehicle is critically stable, meaning that it is tipping over about the *ith* tipover axis. The overall stability regime of the vehicle is defined by Eq. 15.

$$\theta = \min (\theta_i) \quad (15)$$



Fig. 3. Vehicle test bed. In (b) the mast is extended to 340 inch high.

4 Results

The test bed selected to exercise the algorithm is a lift truck. Two platforms were used in testing the algorithm: hardware and simulation. The hardware platform is shown in Fig. 3. A commercial lift truck underwent major retrofitting to incorporate the sensor suite used in the system.

There are two types of sensors used in the system: inertial and articulation. Inertial sensors measure inertial motion and direction of gravity. Articulation sensors measure motion of every articulated part of the vehicle.

The computing platform is a general purpose 3U form factor 8-slot chassis containing a backplane for PXI and Compact PCI modules. The CPU module is a National Instruments PXI-8170 series consisting of a 850 MHz Pentium III processor with 256 MB of memory.

As part of the hardware platform, a data logger system was developed. The main function of the logger is to provide input data to the system from either the vehicle sensors, from previously generated log files or from simulated sensor data. In the design of the logger, care was taken to guarantee deterministic log playback to ensure identical system response regardless of the source of the input data. Experiments to verify the functionality of the system involved driving the vehicle on ramps, and level ground. Different handling maneuvers were executed, e.g., constant curvature/constant speed/constant load location, constant curvature/variable speed, variable curvature/variable speed. Testing on ramps was important to verify if the system was able to decouple inertial accelerations due to motion from gravity:

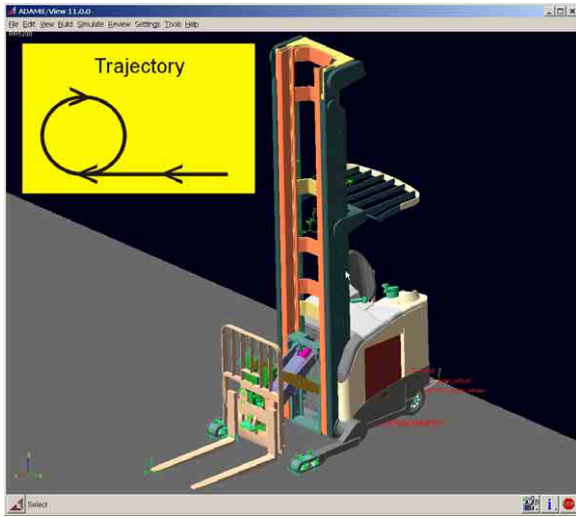


Fig. 4. Rigid body dynamic model of the test bed. The trajectory executed is shown in the inset.

inclinometers and accelerometers are unable to make the distinction. In all these cases, the system estimated the attitude of the vehicle (e.g., terrain grade) to within one degree and was able to compute the stability measure by decoupling inertial accelerations from gravity. More aggressive maneuvers were left to be tested in simulation. The simulation environment is based on a rigid body dynamic model of the truck, which has been developed in ADAMS [1] (Fig. 4).

A test run was executed with a trajectory composed of two sections: acceleration section and maneuver section. A linear path is used to accelerate the vehicle to the commanded speed (5 m.p.h.). This is followed by a constant curvature path with variable longitudinal velocity (5 \rightarrow 10 m.p.h.). This is illustrated in the inset in Fig. 4. The maneuver was executed while the vehicle was carrying a load of 3000 lb. Simulated sensor indications for this maneuver are illustrated in Fig. ??-a while estimated velocities and accelerations at the cg of the vehicle are illustrated in Fig. ??-b.

As the vehicle travels in this constant curvature path, the longitudinal velocity of the truck is increased to 10 m.p.h. This causes an increase in lateral acceleration as illustrated in Fig. ??-b. As a result, the resultant angles that correspond to the outside tipover axes decrease to a point where the truck has reached marginal stability: the resultant angle (θ_2) in Fig. 6 goes to zero. It is at this point where the vehicle's inside wheels are about to lift-off the ground. Verifying the reaction forces at the wheels (Fig. 7), we confirm that the reaction forces on the inside wheels become zero.

Wheel reaction forces. Wheel reaction forces for the inside wheels go to zero as the vehicle reaches marginal stability.

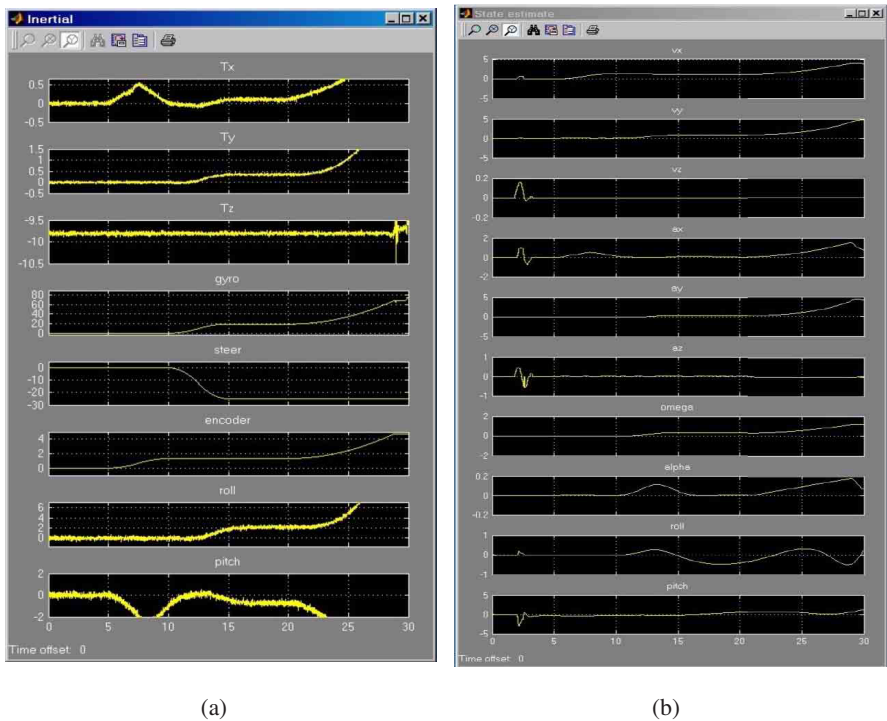


Fig. 5. (a) Simulated sensor indications. Tx, Ty, and Tz represent accelerometer specific force. Roll and pitch are the outputs of the simulated inclinometers. As the vehicle travels the constant curvature path the roll inclinometer is unable to differentiate between gravity or inertial acceleration due to motion. (b) Vehicle cg state. As the vehicle enters the constant curvature path, lateral velocity and acceleration are being estimated. Lateral acceleration is used in the stability measure to monitor the lateral stability of the vehicle within its stability envelope. The attitude of the vehicle is given by the roll and pitch angle outputs.

5 Conclusions

This paper has presented an approach to estimating proximity to tipover for a case much more general than previous approaches. The technique applies to vehicles which articulate significant mass, which experience significant inertial forces, and which move over terrain which is not necessarily level. Key elements of the approach include explicit models of mass articulation, an optimal estimation approach to motion determination, explicit compensation for inertial forces, and a formulation that determines the motion of the cg frame of reference. The generality of the approach makes it potentially relevant to a broad class of outdoor material handling and excavation vehicles whether they are man-driven or robotic. The system could form the basis of a governing system, which discourages aggressive driving for

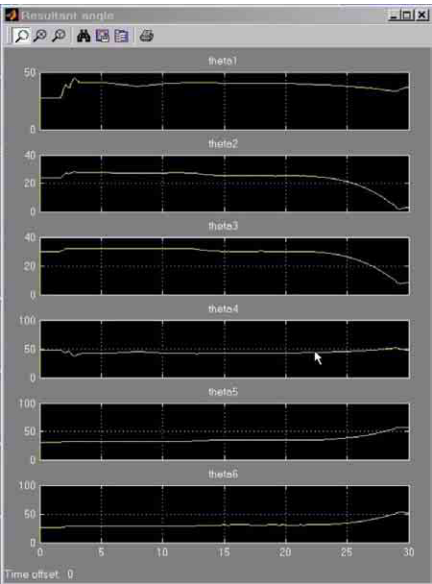


Fig. 6. Stability regime expressed as the angle of the resultant. This vehicle has six candidate tipover axes hence the six resultant angles. Resultant angles associated with the tipover axes 2 and 3 decrease since these axes are the candidate tipover axes for this maneuver. Axes 5 and 6 are on the opposite side therefore their associated angles increase. Finally, axes 1 and 4 correspond to the front and rear tipover axes (e.g., longitudinal tipover).

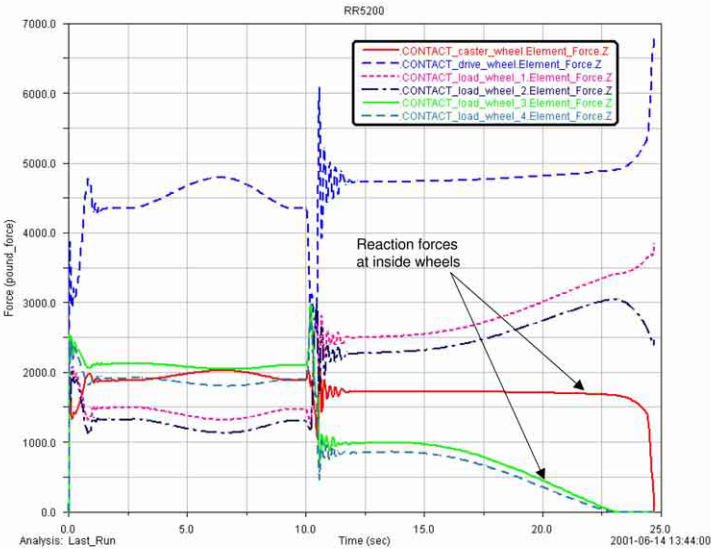


Fig. 7. -90

man-driven vehicles or one that implements a low level reactive control system for an autonomous vehicle.

Acknowledgements

Dr. Antonio Diaz-Calderon is currently employed by Jet Propulsion Laboratory, California Institute of Technology through a contract with the National Aeronautics and Space Administration. The work described in this paper is not associated with JPL, CalTech or NASA.

References

1. MSC.ADAMS: 2003. <http://www.mscsoftware.com>.
2. Dubowsky, S. and E. E. Vance: 1989, 'Planning mobile manipulator motions considering vehicle dynamic stability constraints'. In: *IEEE International Conference on Robotics and Automation*. Scottsdale, AZ.
3. Genta, G.: 1999, *Motor vehicle dynamics: Modeling and simulation*, Vol. Vol. 43 of *Series on Advances in Mathematics for Applied Sciences*. Singapore: World Scientific.
4. Gillespie, T. D.: 1992, *Fundamentals of vehicle dynamics*. Warrendale, PA: Society of Automotive Engineers.
5. Maybeck, P. S.: 1982, *Stochastic models, estimation and control*, Vol. 141-1 of *Mathematics in Science and Engineering*. Paris: Academic Press.
6. Papadopoulos, E. G. and D. A. Rey: 1996, 'A new measure of tipover stability margin for mobile manipulators'. In: *IEEE International Conference on Robotics and Automation*. Minneapolis, MN.
7. Shiller, A. and Y. R. Gwo: 1991, 'Dynamic motion planning of autonomous vehicles'. *IEEE Transactions on Robotics and Automation* **Vol. 7**.
8. Sreenivasan, S. V. and B. H. Wilcox: 1994, 'Stability and traction control of an actively actuated micro-rover'. *Journal of Robotics Systems* **Vol. 11**(No. 6).
9. Steiner, P., P. Weidel, H. Kublbeck, H. Steurer, P. Hora, and D. Zechmair: 1997, 'Rollover detection'. Technical report, SAE Technical Paper no. 970606.
10. Sugano, S., Q. Huang, and I. Kato: 1993, 'Stability criteria in controlling mobile robotic systems'. In: *IEEE International Workshop on Intelligent Robots and Systems*. Yokohama, Japan.
11. UMTRI: 1991, 'Mechanics of heavy-duty trucks and truck combinations'. Technical Report UMTRI-80868, Transportation Research Institute, The University of Michigan. Engineering Summer Conferences.
12. Winkler, C. B., D. Blower, R. D. Ervin, and R. M. Chalasani: 2000, *Rollover of heavy commercial vehicles*. Warrendale, PA: Society of Automotive Engineers.

ROBHAZ-DT2: Passive Double-Track Mobile Manipulator for Explosive Ordnance Disposal

Sungchul Kang¹, Changhyun Cho¹, Changwoo Park², Jonghwa Lee¹,
Dongseok Ryu¹ and Munsang Kim¹

¹ Intelligent Robotics Research Center
Korea Institute of Science and Technology, Korea
{kasch, chcho, jonghwalee, sayryu and munsang }@kist.re.kr
<http://humanrobot.kist.re.kr>

² Yujin Robotics Ltd., Korea
cwpark@yujinrobot.com
<http://yujinrobot.com>

Abstract. In this paper, design and integration of ROBHAZ-DT2 is presented which is a newly developed mobile manipulator system. It is designed to carry out military and civilian missions in various hazardous environments. In developing the integrated ROBHAZ-DT2 system, we have focused on two issues : 1) novel mechanism design for mobility and manipulation and 2) intuitive user interface for teleoperation. In mechanism design, a double tracks connected by a passive joint has been designed to achieve high speed and rugged mobility on uneven terrain. In addition, a six-dof foldable manipulator suitable for the mobile manipulation has been designed. Secondly, a new compact 6-dof haptic device has been developed for teleoperation of the ROBHAZ-DT2. This haptic device is specially designed for simultaneous control both of the mobile base and the manipulator of ROBHAZ-DT2. As a result of integration of RobhAZ-DT2 and the user interface unit including the haptic device, we could successfully demonstrate a typical EOD task requiring abilities of mobility and manipulation in outdoor environment.

1 Introduction

In the 21st century, more robots will take the place of human labor in many areas. In particular, they will perform various missions in hazardous environments such as fire fighting, rescuing people, demining, suppressing terrorist outrage, scouting enemy territory and so on. In order to do that, the robots should have advanced abilities in *mobility and manipulation* in teleoperated environment.

In the hazardous environment applications, the abilities in mobility and manipulation via teleoperation are very important issues because the robot should be able to move on uneven terrain and manipulate a variety of ordnances that are not predefined.

In terms of mobility, as a first issue, there have been a lot of studies so far on the intelligent wheel mechanism for a given terrain condition. Generally, there are three types of mobile mechanisms for this kind of robots : wheel, track and walking type mechanism. Robots with wheel mechanism are inferior to robots with track when they are traveling over rough terrain. Walking robots usually have complex

structures so that they have drawbacks in difficulty in control and slow speed. In that sense, the track mechanism has advantages in high speed driving and mobility under severe conditions[1]. Contrary to these advantages, it consumes much more energy than the other ones. Therefore it is needed to design a track mechanism to overcome this drawback. It is natural, therefore, that recent researches are to develop track type robots with flexible configurations according to the ground conditions for more efficient adaptation to landform. Kohler, et. al. suggested a moving mechanism with 4 flexible tracks[2]. and Maeda, et. al. proposed a moving mechanism that combines the characteristics of robots with four wheels and track-type robots[3]. Iwamoto and Yamamoto developed a track mechanism that changes its configuration while climbing up stairways[4]. Yoneda, et. al. suggested a track type moving mechanism that uses a track with a material having higher coefficient friction and wider contact area between the track and steps[5]. More recently, Schempf, et. al. suggested a robot with flexible track that has a similar structure to our ROBHAZ-DT2 in this paper, but the Pandora[6] changes the track configuration actively to adapt to the ground condition.

Compared with the existing mobile mechanisms described above, the design of mobile mechanism of our ROBHAZ-DT2 has some features that it has double tracks at each side and the double tracks can rotate passively each other. The passive rotational joint connects the front body with the rear one makes the ROBHAZ-DT2 possible to have good adaptability to uneven terrain including stairs. This passive adaptation mechanism reduces energy consumption in moving on uneven terrain as well as it offers simplicity in the design of the ROBHAZ-DT2.

Together with the mobility issue, on the other hand, we consider *telemanipulation*, which is manipulation via teleoperation, as another important issue for the *explosive ordnance disposal*(EOD) tasks. For the purpose of good telemanipulation, we decided to attempt a novel design both of a manipulator suited to be mounted on the mobile base and a haptic device dedicated to mobile manipulation tasks.

The six-dof manipulator of the ROBHAZ-DT2, which is designed to achieve various EOD tasks, has characteristics that it has a compact and foldable structure with light weight. To reduce the weight of the moving bodies of the manipulator as well as to prevent the backlash existing in geared transmission, we chose to use tendon transmission. In teleoperated control of a mobile manipulator system like the ROBHAZ-DT2, a simple and intuitive haptic master device is required which is specially designed for effectively controlling mobile manipulator system. Especially, six-dof haptic feedback function is needed in typical EOD operations. To meet these functional requirements, a new haptics device dedicated to mobile manipulator control is developed. It is composed of two mechanical parts. The lower mechanism offers planar three dofs to control only the mobile base, while the upper one, which is serially attached on the lower part, can expand dof of the haptic master up to six dof. The whole structure forms a parallel mechanism, which leads the haptic master to achieve high stiffness and responsiveness.

Centered on these two issues, we added some other key components such as stereoscopic visual interface with an immersive display, wireless communication for

audio/visual and data exchange, and so on. By integrating these key components, a teleoperation system for the ROBHAZ-DT2 has been developed for typical EOD applications.

This paper is organized as follows; section 2 describes the novel design of double track mechanism and its analysis. Next, the telemanipulation issues including the new design of the manipulator and the haptic device are dealt in section 3 and 4. In section 5, full integration of the ROBHAZ-DT2 system including the structure of control system is described. By showing a typical EOD demonstration task, the result of the integration is presented in section 6. Finally concluding remarks and discussion are summarized in section 7.

2 Passive Double-Track Mechanism

A new track mechanism, which can give a passive adaptability based on a unique link structure, is exploited for the ROBHAZ-DT2 to increase its mobility on rough landform. Fig. 1 shows the design of the double track mechanism of the ROBHAZ-DT2. The double track mechanism is composed of two tracks driven by single motor for each side. The ROBHAZ-DT2 consists of three parts : the front track body, rear track body, and main body. The front-left and rear-left track (or the front-right and rear-left track) in Fig. 1 (b) have the common driving shaft and each motor is equipped for actuating the shaft. Thus, the two tracks at each side rotate in the same direction as that of the driving shaft as illustrated in Fig. 1 (a).

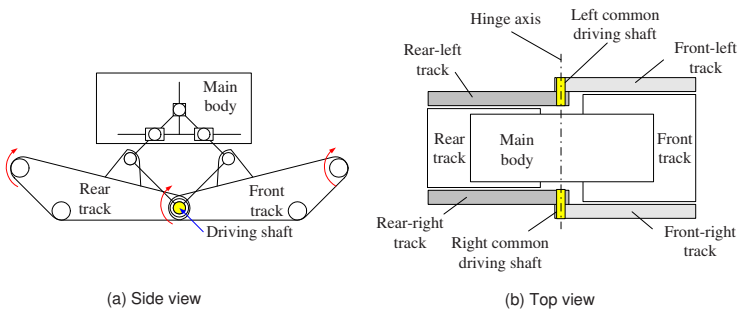


Fig. 1. Layout of the ROBHAZ-DT (patent pending)

The passivity applied to the ROBHAZ-DT2 is simply acquired by attaching the front and rear through a hinge joint without any actuator. The hinge axis is marked in Fig. 1 (b) and is coincided with the axis of the driving shaft. Fig. 2 illustrates a relative motion between the front and rear bodies. As the front track rotates in the angle of β as marked in Fig. 2 from the initial state f to the arbitrary final state f' , the main body also rotates by a link mechanism.

Changing configuration will give alteration of the weight center or ZMP (Zero Moment Position), which gives influence on mobility of a vehicle in rough terrains.

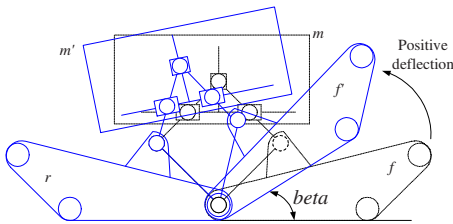


Fig. 2. Relative motion on the hinge axis

It is generally known that single-track mechanisms have limitation in rugged terrain due to a fixed weight center in the local frame, which greatly affects a stability margin (the minimum length between weight center and edges of a supporting area). The stability margin in the case of single-track vehicle is only determined by the other factor such as the inclination of landform. Therefore, it is needed to design to have linked mechanism of multiple track bodies to overcome the limitation. This is a main reason that we developed the double track mechanism. In case of the double-track mechanism, the center of mass varies and the supporting area is also altered by the passive motion when traveling over the landform.

Effect of the passivity is investigated in an example of stairway climbing as shown in Fig. 3. Stairway is one of the good landform to verify the mobile capability of a vehicle, and it is widely used among many researches related in developing a vehicle for such irregular terrain due to its difficulties on traveling. For simplicity, the example is drawn in 2D vertical plane. $\{B\}$ is a body fixed coordinate frame and r_m is a position vector of center of weight. L is a projected line representing the supporting area as depicted in Fig. 3 (a). The point A is a center of weight projected on a supporting area, and δ is a stability margin.

During climbing the stairs, both the positive and negative rotations are observed in Fig. 3. In the situation of Fig. 3 (b), for instance, a supporting area is increased compared with that of Fig. 3 (a). The L would be shorter in case of single track. It means that the double-track design has advantage in low gravity center and large stability margin. Note that no actuator is used to generate the rotation and the rotation is only produced by gravity on the given landform. This is the feature that the double-track mechanism shows relatively better performance than a single-track mechanism. Thus the passivity can improve adaptability to a rough terrain and the motion (i.e., the positive and negative deflection) occurs successively during stairway climbing.

3 Manipulator Design

A manipulator mounted on a mobile base like the ROBHAZ-DT2, which is designed to achieve various EOD tasks, is required to meet the following design objectives.

- Compact, light and rugged design
- Good payload and large gripping stroke

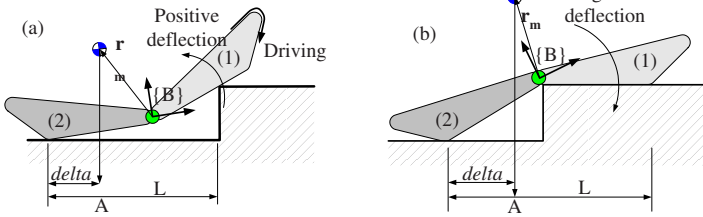


Fig. 3. Passivity in stairway climbing

- Small backlash
- Compliant control function
- Cheap components

The ROBHAZ-DT2's manipulator has been designed and manufactured to satisfy the given these design objectives as many as possible. For compact and light design, first, it has a foldable structure and its lower three joints are indirectly actuated by tendon transmission to exclude the weight of the actuator from the link body. Additionally, the tendon mechanism can eliminate the backlash problem. Through the tendon transmission, we can get the large duty ratio (payload/total weight) compared with other geared (or belted) transmission. Fig. 4 (a) shows the second joint with tendon transmission of the manipulator. In the wrist design as shown in Fig. 4 (b), we adopted the *double active universal joint*(DAUJ) mechanism to generate two-dof wrist motion with very compact volume. The DAUJ mechanism was originally exploited for anthropomorphic finger's joint.[7]

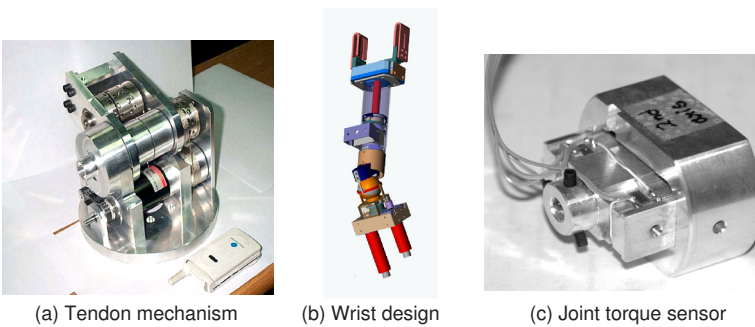


Fig. 4. Design of a manipulator for ROBHAZ-DT2

In telemanipulation for EOD, compliant manipulation is one of the essential functional requirements. In this work, active compliant control has been implemented by strain-gage type joint torque sensors at the lower three joints and gripper force

sensors which are integrated into each jaw of the gripper. The strain-gage type joint torque sensor shown in Fig. 4 (c) is available with lower price than the commercial force/torque sensors. To produce the programmable active compliance, the joint torque and grip forces measured at the strain gauges are transformed into Cartesian forces/torques via kinematic computation and then fed back into the damping-based compliant controller that maps the forces/torques into the linear/angular velocities.

4 Haptic Device for ROBHAZ-DT2

The proposed haptic master is composed of upper and lower mechanisms as shown in Fig. 5 (a). A parallel mechanism is adapted to making both mechanisms. It leads the haptic master to achieve high stiffness and accuracy. This structure allows actuators to be placed on the base, thus leading to good back-drivability because the actuators need not move during operation of the device.

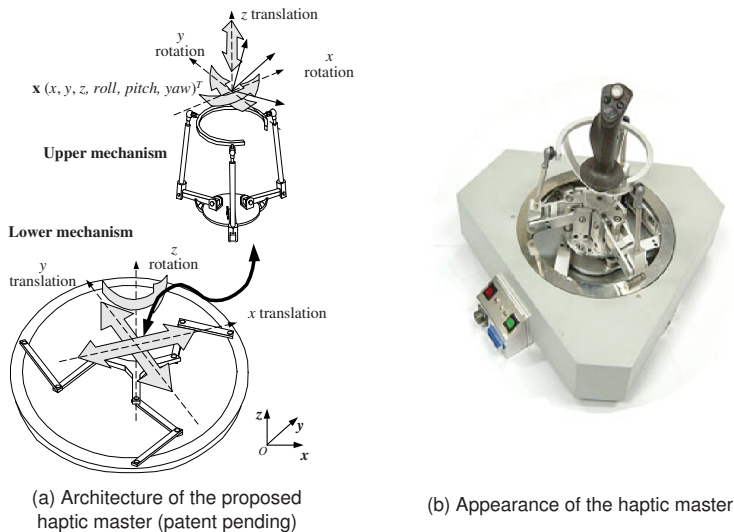


Fig. 5. Haptic master for ROBHAZ-DT2

The lower mechanism is designed to be a planar 3-dof parallel manipulator. When selection of planar 3-dof parallel manipulators is limited to three identical limb structures, seven planar 3-dof parallel manipulators (RRR, RRP, RPR, PRR, RPP, PRP, and PPR) are feasible. [8,9]

In robot design with parallel mechanism, a prismatic joint is preferred to get high resistance from disturbance on the end-effector. On the contrary, as back drivability is regarded as a positive factor in haptics, there is the reason that the RRR planar mechanism is selected from the seven candidates mentioned above. The RRR mechanism is shown in the lower part of Fig. 5 (a). For design of the upper part,

we have investigated various spatial parallel devices. [10–12] In the same reason as the selection of upper part, the RRR parallel mechanism is also applied to the upper mechanism. Using a revolute joint has additional merits such as stiffness and rigidity, which result in making structure simple and firm. Finally, a new 6-dof haptic master is constructed by putting both mechanisms together serially.

In most six-dof haptic devices, all six actuators should be simultaneously activated to create force feedback in full operational space even when only a few dof motions are needed, since the Cartesian space and joint space are closely coupled in the kinematics of the device.

In developing a haptic master for a mobile manipulator, we need to consider that the mobile manipulator mainly works in planar three-dof motion : two translations and one rotation. If a general six-dof parallel haptic device would be used in this case, all actuators should be controlled all the time regardless of the number of dofs needed in operational space.

As a result of consideration of dedicated use for teleoperation of a mobile manipulator, a new design of the six dof haptic master for ROBHAZ-DT2 has been manufactured as shown Fig. 5 (b). It covers the whole six-dof in motion and force feedback. Only three actuators are activated to create force feedback in planar three-dof motion for navigation of the mobile base. It can be utilized efficiently and intuitively in this application.

5 Integration of ROBHAZ-DT2 System

The ROBHAZ-DT2 has been developed as shown in Fig. 6 (a). As well as unique design in the track mechanism and the manipulator, this robot has some other features for practical deployment in outdoor environment. Between the main body and the front, rear track bodies, hydraulic dampers are equipped for mitigating shock on traveling. A clutch in driving shaft is designed for shifting high and low speed gears trains. The ROBHAZ-DT2 has nine ultrasonic sensors in order to perceive obstacles and a posture sensor to measure its posture during traveling. The ultrasonic sensors can detect not only circumferential obstacles like walls but also a ditch or cliff. The ultrasonic information is used for force feedback and also escaping obstacles in reactive way. As well, a pan-tilt stereo camera is prepared for monitoring a circumference view at a long distance. The feature of the stereoscopic monitoring system enables the operator to view the surroundings of the ROBHAZ-DT2 more realistically by the stereo camera attached on the pan-tilt mechanism. When the operator works at narrow distance such as moving mines or explosives, it gives the remote a more realistic sense. The image information of video is sent to the remote control station via an independent RF channel to reduce the traffic in data communication.

A cooling system by means of heat exchange is introduced to ensure the dust-proof cooling for safe outdoor operations. The major specification data of the ROBHAZ-DT2 are listed in Table 1.

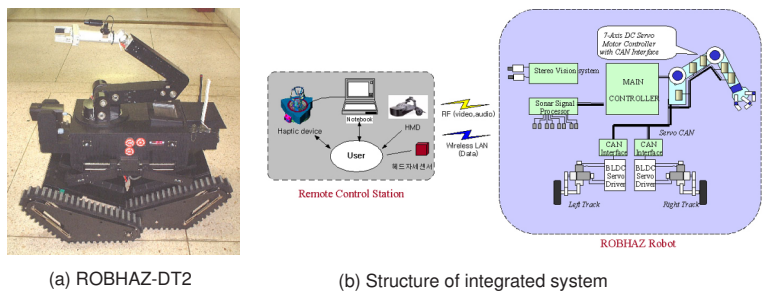


Fig. 6. Integrated ROBHAZ-DT2 system

Table 1. Specifications of the ROBHAZ-DT2 robot

External size(W H L)	690 500 910 mm(manipulator excluded)
Weight (batteries included)	145 Kg
Max. velocity	2.78 m/sec (10 Km/h)
Maximum climbing-up angle	32°
Passive joint limit	+10 ~ -30
Batteries	actuators : lead-acid controllers : Lithium ion
Continuous operation time	1hr

Fig. 6 (b) shows the structure of integrated control system that consists in the ROBHAZ-DT2 robot and the remote control station. The main controller of the robot is composed of one main-controller (computer the Pentium III 850MHz) and six sub-controllers (AVR-micro processor, motor driver, DI/O interface module).

The main-controller exchanges information with the remote operator. The sub-controllers are connected to the main-controller through CAN (Control Area Network) bus and exchanges the speed data of motors and various sensor information. This two-layered control structure has been built to enable the sub controllers to control real time to share the calculation load. The command and monitoring data exchange between the ROBHAZ-DT2 robot and the operator is done via a wireless LAN. The ROBHAZ-DT2 sends various sensor information to the remote operator station while the remote operator commands to move the ROBHAZ-DT2 in a desired direction and speed. In this control system, all control software is coded by Visual C++ running in Windows 2000.

6 EOD Experiment

As a result of full integration of the robot system and remote control station, a simple EOD demonstration task was successfully executed in real outdoor environment shown in Fig. 7 and Fig. 8. The demonstration setup for the EOD is artificially constructed. As shown in Fig. 7, an operator remotely controls the robot by means of the haptic device, a head mount display, and a laptop computer. First, he controls

the mobile vehicle and manipulator to access the explosive ordnance. By means of the double-track mechanism, as shown in Fig. 8 (a)-(c), the ROBHAZ-DT2 could easily travel over stairways and irregular surface. During the accessing motion, the operator could monitor the stereoscopic view transmitted from the pan-tilt stereo camera mounted in front of the robot. In manipulating the object lying in a car shown in Fig. 8 (d), the operator could approach, grip and pick up an imitated bomb by feeling the contact forces via the haptic interface and monitoring fine view transmitted from the camera equipped at the arm's gripper. Then it moved to a safe place for the disposal in Fig. 8 (f).



Fig. 7. Remote operator station



(a) Start and climbing



(b) Running on irregular surface



(c) Approach



(d) Grip



(e) Moving for disposal



(f) Disposal

Fig. 8. Demonstration of explosive ordnance disposal

7 Concluding Remarks

In this work, we have aimed to develop a practical robot system for EOD that can be commercialized and deployed in real EOD application in the near future. In order

to do that, we could have focused our research on the design of advanced track mechanism and improvement of telemanipulation function because the two issues are keys to upgrade current mobile manipulator systems for hazardous applications such as EOD.

As a result of survey on military application, we found that EOD by water-disrupter device is widely used and that it is a good application to produce a lot of demands. As well as EOD applications, we found that there are great deals of demand on teleoperated mobile manipulators in the CBR(chemical, biological and radioactive) warfare operation. Likewise, we are currently classifying the real EOD operations and investigating the feasibility for deploying our ROBHAZ-DT2 system into real EOD fields.

References

1. T. Iwamoto, H. Yamamoto, "Mechanical Design of Variable Configuration Tracked Vehicle," *J. of Mechanical Design*, vol. 112, pp. 289–294, 1990.
2. G. W. Kohler, M. Selig, M. Salaske, "Manipulator Vehicle of the Nuclear Emergency Brigade in the Federal Republic of Germany," *Proc. of 24th conf. On Remote System Technology*, pp. 196–218, 1976.
3. Y. Maeda, S. Tsutani, S. Hagihara, "Prototype of Multifunctional Robot Vehicle," *Proc. of Int. conf. on Advanced Robotics*, pp. 421–428, 1985.
4. T. Iwamoto, H. Yamamoto, "Stairway Travel of a Mobile Robot with Terrain-Adaptable Crawler Mechanism," *J. of Robotic Systems*, vol. 2, No. 1, pp. 125–134, 1985.
5. K. Yoneda, Y. Ota, S. Hirose, "Development of Hi-Grip Crawler using a Deformation of Powder," *J. RSJ*, vol. 15, No. 8, pp. 1188–1193, 1997.
6. H. Schempf, E. Mutschler, C. Piepgras, J. Warwick, B. Chemel, S. Boehmke, W. Crowley, R. Fuchs, J. Guyot, "Pandora: Autonomous Urban Robotic Reconnaissance System," *Proc. of Int. conf. on Robotics and Automation*, vol. 3, pp. 2315–2312, 1999.
7. H.R. Choi; Ryew, S.M, "Anthropomorphic joint mechanism with two degrees of freedom," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1525–1530, 2000.
8. Lung-wen Tsai, *Robot anaysis, The mechanics of serial and parallel manipulators*, A Wileys-Interscience Publishing, 1999.
9. Gosselin, C. and Angeles,J., "The optimum kinematic design of a planar three Degree of freedom parallel manipulator," *ASME J.Mech. Transm. Autom. Des.*, vol. 110, pp. 35–41, 1988.
10. J.A. Carretero, M. Nahon and R.P. Podhorodeski, "Workspace analysis of a 3-dof parallel mechanism," *Proc. IEEE Int. Conf. on Intelligent Robots and System*, pp. 1021–1026, 1998.
11. Lung-wen Tsai and Sameer Joshi, "Comparison study of architectures of four 3 degree-of-freedom translational parallel manipulator," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1283–1288, 2001.
12. E. Ottaviano, C.M. Gosselin, and M. Ceccarelli, "Singularity analysis of CaPaMan: A three-degree of freedom spatial parallel manipulator," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1295–1300, 2001.

Towards Safer Roads by Integration of Road Scene Monitoring and Vehicle Control

Lars Petersson¹ and Alexander Zelinsky²

¹ National ICT Australia Limited

Locked Bag 8001

Canberra ACT 2601

lars.petersson@nicta.com.au

² Robotic Systems Laboratory

Department of Systems Engineering, RSISE

The Australian National University

Canberra, ACT 0200 Australia

alex@syseng.anu.edu.au

Abstract. This paper introduces the Autonomous Vehicle Project (AVP) at the ANU together with a discussion and an example of driver assistance systems. A set of necessary core competencies of such a system is identified and in particular a system for force-feedback in the steering wheel when departing from a lane is presented. A system like this is likely to reduce accidents due to driver fatigue since unintentional lane changes become more difficult. The presented system utilises a robust lane tracker which is experimentally evaluated for the purpose of driver assistance.

1 Introduction

It is the belief that changes in the way we use our cars are going to be slow and advanced technology will enter as small sub-systems that solve well defined tasks. We have in fact already seen this, e.g. cruise control, power-steering, ABS-brakes etc. A variety of well defined tasks can be found that are in line with the research performed within AVP [1,4]. The direction of the research of the ANU's Autonomous Vehicle Project (AVP) is hence to investigate supporting systems. One of them is to give force-feedback in the steering wheel when an attempt is made to change lanes when monitoring systems have detected a potentially dangerous situation. Systems which perform these types of supporting tasks can generally be called *Driver Assistance Systems (DAS)*. This particular task could potentially reduce the number of accidents due to driver fatigue. The example above demonstrates the aim of the research, i.e. the aim is driver support or driver assistance rather than autonomous driving.

Robustness is of paramount importance when creating systems to be used in cars that are driving on public roads. The sensing and detection problem must be solved reliably. Fortunately, roads are designed to be: high contrast, predictable in layout and free of out of context objects. This makes the sensing problem somewhat easier, although by no means trivial. Complementary sensors and algorithms can be used to reduce the likelihood of a catastrophic failure. Various obstacle detection algorithms can be using computer vision, laser range data or radar, whereas the state of the vehicle can be monitored by the help of gyros, accelerometers, odometry and GPS.

The outline of the paper is as follows: Section 2 discusses driver assistance systems in general and lists a set of core competencies to achieve common assistance tasks. Section 3 gives a brief overview of the hardware that is installed in the car and how it is related to the previously listed core competencies. Section 4 mentions some software design issues. In Section 5, a robust lane tracker using a novel approach to cue fusion is presented. A particular driver assistance mechanism providing force-feedback for the steering wheel to prevent unintentional lane departure is outlined in Section 6. Finally, there is a conclusion and future work to be carried out.

2 Driver Assistance Systems

A Driver Assistance System (DAS) may perform activities like relieving the driver of distracting routine activities, warn about upcoming situations and possibly take control of the car if an accident is imminent. Depending on the task to be performed, a DAS must have the appropriate levels of competencies in a number of areas. If we, for a moment, consider the DAS to be a human co-pilot it is easier to pick out the important aspects. To be of any help, the co-pilot would have to be aware of what is going on outside of the car, e.g., are there any pedestrians in sight, where are they going, how is the road turning etc. Moreover, we would like our co-pilot to warn us if we have not noticed an upcoming situation. That means that not only should the co-pilot be aware of what is going on outside of the car, but also what is happening inside, i.e. the driver's responses. In addition, our co-pilot must know where the vehicle is going, how fast, if we are braking, accelerating etc. to make good calls. Further on, good calls is a result of good reasoning. A successful team of a driver and his co-pilot requires good communication between them. The co-pilot should not be intrusive and present the driver with too much information. Finally, if the co-pilot notices that the driver does not respond properly to a situation and there will therefore be an accident, he must be able to take control over the car.

Returning to our non-human co-pilot, the DAS, we can condense the above to having the appropriate level of competencies in categories that include the following:

- Traffic situation monitoring
- Driver's state monitoring
- Vehicle state monitoring
- Communication with the driver
- Vehicle control
- Reasoning system

The first three collect information which the DAS can use to analyse the current situation. The fourth, *Communication with the driver*, can be used both as input to the DAS and output to the driver. The driver can e.g. specify an overall goal or the DAS can give information to the driver. *Vehicle control* is of course necessary if it is expected that the DAS should be able to perform any semi- or fully autonomous maneuvers. A *reasoning system* may in the simplest cases consist of a direct mapping from an input to an output, whereas in the complex cases the latest advances in artificial intelligence (AI) might be utilised. Naturally, the level of competence in each category is dependent on the specific task to be solved. The example force-feedback system presented in Section 6 does not e.g. use information about the driver's state or have an advanced reasoning system.



Fig. 1. The research platform. A 1999 Toyota Landcruiser. It is equipped with the appropriate actuators, sensors and computing power.

3 Hardware Overview

The platform that is used in the project is a 1999 Toyota Landcruiser 4WD. It is equipped with the appropriate hardware to provide an environment in which the competencies identified in the previous section can be implemented. In detail, **traffic situation monitoring** can be performed using the active vision head, called CeDAR, which has previously been developed at the ANU [10]. The CeDAR is looking out through the windscreen, monitoring the road scene ahead of the car. Further on, a SICK laser range sensor and a millimeter radar mounted at the front of the car are also useful for monitoring the traffic situation. Inside the car, there is another stereo camera pair **monitoring the driver's state**. It is directed toward the driver and is tracking the gaze direction, head pose, blink rate etc. The tracking is performed using the faceLAB system from Seeing Machines (see Section 4.1). The faceLab system is particularly suitable to monitor driver fatigue. The torque with which the driver is turning the steering wheel is also monitored by the use of strain gauges on the steering axis. The **vehicle state monitoring** is made possible by a GPS, a three axis gyro, a three axis accelerometer, the odometer of the car and a sensor measuring the steering angle. **Vehicle control** is carried out by actuators that control throttle, steering and braking. In addition, an internal Ethernet network connects a number of computers which provide the appropriate computing power. Currently, the normal configuration consists of three Pentium III computers. Two of them are dedicated to performing computer vision tasks, whereas the third controls the active vision head and other, less computationally intensive tasks. A **reasoning system** can be implemented and run on any of the computers. Efficient **communication with the driver** is a competence that is not provided for at the moment, other than what is available on computer screens. It is not clear how information should be communicated to the driver in an efficient way. In fact, it is a field of research in its own and is out of scope for the effort within AVP.

4 Software Overview

The software is generally organised according to a client/server paradigm to achieve flexibility in terms of experimental configurations, code reuse and software development. New applications can easily use a number of resources such as GPS, stereo

head and steering control to mention a few. Interprocess communication is performed using CORBA as implemented by omniORB. omniORB is an open source implementation of the CORBA protocol which proved to be simple and easy to use in comparison with other implementations. The use of CORBA further increases the flexibility since components running under different operating systems and/or written in different languages can still communicate with each other using exactly the same interfaces. By using CORBA, a particularly important aspect of the software design process is gained and that is *maintainability*. The interfaces become well defined and CORBA itself is very well documented.

4.1 faceLAB

faceLAB [12] is a driver monitoring system commercialised by Seeing Machines [9] based on research and development work between ANU and Volvo Technological Development Corporation. It uses a passive stereo pair of cameras mounted on the dashboard of the vehicle to capture 60Hz video images of the driver's head. These images are processed in real-time to determine the 3D position of matching features on the drivers face. The features are then used to calculate the 3D pose of the persons face ($\pm 1\text{mm}$, $\pm 1\text{deg}$) as well as the eye gaze direction ($\pm 3\text{deg}$), blink rates and eye closure.

5 A Robust Lane Tracker

A robust lane tracker has been developed to monitor the lateral offset relative to the lane. Despite many impressive results from lane trackers in the past [2] [3] [11] [13], it is clear that no single cue can perform reliably in all situations. The lane tracking method used here is based on a novel method for target detection and tracking that combines a *particle filter* [6] with a cues fusion engine which is suitable for both low and high dimensional problems. The algorithm is robust, easily extended and self-optimised to maximise the use of the computational resources available. The basis of the algorithm, is that multiple cues are utilised to search for a target, but their performance over time is evaluated and the set of cues that are performing optimally are *distilled* to a select few that can run at frame rate. The rest of the cues are maintained at speeds less than frame rate so that their results can also contribute to the overall tracking progress and so that they can be reinstated to run at frame rate if their results improve. A complete description of this algorithm has been published in [7]. The state space for the particle filter is the lateral offset of the vehicle relative to the skeletal line of the road, the yaw of the vehicle with respect to the skeletal line and the road width (see Figure 2). The cues chosen for the force-feedback system in Section 6 were designed to be simple and efficient while being suited to a different set of road scenarios. Individually, each of the cues would perform poorly, but when they are combined through the cue fusion process they produce a robust solution to lane tracking. Each cue listed below uses the road model shown in Figure 2 to process the probability of each hypothesis from the particle filter. In Figure 3, cue number one to four below is depicted with the corresponding set of particles. Cue

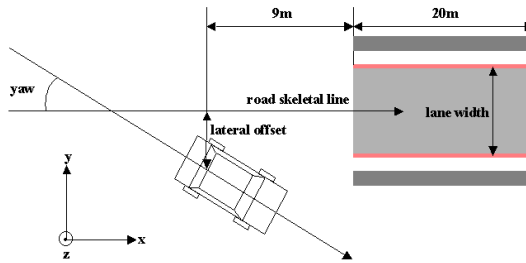


Fig. 2. Road model used for the particle filter. The dark shaded region is used as the non road boundary in the colour cues while the light shaded region is the road region. Note that the figure is exaggerated for clarity.

number five and six, which are not based on any visual input, are not shown. The pictures in Figure 3 were derived from an image sequence acquired during a test run.

1. **Lane Marker Cue** is designed for roads that have lane markings. A modified ternary correlation to preprocess an intensity image of the road and the cue returns the average value of the pixels along the hypothesised road edges.
2. **Lane Edge Cue** is suited to roads with lane markings or roads with defined edges. It uses a preprocessed edge map and returns the average value of the pixels along the hypothesised road edges.
3. **Road Colour Cue** is useful for any roads that have a different colour than their surroundings (both unmarked and marked roads). It returns the average pixel value in the hypothesised road region from a colour probability map that is dynamically generated each iteration using the estimated road parameters from the previous iteration.
4. **Non Road Colour Boundary Cue** is the opposite to the Road Colour Cue and returns the average road colour probability of the non-road regions.
5. **Road Width Cue** is particularly useful on multi-lane roads where it is possible for the other cues to see two or more lanes as one. It returns a value from a Gaussian function centered at a desired road width given the hypothesised road width. The desired road width used in this cue was 3.61m which was empirically determined from previous lane tracking experiments to be the average road width.
6. **Elastic Lane Cue** is used to move particles toward the lane that the vehicle is in. It returns 1 if the lateral offset of the vehicle is less than half of the road width and 0.5 otherwise.

The lane tracker was tested in several different scenarios including:

- Highway driving with light traffic.
- Outer city driving with high curvature roads.
- Inner city driving with moderate levels of traffic.

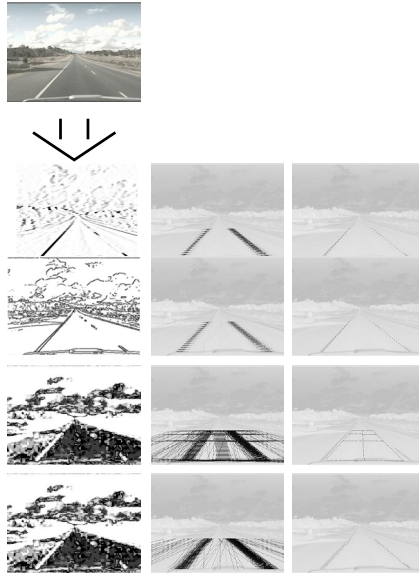


Fig. 3. The top left image shows the video input to the lane tracker. The left column below shows preprocessed images which acts as input to the different particle filters. In the middle column, the current set of hypotheses are overlayed in image space. The most likely hypothesis, or particle, for each cue is then shown in the rightmost column. The order of the cues from top to bottom is: Lane Marker Cue, Lane Edge Cue, Road Colour Cue and Non Road Colour Cue.

Figure 4 shows the output of the lane tracker in the above scenarios using the 6 different cues. The lane tracker was found to work robustly and solved the problems typically associated with lane tracking including:

- Dramatic lighting changes (see A in Figure 4).
- Changes in road colour (see D-F in Figure 4).
- Shadows across the road (see C,G-H in Figure 4).
- Roads with miscellaneous lines that are not lane markings (see I in Figure 4).
- Lane markings that disappear and reappear.

This can be attributed to the combination of particle filtering and cue fusion. Because of the particle filter, cues only have to validate a hypothesis and do not have to search for the road. This indirectly incorporates a number of *a priori* constraints into the system (such as road edges meeting at the vanishing point in image space and the edges lying in the road plane) which assist it in its detection task. Cue fusion was found to dramatically increase the robustness of the solution due to the variety of conditions the cues were suited to. The final configuration of cues in the system is a direct result of earlier experiments uncovering certain conditions in which the cues would fail.



Fig. 4. Results from the lane tracker. The boxes indicate the end of the lines that mark the tracked lane.

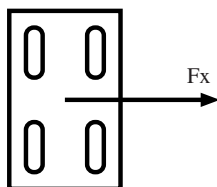


Fig. 5. A virtual control force, F_x , applied to the center of gravity of the car.

6 A Force-Feedback System

We would like to be able to automatically prevent unintentional lane changes. As a means of detecting whether a lane change is intentional or not, information from the traffic situation monitoring system can be utilised in conjunction with the driver monitoring system. However, this is out of scope for this paper. For the purpose of demonstration, it is assumed below that intentional lane changes are those where the driver turns the steering wheel with a torque greater than a certain threshold. The applied torque is measured by strain gauges on the steering axis. Preventing unintentional lane changes can then be achieved by monitoring of the lateral offset relative to the center of the lane, and use this information to control the car in such a way as to keep the car within the limits of the lane.

A car that is controlled in a steer-by-wire fashion has the ability to fuse the control commands from the driver with the control commands from the DAS in an arbitrary way. Moreover, the driver experience can be engineered to suit the individual. However, the platform that is used in this paper, has a standard steering system where the steering wheel is physically connected to the wheels. This introduces a constraint, where a trade-off has to be made between the performance of the lane keeping and the driver experience. The desired behavior of this force-feedback system is not to entirely prevent lane changes. Instead, a weak repulsive force should be encountered as the car drifts away from the center of the lane. As mentioned above, the driver may want to overcome the repulsive force if he intentionally wants to enter the other lane while, e.g., overtaking another car. In [8], a virtual force framework for lateral vehicle control is developed which is powerful as it allows superposition of virtual forces to the car that can represent control inputs from a range of different kinds of DAS. This is also exploited to some extent in [5]. The framework assumes

steer-by-wire control of the car so that control commands from the driver and the DAS can be fused arbitrarily. However, the concept of virtual forces can also be used on the platform in this paper, with the difference that the control originating from the DAS will be, through the mechanical connection, fed back to the driver.

Below is a control method outlined which uses information from the robust lane tracker, which was described briefly in Section 5. An appropriate control force, F_x , is calculated that is then used to derive the actual control command as described in [8]. The control force F_x is calculated using a potential field approach, where for the purpose of demonstration, a very simple shape of the potential field is chosen. This approach will generate a control law that can be tuned to conform to the desired behavior mentioned above. Suppose we would like our car to drive in a virtual valley as is shown in Figure 6. That can be accomplished by calculating a virtual force as a function of the lateral offset which then acts on the car.

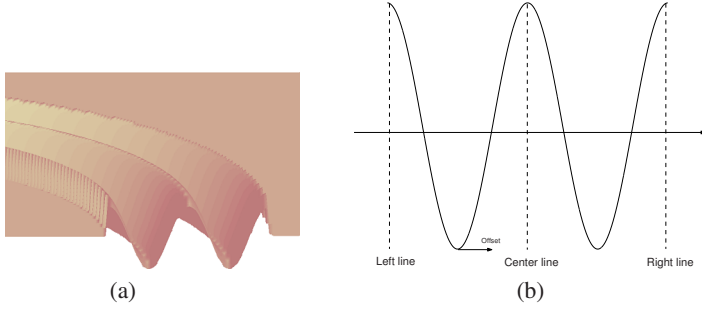


Fig. 6. (a) The potential field creates virtual valleys which will keep the car inside the lane it is currently driving in. The picture shows the potential field as the car drives on a curved section of road with two lanes. (b) A potential field, V , for a two lane road. The centers of each lane are at the two local minima. The offset can then be used in conjunction with the potential field to control the applied torque to the steering shaft of the car.

The potential field in Figure 6 has a sinusoidal cross section. The potential field can be generated by:

$$V = -\cos(l_{offset}\pi/l_{LW}) \quad (1)$$

A virtual force that acts on an object on the curve, assuming straight section of road and largely aligned with the lane, can then be calculated as being proportional to the derivative of Equation 1 with respect to the offset:

$$V' = \pi/l_{LW} \sin(l_{offset}\pi/l_{LW}) \quad (2)$$

A control law that will exert forces on the car in the lateral direction can then use the derivative in Equation 2.

$$F_x = -k \sin(l_{offset}\pi/l_{LW}) - D(l_{offset}, l'_{offset}) \quad (3)$$

where k is the gain. The control law is augmented with the term $D(l_{offset}, l'_{offset})$ which is a damping. A suitable damping term will prevent the car from oscillating back and forth in the valley. Strain gauges are used to monitor the torque the driver applies so that the driver can always override the system if necessary.

The specific potential field that was chosen, Figure 6, may of course be exchanged for something that is more intuitive to the driver. In Figure 7, the potential field makes it more difficult to departure from the road than to cross lanes which is natural. Moreover, there is a flat section in the center of each lane that gives the driver complete control over the vehicle.



Fig. 7. A potential field which is more intuitive to the driver than the one in Figure 6. Flat sections in the middle of each lane where there is no interference with the driver's intention and it is more difficult to departure from the road than to cross lanes.

It is worth noting that the approach allows extensions in the form of other complementary DAS such as e.g. stability control. Care must, however, be taken so that the sum of superimposed potential fields does not have local minima at the wrong places.

7 Conclusion and Future Work

In this paper, driver assistance systems were discussed. A number of necessary core competencies were identified by a comparison with a human co-pilot. These include traffic situation monitoring, driver's state monitoring, vehicle monitoring, communication with the driver, vehicle control and a reasoning system.

A particular example of a driver assistance system was also presented, a system for force-feedback in the steering wheel when crossing lanes. A force-feedback system like that is, e.g., likely to reduce accidents due to driver fatigue since unintentional lane changes become more difficult. The presented system controlled the lateral position using a virtual force framework [8] in combination with a potential field. Monitoring of the lateral offset was performed by a robust lane tracker which was experimentally evaluated for the purpose of driver assistance. The lane tracker used a method based on particle filters to fuse a number of complementary cues.

Future work consists of increasing the level of competence in each of the categories mentioned above. The aim of this project is to fully monitor events that occur both inside and outside of the vehicle and make use of this information to create driver assistance systems capable of reducing accidents. This includes monitoring driver head pose and eye gaze directions, eye blink rate and closure, lane tracking, obstacle detection and vehicle state.

Acknowledgement

National ICT Australia is funded by the Australian Department of Communications, Information & Technology & the Arts and the Australian Research Council through Backing Australia's ability and the ICT Centre of Excellence Program.

References

1. Nicholas Apostoloff and Alexander Zelinsky. Vision in and out of vehicles: integrated driver and road scene monitoring. In Bruno Siciliano and Paolo Dario, editors, *Experimental Robotics VIII*, Advanced Robotics Series. Springer-Verlag, 2002.
2. Parag H. Batavia, Dean A. Pomerleau, and Charles E. Thorpe. Overtaking vehicle detection using implicit optical flow. In *IEEE Transport Systems Conference*, 1997.
3. Ernst D. Dickmanns. An expectation-based, multi-focal, saccadic (ems) vision system for vehicle guidance. In *International Symposium on Robotics and Research*, Salt Lake City, Utah, October 1999.
4. Luke Fletcher, Nicholas Apostoloff, Jason Chen, and Alexander Zelinsky. Computer vision for vehicle monitoring and control. In *Australian Conference on Robotics and Automation*, 2001.
5. J. Christian Gerdes and Eric J. Rossetter. A unified approach to driver assistance systems based on artificial potential fields. *Journal of Dynamic Systems, Measurement and Control*, 123(3):431–438, September 2001.
6. M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision* 92(1), pages 5–28, 1998.
7. Gareth Loy, Luke Fletcher, Nicholas Apostoloff, and Alexander Zelinsky. An adaptive fusion architecture for target tracking. In *The 5th International Conference on Automatic Face and Gesture Recognition*, Washington DC, May 2002.
8. Eric J. Rossetter and J. Christian Gerdes. A study of lateral vehicle control under a 'virtual' force framework. In *Proceedings of the 2002 AVEC Conference*, Hiroshima, Japan, 2002.
9. Seeing Machines. Facelab face and eye tracking system. <http://www.seeingmachines.com>, 2001.
10. Orson Sutherland, Harley Truong, Sebastien Rougeaux, and Alexander Zelinsky. Advancing active vision systems by improved design and control. In *Proceedings of International Symposium on Experimental Robotics (ISER2000)*, December 2000.
11. A. Suzuki, N. Yasui, N. Nakano, and M. Kaneko. Lane recognition system for guiding of autonomous vehicle. In *Proceedings of the Intelligent Vehicles Symposium, 1992*, pages 196–201, 1992.
12. Trent Victor, Olle Blomberg, and Alexander Zelinsky. Automating driver visual behaviour measurement. In *Vision in Vehicles 9*, 2001.
13. Todd Williamson and Charles Thorpe. A trinocular stereo system for highway obstacle detection. In *International Conference on Robotics and Automation (ICRA99)*, 1999.

Performing Skilled Work with an Interactively Operated Service Robot

Aarne Halme, Jouni Sievilä, Ilkka Kauppi, and Sami Ylönen

Helsinki University of Technology, Automation Technology Laboratory,
PL5500, 02015 HUT, Finland
aarne.halme@hut.fi
www.automation.hut.fi

Abstract. In the paper we consider skilled tasks that are performed by a human-like robot in outdoor unstructured environment when working interactively with a human operator. As skilled tasks we understand tasks, which - like when humans perform them - are non-trivial and demand learning or training. Typically such tasks may also fail and require several trials to be successfully executed. Performing requires use of senses and development of a unique mini-plan, which takes into account the present situation. The mini-plan, which may also be called "skill", produces a sequence of unit operations needed for successful performing of the task. The paper discusses preliminary experiences on implementing and experimenting such skilled tasks with a mobile centaur-like service robot.

1 Introduction

In the paper we consider skilled tasks that are done by a centaur-like robot in outdoor unstructured environment when working interactively by a human operator. As skilled tasks we understand tasks, which - like when humans do them - are non-trivial to perform and they demand learning or training before can be successfully executed. Typically such tasks may also fail and require several trials. Performing requires use of senses and creation of a mini-plan, which takes into account the present situation. The mini-plan, which may also be called "skill", produces a sequence of unit operations needed for successful performing of the task. The sequence is not unique and may vary from one case to another depending on e.g. the information currently available. A similar approach has been considered in [1] in the case of navigation with event maps. As an illustrative example one may consider gripping and carrying objects whose size, weight and shape are not known beforehand. The robot may first try to classify the object by remotely sensing it (like a human does), then to find out its weight by pushing it or trying to lift it, and finally finding the best gripping policy for carrying it. One may rather easily produce a mini-strategy, which in principle solves the problem; i.e. controls the motions properly when perceptual data is available. In practice, however, to make such a strategy operational much learning or training is needed, i.e. the detailed structure of the unit operations and related parameters has to be found out. In addition to be operational, the strategy might also be further optimized through practicing against some proper criteria, for example execution time, flexibility of motions or energy conservation.

In what follows we propose an approach how to develop a robot control system for skilled tasks. The test robot is WorkPartner, shown in Fig. 1, introduced earlier in FSR 2001 [2]. WorkPartner is an interactive service robot designed to work mostly outdoors together with its user. It is a centaur-like mobile platform, which has a human-like two-hand manipulator. The human-robot interface utilizes cognitive properties of both the human operator and the robot. It means that commanding and communication between the two entities utilize perception and conception of the presence where they both currently exist. The presence is a virtual model representing geometrical features of the surrounding world augmented by an object data base representing physical and conceptual objects [3].

The ideas presented in the paper are demonstrated by a task where the robot surveys and map an unknown area including holes marked with colored signs. At each hole it locates the hole in the local coordinate system and probes it with a stick-like sensor. The task imitates measuring the 3D characteristics of a drill hole field in a mining or construction work site.



Fig. 1. WorkPartner robot

2 Control Architecture of the Robot

The overall control architecture of WorkPartner robot is illustrated in Fig. 2. The architecture allows interactive operation between the user and the robot. The user introduces tasks by the aid of a high level language (WoPa-language in the figure). The syntax of the language includes commands and objects that are related to the presence, which means in practice the physical environment, objects in it and their possible attributes. The objects can be either physical objects or targets for operations focusing to certain area of the physical world. The presence is modeled as an object database, which can be illustrated as a 2D map showing the objects in their positions. Objects are boxes of different size carrying a nametag and list of content. The size of the box is related to the size of the object(s) it covers. The program called "manager" interprets the message of the user and plans the actions needed to execute the tasks. The output of the plan is a sequence of commands of an intermediate language which controls the actions of different subsystems of the robot [4]. The intermediate language consists of a set of commands, which are executed in a certain controllable way. They may utilize measurements from several sensors during execution. A different combination of these commands enables the robot to perform complex and simultaneous tasks with the platform and the manipulator.

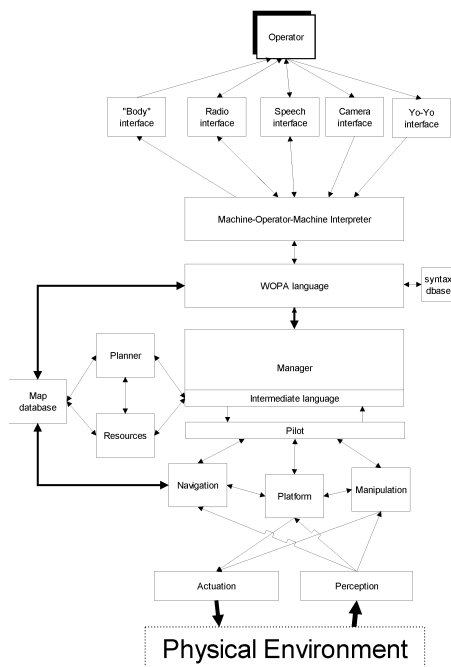


Fig. 2. Overall control architecture of WorkPartner robot

2.1 Illustrative Examples on Interactive Tasks

Let us consider two examples of interactive tasks, which can be performed in the above control architecture. First a simple one. The robot can follow the operator when it recognizes him/her from a color coat by using the head camera and the laser scanner on its chest. The operator wears a color coat as a part of the Human Robot Interface (HRI) to be identifiable as the user of the robot. The robot measures the current direction to operator by tracking him/her by the camera and finds the current distance by the scanner. The user command "follow me" is interpreted and the corresponding action is planned as two intermediate language commands "find target" and "follow target". "Find target" start camera scan to find targets with a specific color around the robot. When finding one it fixes to that target. The specific color is the one of the operator's jacket, which is defined as a rare color in the color map to make it different from other colors in the surrounding. "Follow target"-command moves the robot when the operator moves in such a way that a certain distance between them remains.

A much more complicated task is picking up an object. The user command could be "fetch the ball", which includes the action part "fetch" and the object part "ball". When the planner start making the plan for execution the command the action part is first interpreted so that the robot has to go to the place where the ball is, take it and bring it back to the user. Next the object "ball" has to be identified, which means that it is recognized in the presence. The simplest case is that there is only one object

named "ball" and its position is known, i.e. a box including the object "ball" exists in the presence model. Then the planner needs to plan only the route to the box and after that the gripping operation, which in this case is easy because a ball shape object can be gripped between the hands from every direction. The ball has to be found, however, either by the camera or the laser scanner before the gripping can be controlled. So, the planner initiates these sensors when closing the area where the ball is. The situation is more complicated, if there is more than one ball in the presence database or the position is not known. Then the robot has to go to the interactive mode and ask more information from the user. Suppose the position is not known exactly and the user gives it in the form "near lamppost", where the lamppost is a known object in the presence. The planner first plans the route to the lamppost and then starts searching task to find the ball. The overall plan, which takes into account all alternatives, can be made rather easily. Conversion to a sequence of intermediate language commands can be also made, but in this case only on-line during execution, because the plan includes several branching points and loops depending on the observed data. The intermediate language controller monitors the execution of the commands and prohibits the actions that possibly could harm the robot.

3 Learning and Performing Skilled Tasks

The control architecture allows performing of skilled task and learning by implementing such features into the structure of the planner. The underlying idea is that skilled tasks are represented as executable sub-plans, which are joined to the master plan when the task is called. A sub-plan is a dedicated procedure for the skilled task, which is trained separately under the supervision of the operator or under a reinforcement learning procedure. Each sub-plan is represented as a chain of automatically controlled unit operations, which are executed interactively under the robot perception system. Learning is directed both to the structure of the chain and to the parameters of the unit operations. In the current version of the control architecture the planner includes only explicitly solved (planned with all alternatives) individual commands of the WoPa-language.

Several classical methods and approaches are available in the systems theory for learning/training, but most of them are not feasible in tasks considered here. The problem is that the kinematics/dynamics of mobile robots operating in unstructured 3D environment are extremely complicated. Motions during execution cannot be described with a unique accurate enough mathematical model. Every task, when executed, is an individual operation in details. The number of unit operations needed for a skilled task may vary and is usually relatively high. The parameter space for learning should be able to describe at the same time both the execution strategy of the task and tuning of individual motion trajectories. Learning methods, which are based on optimizing the parameters of the input-output mapping (like Neural Nets), are not feasible, because structural changes in the behavior cannot be easily considered. A quite feasible approach for such complex optimization problems is evolutionary learning based on genetic algorithms or similar methods. Genetic algorithms allow direct searching in parameter space, which may describe the original input-output relation in a complex non-mathematical way. It suits well for configuring the

program blocks executing the mini-strategy of a skilled task. A central issue is to choose a proper fitness function. There are many alternatives, for example execution time or the number of trials and errors needed for a successful execution. It looks like that the execution time is much favored among biological creatures and it is also easy to calculate in the case of robots. However, constraints like power and perceptive resources have to be taken into account.

Learning by genetic algorithms is based on guided stochastic searching, which might need several tens of generations to become close enough to a sensible solution. With a robot it means that most of the parameter sets at the beginning of searching do not generate a sensible behavior. Although irrelevant parameter combinations (chromosomes) are relatively easy to strike out by observing the behavior of the robot the time needed might be too long for practical use. In the starting phase learning the aid of the operator can be used. This means in practice that the operator runs the robot through the mini-strategy of the task and selects the initial generation by finding such parameter sets most of which produce a sensible behavior allowing execution of the task within a finite time. The genetic algorithm is then used only for fine-tuning of the parameter sets. When making the crossover operation fixed schemata can be applied to guarantee that irrelevant behaviors are not run across too often.

As an illustrative example let us consider a "picking a box"- task. The robot initially knows only that the object is box-type, not its size or weight. The task is designed to larger objects, which can be gripped and carried between the hands, as illustrated in Fig 1. The robot initializes the task when it is close enough to the place, where it supposes the box exists, by starting finding it. Box-like objects can be found easily by camera. Camera searching includes a few parameters, like scanning sector and speed, matching model parameters, etc. In the case the object is not found, further instructions are asked from the operator. Suppose the box is found. The next step is to approach it to a proper distance from a proper direction. For this purpose the laser scanner is first used. The size and pose (position + attitude) of the box with respect to the robot can be determined. Most probably the robot has to look the object from a couple of different directions, which demands motion relative to the object. Using a standard pattern, which includes a few parameters, like distances, angles and speeds, it can do this. After getting a more precise idea about the box the next step is to approach it for gripping. This can be done again with a standard procedure, which includes the approaching speed as the main parameter. The local terrain might be different in different cases and the speed needs to be adjusted for the current situation. The starting speed is the maximum one leading to successful performance during the learning process. In addition, some maneuvering might be needed to get the robot to the right direction for gripping. The gripping is done by clamping the box between the hands using a preset force. Now, because the weight is not known, gripping might fail if the box is too heavy. Two possibilities exist to continue the task. In the case, the robot can lift the box, but hands are slipping, the clamping force might be increased until the box is firmly gripped. In the case, the box is too heavy for the back joint, which is weak like the one of humans, the robot might try lifting by using simultaneously the leg and back joints [2]. The mini-strategy includes in this case five micro-tasks, which can be programmed and controlled by certain key parameters.

4 A Case Study

As a case study, a surveying problem was studied, which includes a probing action at each found target location. The problem can be considered in a more generic framework as a "search and probe"-problem, where certain types of objects are being searched within a limited environment, located and studied in details. It simulates e.g. surveying a drill hole field in a construction or open query site. A drill hole pattern has to be drilled before explosives are loaded and the rock is exploded. In the modern construction and mining technology the dosage of explosives and consequently resulting rock separation can be made very accurate provided the geometry of the drill hole pattern is known in 3D. For this purpose the geometry of drilled holes should, however, be measured and an accurate enough map of the drill field created. This has to be done after drilling because drill holes can make unpredictable trajectories inside the rock. The problem under studying is how the necessary surveying operation, i.e. accurate location of the holes and measuring of their 3D profile, could be made by a mobile service robot instead of a man. Searching and measuring are skilled tasks, because the terrain on which the robot moves can vary considerably, each hole has to be found and measured in spite of difficulties related to individual holes. The WorkPartner robot is not an optimal robot designed for this task, but it could do it in practice. The results presented below are from laboratory tests preceding field tests.

When converting the exploration and surveying operation to the robotic scenario roughly the following task description is obtained. First, the robot has to move to the field and find a proper corner of the drill field where to set the local co-ordinate origin. This is pointed by the operator who transports the robot to the field. The drilled holes are covered by plastic hubs with red color. The operator places cylindrical passive beacons around the field (places not fixed nor measured) to help the local SLAM operation done by the aid of the laser scanner (in the breast of the robot). The robot explores the field, finds the holes and measures each of them by the aid an inertial probe. The probe is a stick-like device, which has to be put in the hole, let it go down to the bottom and lift it then up. The 3D-profile measurement is done during the down and up going movement of the probe. The map of the drill field is made in the local co-ordinate system.

The holes are found by the aid of the head camera of the robot. The head also includes a range laser (parallel to the camera) by which the hole coordinates can be transferred to the manipulator base coordinate system. The laser range scanner at the breast of the robot is used for localize the robot and the holes in the local coordinate system. The beacons, which are set manually around the drill field, can be recognized also by the laser scanner. Surveying is based on a SLAM algorithm, which utilizes laser odometry and simultaneous position estimation of the beacons and the robot. The 3D-probing operation is done by the aid of a special inertia probe (not used in this study). The scenario includes three skilled tasks. One is finding the holes, other is to localize them and the third one is to make the probing operation. Probing includes approaching a hole, placing the probe into the hole and performing the measurement.

4.1 Finding Drill Holes

Red plastic cones that prevent rocks to fall into the hole cover all drill holes. The robot can see these cones because their color differs from the background. The sensor used for cone finding is a CCD-video camera. The recognition uses the color information of the image. The image is first changed from RGB to HSV color format. The image is then further divided into three channels, hue, saturation and value. The first channel is most important for finding the red color. Certain thresholds are set to all three channels and the combination of the corresponding images is formed resulting a binary image, in which cones can be detected as white objects as shown in Fig. 3. The holes are drilled roughly in the form of a pre-designed grid. The robot has to find all holes, which makes the searching task skill demanding. Because the rocky terrain might be very difficult the holes cannot be drilled always in the intended places. If a hole is not found in the place where it should be, searching must be extended to the directions, where it most probable could be found. This means extra motion with a slower speed and stopping time to time to look around.



Fig 3. Finding drill holes by a color camera.

4.2 Measuring the Position of a Drill Hole

The relative position of the drill hole with respect to the robot body fixed coordination system is measured with the range laser. This laser is situated next to camera on the robot head and points along the optical axes of the camera. The head is controlled with a pan-tilt unit. When there is a red cone in the image, the robot turns its head to the direction where the object situates at the center of the image. The distance of the cone from the robot is measured with range laser. When knowing the distance and pan and tilt values of the head the position of the cone in the robot body coordinate system can be calculated. All the drill holes that are found during the surveying operation are situated in a dynamic map. This is done by the aid of a SLAM algorithm, which utilizes the laser scanner and artificial cylindrical landmarks set around the drill field. The SLAM-algorithm is automatic, but the cone positioning needs skilled operation because the laser beam has to hit the cone. Shooting the cone needs small head movements, which are defined by a couple of parameters.

4.3 Moving to the Hole and Making the Measurement

When the robot comes to a hole that is not yet measured it first moves towards the hole with a preset slow speed. While moving it follows the red cone with the camera. A new more accurate measurement of the relative position of the hole is done from approximately 1,5 meters distance. Robot moves closer to the hole and measures the hole position ones more. From the latest measurement it knows the position of the center of the hole and is able to move its hand to the place that is straight up from the hole. From this position the probe can be let down into the hole. The probe is stick-like device including inertia sensors. In laboratory tests the actual probe is replaced with a wooden stick. The robot let the probe to the hole by the aid of a winch and takes it up. What make this task skill demanding are the sequences of unit operations for approaching the hole and for setting the probe to the hole. Both have speed parameters, which have to be tuned for optimal performance.

5 Conclusions and Discussion

Evolution of robots capable to interactive work with humans is still in its early phase. So is our study. Experiments done this far show, however, that the approach chosen to establish the cognition-based communication between the user and the robot in one hand, and to structure the skilled work by using parameterized unit operations on the other hand, is fruitful. By the aid of an interactive HRI, which utilizes both the real and virtual presence, the user can load the robot to work without being himself loaded too much. The structure of skilled work tasks seems also suitable for learning during working, although we do not have enough experience to prove it yet.

The planner which interprets the human -robot communication and plans the required command sequences for the robot subsystems is in the key position when controlling task execution. In our present approach, learning is separated from the planner so that each skilled task is learned separately when its parameters are optimized. This is possible a clumsy way, but it allows learning in small steps and user controlled way.

WorkPartner robot is in its current state an excellent platform for experimental testing of skilled tasks. Its basic software already includes all necessary features to implement non-trivial interactive tasks with perception needs. The research done around this robot is public and can be follow from the web-site www.automation.hut.fi/IMSRI/workpartner/

References

1. Barber, R. and Salichs, M.A., Mobile Robot Navigation Based on Event Maps, *Proceedings FSR 2001 Conference*, Espoo, Finland, 2001
2. Halme, A., Leppänen, I., Suomela, J., Ylönen, S. and Kettunen, I., WorkPartner: Interactive Human-like Service Robot for Outdoor Applications. *The International Journal of Robotics research*, 2003. Vol.22, nr 7-8, pp.627-640
3. Suomela, J. and Halme, A., Novel interactive control interface for centaur-like service robot, *Proceedings IFAC 2002 World Conference*, Barcelona, 2002.
4. Kauppi I., Blom, M. and Lehtinen, H., Motion control language for autonomous vehicle applications, *Proceedings FSR 2001 conference*, Espoo, Finland, 2001.

A Multi-purpose Eight-Legged Robot Developed for an Evaluation of a Neural Interface

Takashi K. Saito¹, Itsuro Saito², Nobuyuki Nemoto³, Koki Takiura², Toshinaga Ozeki², Naoto Kakuta⁴, Takahiro Tohyama², Takashi Isoyama⁵, and Tsuneo Chinzei²

¹ National Cardiovascular Center Research Institute

5-7-1 Fujishirodai, Suita, Osaka 565-8565, Japan

saitotk@ri.ncvc.go.jp

http://www.ncvc.go.jp/english/res/Car_Dyn.html

² Res. center for advanced sci. and technol., Univ. Tokyo

4-6-1 Komaba, Meguro-ku, Tokyo 153-8904, Japan

http://www.bme.rcast.u-tokyo.ac.jp/hongo/Year2002_english.html

³ Grad. Schl. of Inform. Sci. and Technol., Univ. Tokyo

4-6-1 Komaba, Meguro-ku, Tokyo 153-8904, Japan

<http://www.mels.ipc.i.u-tokyo.ac.jp/index-e.html>

⁴ Dept. Mechanical Eng. and Intelligent Sys., Univ. Elec. Commun.

1-5-1 Cyofugaoka, Cyofu, Tokyo 182-8585, Japan

<http://www.ymdlabs.mce.uec.ac.jp/English/index.htm>

⁵ Dept. Biomed. Eng., Grad. Schl. of Medicine, Univ. Tokyo

7-3-1 Hongo, Tokyo 113-8656, Japan

http://www.bme.rcast.u-tokyo.ac.jp/web/Hongo/default_english.html

Abstract. To evaluate a neural interface, which comprises integral microelectrodes for neural recording or stimulation used for the control of human prosthetic devices, such as artificial limbs, a proto-type artificial body is being prepared for implantation of the interface in a rat. As a result of the first trial, new linkage mechanisms were developed, and a robot equipped with the mechanism was able to perform various walks like an animal, including sidestep at high speed by swinging its legs.

1 Introduction

A movement system using the legs is suitable even for machines with wheels or crawlers that move through dirt or mountains. However, at present, there are only a small number of multi-legged walking robots that can be put to practical use because the leg-driving motors require precise control and because low-cost power sources, such as conventional internal combustion engines and high-output commercial motors, cannot be used. At present, robots that walk using legs with a motor at every joint are being widely studied. Although the degree of freedom is high in these robots, mutual and systematic control of the motors is difficult. Moreover, the motors require a contradictory performance, as they require high speed and high torque. Therefore, robots have been toys that walk, and making them practical has not been possible. Conventional robots that walk on legs that have a motor at each

joint (e.g., ERS-110 "AIBO" by Sony Corp.) are able to move in a variety of ways. However, the control of the legs is complicated; the speed of movement is slow; and the load-bearing capacity is less than 30% of the total weight. On the other hand, conventional robots that walk on legs (e.g., "ZOIDS" by TOMY Co., Ltd., or the robots for KAWASAKI robot games [1]), which are driven by linkage mechanisms from a concentrated power source (one or two main motors), have simple structures and control mechanisms. They are suitable for robots that walk on legs and use low-cost integral power from a conventional internal combustion engine or high-output commercial motors; moreover, they move at the same speed as a crawler vehicle, but their mobility is limited, as is that of crawler vehicles. As a result of the trial, new linkage mechanisms were developed, including legs with novel swing mechanisms, a new arrangement for the legs to swing, a control method, and control software for the swing angle of the legs. The novel mechanism combines the advantages of both types of robots and gives adequate performance for practical purposes.

1.1 Neural Interface

The production of human prostheses, such as high-performance artificial limbs, has been a target of applications for robotics using integrated sensors and actuators. A combination of instruments with neural electrodes is thought to be a type of complementary technology [Fig. 1]. For example, artificial cochlear implants, which make use of electrical auditory instruments in conjunction with neural electrodes to stimulate the acoustic nerve, have been used since the 1980s. Another promising approach to these challenges is regenerative medicine, which uses cells. However, there is still much to learn concerning the details of the process that allows the regeneration of sensory organs or limbs, which are made up of different kinds of cells making up complex systems.

The research field that is the focus of our study is related to the development of a neural interface, which consists of integral microelectrodes for neural recording or stimulation and becomes the control terminal of artificial limbs in applied robotics. An artificial body will be prepared to evaluate the interface. It will consist of software that enables a bi-directional conversion of neural and electrical signals and hardware that will enable the electrode to control a body as if it were a living body.

A prototype robot was developed for the evaluation of the interface through experiments using interface-implanted rats. The robot was designed to achieve real-time motion traceability to each of the rat's limbs, to the simulated assorted movements of the rat, and to its load-bearing capacity to carry instruments. To optimize the interface, we can compare the behavior of the robot and the rat. The behavior of the robot and the rat can be compared to optimize the interface. For example, the right paw of the rat moves in tandem with the right hand of the robot, and when the right hand of the robot touches a bar, the rat senses that its own leg has touched a bar. This technology is an extension of our research on electrical stimulation of human sensory nerve fibers concerning tactile sensations received from a robot hand [2]. In the field of such applications, which require real-time motion traceability, there are some partial-model robots that mimic animal bodies [3], [4]; however, there are no

complete whole-body robots that are loaded with actuators, batteries, and processors. The robot presented in this study would satisfy the current need for such a robot.

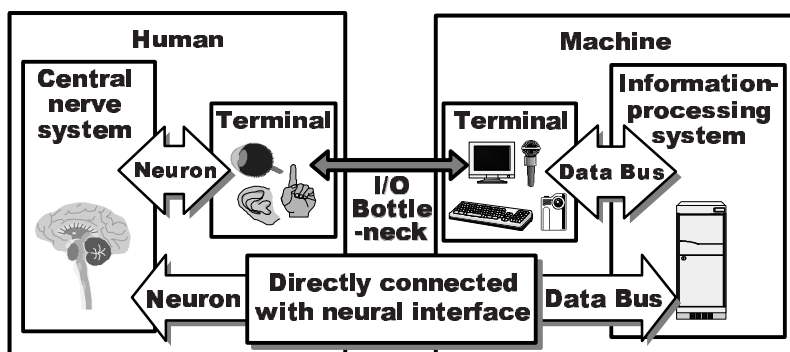


Fig. 1. Neural interface

1.2 Approach from Cell Engineering

The key factor to increase the performance of the instruments used as organ complements is the volume of data transferred by the neural interface, which is still inadequate, similarly to a narrow-band network. The breakthrough on neural electrodes that we have made will help solve the problem of the volume of transferred information described above by working as a broad-band network. The key point is a membrane penetration method for intracellular microelectrodes.

Conventional multi-channel electrodes for recording or stimulating neurons are the extracellular type. The problems of extracellular electrodes are described elsewhere [5]. On the other hand, conventional intracellular electrodes, e.g., micro-glass electrodes and patch electrodes are limited to basic research. This limitation comes from the properties of cell membranes, which are soft and highly fluid. Hence, simple micro-fabricated electrodes could not be inserted into a cell membrane. When such electrodes are inserted by force into a cell, it produces cell death at a high rate. We found that temporary and partial-membrane degeneration resulting from membrane damage, e.g., that by peroxidation, significantly increased the survival ratio of the membrane-penetrated cells. The degenerated membrane could recover after the limited degree of damage inflicted on it by either of two processes: physically, by resealing itself as a result of membrane fluidity, or chemically, using the antioxidant system or metabolism [6]. The cell engineering done for this study was in the field of microinjection [5]. A high survival rate that was close to 100% of the injected nerve cells was achieved. Various injected functional molecules, fluorescent dyes, antibodies, and mRNA as well as their functions were preserved in the cells [7]. The research on cell engineering is progressing along these lines; however, specific

aspects are being used to develop an integrated intracellular neural electrode for peripheral neurons, which was the original objective.

1.3 Bioethics

Plans are to use an interface-implanted rat that will serve as the operator of the robot. The interface-implanted rat would act at its own discretion, which would not cause an ethical problem; however, some brain experiments involving tele-operating, in which rats were controlled by computers, were controversial [8]. The controlling method was direct electrical brain stimulation with implanted electrodes. It could not cause pain for the rat because the stimulating points were in the reward- or pleasure-sensing areas from the right and left whisker bundles of the brain. For these kinds of studies, the NIH guidelines for bioethics (NRC, Guide for the Care and Use of Laboratory Animals, 1996) will be followed. According to the guidelines, it appears that this specific kind of tele-operating would be acceptable; however, the use of the system with vertebrates that were awake might raise some ethical issues. An objective of this research was to use the tele-operated rats as sensor-carrying devices that would search for and find survivors in emergencies or disasters, as the mobility of current robots is not enough for that purpose. Hence, since the applications of the research aim at saving lives, bioethical objections might be fewer. It appears that the best way to avoid negative consequences from the ethical questions of brain control would be to develop robots that perform at a level that is superior to that of animals. The concept proposed in this study, in which an animal mimics a robot, appears to have potential in this sense.

2 Mechanisms

The prototype robot named “Whitegoat 02” is shown in Fig. 2. The basic construction of the linkage that drives the leg follows that used by Yuminamochi in his design for the eight-legged robot called “Tanpopo” [1]; the linkage mechanism is shown in Fig. 2a. The trajectory of the leg is generated using the Chebyshev linkage mechanism. The linkage is a simple mechanism, which generates quasi-uniform motion and arch-like quick return movement.

The main characteristic of the novel mechanism is the addition of a horizontal linkage to swing the legs to the main vertical linkage mechanisms in order to drive the walking action with the main power source. In order to enable the deflection of the leg drive direction, two thrust joints and one twist joint were added to the vertical link (Fig. 3b).

Point G moves constantly, as shown by the arrow, and part F of Fig. 3b constrains the direction of the trajectory of the leg, as shown by arrows H-H'-H'', swung by the servomotor joint E and double-thrust joints I and J. As the revolution angle of a leg becomes large, the motion of the leg-landing phase is shifted from a quasi-uniform motion. With improvements, a smoother walk could be achieved by controlling the revolving speed of the motor driving the leg according to the revolution angle of the leg.

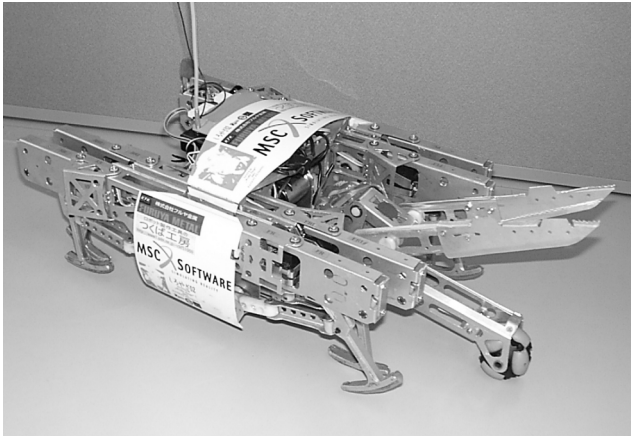


Fig. 2. “Whitegoat 02”

Two linkages (black and gray) are arranged in parallel but with some shifting to allow the legs to swing without conflict. The linkages are driven with a 180-degree phase difference (Fig. 4). For our robot, we have selected a more compact and simple type of counterbrace Chebyshev linkages called “mirror Chebyshev,” which are developed by Ogura and Nakaue [9], rather than the conventional parallel Chebyshev linkages. This robot’s power unit needs to be able to supply driving force independently on the right- and left-side legs. The power unit for the existing crawler vehicles can be diverted to such a use.

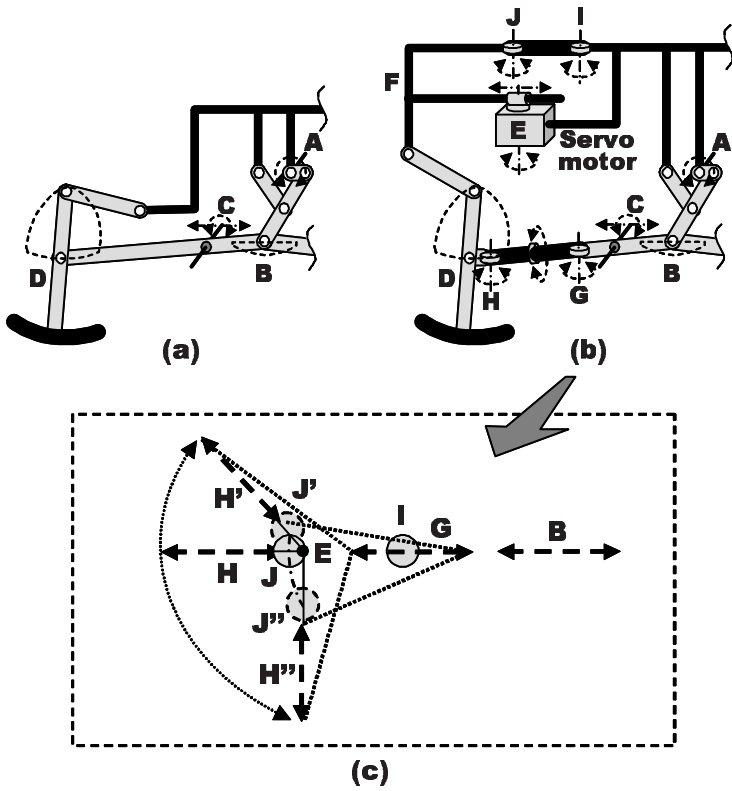
The robot can perform various walks, as shown in Fig. 5, by combining the rotation control of a motor with the revolution of the legs.

The walk control system is composed of two inexpensive one-chip microprocessors (Fig. 6). The system lowers the cost of production, operation, and maintenance, and it enables high durability in severe environments, such as those involving high temperature, high humidity, dust, or dry conditions.

2.1 Design Concepts

In designing the prototype robot, the following points were considered regarding future practical applications.

1. Although the current linkage mechanisms of the robot were designed with circular feet, such as a prosthetic foot that could be used for athletes, a simple bar-style leg was chosen by use with another linkage setting. It would then be easy to repair a broken leg by using a simple spare stick, such as a stilt.
2. The body parts are composed mostly of sheet metal to enable mass-production of the parts with press punching.



- A: Drive shaft**
- B: Chebyshev linkage endpoint and its trajectory (dotted line)**
- C: Linear slide & rotational joint**
- D: Slide inverted Chebyshev linkage endpoint and its trajectory (dotted line)**
- E: Linear slide & rotational joint**
- F: Leg Steering part**
- G~J: Thrust joints**

Fig. 3. Schematic view of the linkage system
(a) Side view of a conventional leg-driving vertical linkage
(b) Side view of the horizontal swing linkage combined
(c) Top projected view of the trajectory of the horizontal swing linkage points

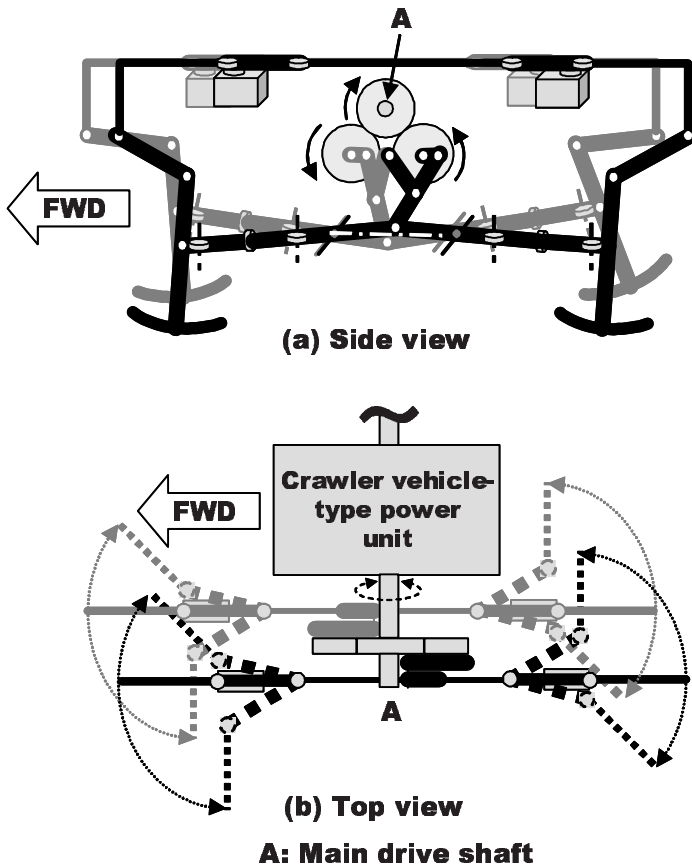


Fig. 4. Schematic view of the parallel linkages(left side)

3. The servomotor that controls the swing direction of the legs hardly contributes to the weight of the robot and does not demand the precise joint-angle control that is required in a conventional walking robot. Hence, the low-output, low-precision, and low-cost servomotor is sufficient for the robot.
4. The leg-driving power uses continuous rotary input through a shaft from the center gearbox to the right and left leg units. Because of this mechanism, the present design of the robot can be extended by installing an internal combustion engine and a gearbox for conventional crawlers, although the current prototype uses a DC motor. It would be easy to add wheels to allow movement on pavement by taking the power provided by the belt drives from the described input shafts from the center gearbox.

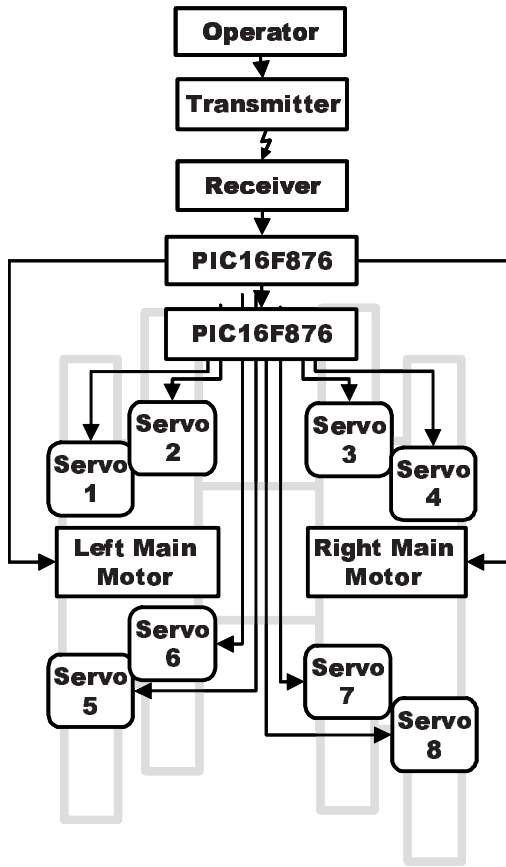


Fig. 5. Control system

2.2 Specifications

The prototype robot has been developed with the following specifications:

4-channel, radio-controlled

Processors: Microchip PIC16F876, x2

Main motors: Johnson 380, x2

Servomotors: Futaba S3102, x8 (3.7kg/cm, 0.25sec/60deg.)

Energy consumption: 40W, Battery: Sanyo 8.4V 1900mAh Ni-Cd

Maximum speed: 80cm/s (2.9km/h)

Length: 51cm Width: 25cm Height: 17cm, antenna excluded, Total Weight: 3500g

Weight capacity: ca. 4kg

The robot can perform various walks, including high-speed sidestepping, by swinging its legs. In March 2002, a patent application was made for the novel swinging-leg system from the Japan Science and Technology Corporation (JST), application number 2002-066321, “Multi-legged walking robot.”

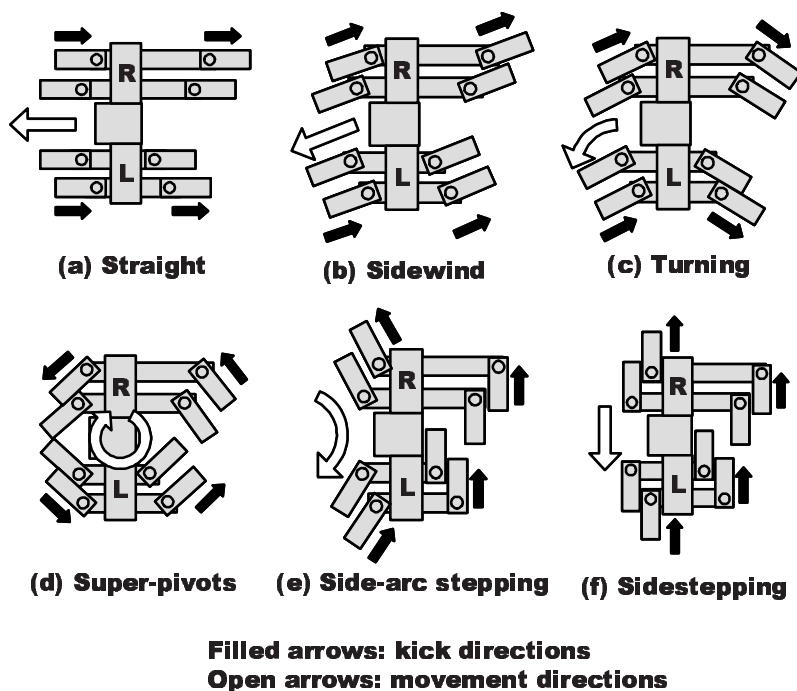


Fig. 6. Walking variations

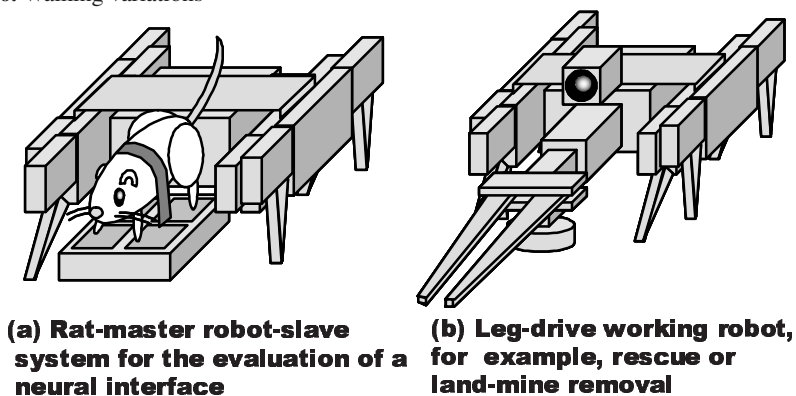


Fig. 7. Applications

3 Conclusion

The prototype robot “Whitegoat” is an eight-legged, radio-controlled walking robot. In the next trial, four legs will be used to make the robot more animal-like so that it can be used in medical engineering, which was the original objective in this study (Fig. 7a). The possible applications for the robot might not be limited to medicine,

however. Plans are also underway to apply it to various other fields, considering its advantages in terms of practicality, mass productivity, and mobility (Fig. 7b).

Acknowledgement

We would like to thank MSC Software, Ltd. for financial and technical support during the development of our "Goat" series of robots.

References

1. M. Yuminamochi, "Tanpopo," *ROBOCON Magazine*, Vol. 7, pp. 10–17, 1999. (in Japanese)
2. M. Shimojo et al., "Development of a System for Experiencing Tactile Sensation from a Robot Hand by Electrically Stimulating Sensory Nerve Fiber," *IEEE Int. Conf. on Robotics & Automation*, Taipei, Taiwan, Vol. CD-ROM, pp. 1264–1270, 2003.
3. J. K. Chapin et al., "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nature Neuroscience*, Vol. 2, pp. 664–670, 1999.
4. J. Wessberg et al., "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, Vol. 408, pp. 361–365, 2000.
5. T. K. Saito et al., "Photodynamic assistance increases the efficiency of the process of microinjection in animal cells," *Biotechnol. lett.*, Vol. 24, pp. 309–314, 2002.
6. T. Saito et al., "Light dose-and time-dependency of photodynamic cell membrane damage," *Photochem. Photobiol.*, Vol. 68, pp. 745–748, 1998.
7. R. Yano et al., "MALC, a novel microinjection method for loading of macromolecules into cultured neurons," *NeuroReport*, Vol. 13, pp. 1263–1266, 2002.
8. S. K. Talwar et al., "Behavioural neuroscience: Rat navigation guided by remote control," *Nature*, Vol. 417, pp. 37–38, 2002.
9. T. Ogura, T. Nakaue, web Page, <http://asahi-net.or.jp/pg9t-ogr/> (as of Oct. 1999)

Online Interactive Building of Presence

Jussi Suomela, Jari Saarinen, Aarne Halme, and Panu Harmo

Helsinki University of Technology, Automation Technology Laboratory,
PL5500, 02015 HUT, Finland
jussi.suomela@hut.fi
<http://www.automation.hut.fi/>

Abstract. This paper presents methods for cooperative mapping of partially or totally unknown area with human and robotic explorers. Mapping is supported with online modelling of presence, which will create a common understanding of the environment for both humans and robots. The studied key methods are human navigation without ready installed beacons, human and robotic SLAM, cooperative localization and cooperative map/model building for common presence. Methods are developed, tested and integrated in a European Community research project called PeLoTe.

1 Introduction

Development of mechatronic machines and robotics has been fast on the level of automation of machine functions, but relative slow in the area of user interfaces. The main difficulty in the interface design is the different level of cognition of humans and robots. In order to work and communicate well together robots and humans should understand the same language, its contents and abstractions and the common environment similarly. The processes of human cognition – especially the understanding level – are studied a lot but they are still practically unknown. On the other hand we know very well what our cognition can do. Both functions and capability of robot/computer cognition are well known because we have created them. As a result we can be sure that both the function and performance of human and robot cognition are different. One of the key issues is the environment. When robots and humans are working in same area they should be able to understand the common environment (“presence”) and change information relative to it. This is especially important in mapping type of tasks where both robots and humans generate information of the environment and both should be capable to exploit the information generated by the other.

1.1 PeLoTe Scenario

Pelote (Building Presence through Localization for Hybrid Telematic Systems) is part of the IST programme of the European Community. The target of the project is to study how to map a totally or partially unknown area with group of human and robotic entities and form a common environment model (presence) from the mapped data produced by both entities. The model will update in real-

time and provide presence for both humans and robots. This type of scenario is typical in rescue, military and planetary exploration tasks.

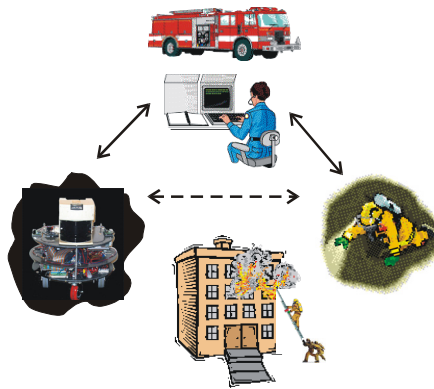


Fig. 1. Pelote scenario: Human and robotic entities are exploring and mapping a common area.

The case example in Pelote is a rescue task where firemen and supporting robots are mapping a common area together with help of a remote operator Fig. 1. Both entities specialize to tasks, which are natural for them. Robots can perform accurate navigation and measurements from the environment even in hostile conditions. Humans can give fast verbal descriptions of the situation and conditions. Human senses are also more versatile than robot senses. Exploring entities are supported by a remote human operator in a mobile control room. He teleoperates robots and supervises the firemen. He also summarizes the information obtained from both entities to the common environment model.

2 Human Entities (HE)

2.1 Personal Navigation (PeNa)

In a rescue situation it's essential to know the position of all human entities. Unlike robots, the humans only rarely know their accurate position. Only in situations where human can identify a known object, like corridor crossing or stairway, he can know his accurate position relative to that object. Personal navigation (PeNa) under development is intended for continuous automatic localization of humans. PeNa is based on human dead reckoning supported with radio beacons (if available), and the human's own senses and perception. Also robots can support PeNa. Dead reckoning includes step measurements (pedometer), magnetic sensors (compass) and inertial measurements (gyro and accelerometers). Possible existing radio beacon systems are WLAN, GPS and pseudolite GPS, which can also be built up for the rescue mission. Robots have relatively better position than humans. To help human positioning robots can either carry or drop wireless (radio or/and US) beacons, which humans can use. Additionally, the feasibility of human SLAM based on laser scanner is studied.

In addition to PeNa hardware, human entities are equipped with data/speech communication links and a HMD, which provides real-time mapping information with own position based on Pena. This whole system is called Personal Assistance System (PAS) illustrated in fig. 2.

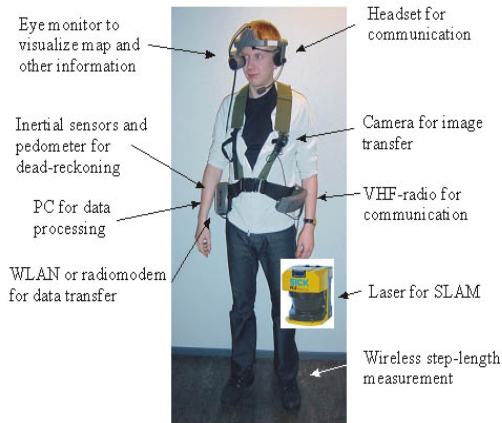


Fig. 2. Human hardware, the “personal assistance system” including PeNa

2.1.1 Human dead-reckoning

Rescue tasks are mainly done indoors where existing radio beacon systems like GPS, cellular or WLAN transmitters are not always available. It might be possible to set up local beacon system around the building but in practise the availability inside would be more or less zero. Unlike robots humans don't have wheels or tracks with clear kinematics and good odometry. The only possibility is to use inertial navigation combined with pedometer and robot based beacon system. The core of personal navigation is DRM-III module (developed by PointResearch Corp.) supported with Crossbow's inertial measurement unit (IMU).

DRM module is low-cost and small sized. The accuracy of the module is promised to be from 2% to 5% of distance travelled. DRM module can be used with GPS, where GPS data is available, which makes the outdoor localization accurate (within GPS error). Moreover the module offers capability to reset the position, when accurate position information is available. The DRM measures the displacement of the user from the initialization by measuring the direction and the distance travelled with each footstep. Direction of travel is measured using an electronic compass. Distance is measured by tracking each footstep using accelerometer data. The preliminary tests showed that the module relies strongly on initialised stride value.



Fig.3. Human dead reckoning with DRM and IMU. Walked distance is about 200m and the real route was enclosed.

The module estimates the stride taken by human to be close to this initial value; even the length of step would be something else. In spite of manufacturer's promises, the module failed also when walking backwards or sideways. Due to these facts it is impossible to say the accuracy of the measurement the given by DRM-III. It can be close to correct, but it can be also totally wrong. Under normal walking and running conditions the average error is within 10% from distance travelled. Preliminary test data is illustrated in fig. 3.

In order to improve the odometry a continuous real time stride length measuring unit (SiLMU) is under work. Stride length will simply be measured between sensors in ankles. Measurement is based on ToF of ultra sound. The hardware is the same, which is used in cooperative localization system presented in following chapter. Only software is modified and the radio connection replaced with wire. Measurement is feasible during walking, running and even crawling. Accurate stride length estimate will improve the odometry significantly. It's supposed that with DRM and SiLMU the accuracy of human odometry will be less than 5% in normal conditions (walking or running forward or backward, no side steps).

Even when the odometry is relatively good the navigation is mostly depending on the accuracy of heading information. The magnetic heading sensor in DRM has unpredictable performance in indoor conditions. To improve the angular accuracy an IMU is included in the system. IMU is equipped with three vibration gyroscopes and three silicon-chip accelerometers. IMU is used to correct the heading and direction errors of DRM. Data fusion is done with EKF. IMU is also used to get pose of laser scanner in human-SLAM tests.

2.1.2 Cooperative localization

Despite of how well the human dead reckoning will succeed, there will always be accumulative error, which can grow very rapidly in special situations like during creeping or climbing over an obstacle. The only way to correct this accumulative error is to use beacons to correct the error always when possible. In the case of an apriori map of the environment, human can use visual beacons to correct his position. HE sees his position on the map from the HMD. If the error is noticeable he can identify a known object, like corridor crossing or stairway, and simply describe his real position and ask the operator to correct the position. However, in most cases humans don't have time to do this or the map is not available. When external beacon systems like GPS, cellular or WLAN base stations are not available indoors, a mobile beacon system based on robots will be developed.

Cooperative localization is based on the fact that robots will know their position much better than humans. Robots have more accurate dead reckoning compared to humans and they will also perform continuous SLAM, which produces excellent position information especially in the case of an apriori map. Robots will be equipped with radio/US beacons, which can either be onboard all the time or dropped to important places like corridor crossings.

To ensure the robust function and keep the beacon price reasonable, the system is simplified as much as possible. Human positioning is based on continuous tracking of available beacons. Tracking is done by radio signal by calling a defined beacon based either on position and model information or simply going them all through. When a beacon receives a call from observer unit with its ID, it replies its position by radio and sends a distance signal by ultrasound. After receiving these signals the positioning system knows the position of the called beacon and the distance to it. The distance information of beacons is added to position filter.

2.1.3 Human localisation and mapping

The objective in mapping is that robots perform the accurate measuring i.e. SLAM based on laser scanner and other possible perception sensors and humans provide verbal information to the operator, who then processes the verbal information into the model. However, the human SLAM based on laser scanner would both improve the human positioning and provide more measuring information from environment without increasing the use of human actions or brain load. In some conditions the lower location (compared to eyes) and longer wavelength of the laser scanner will also provide better visibility than with eyes only.

Basically the human SLAM is the same as robot one. The main difference is the lack of accurate odometry between scans and continuous swinging of the scanner. Especially the floor and ceiling echoes, resulting from large tilting of the sensor, are difficult for algorithms. In this project a standard Sick-scanner is

used even it's clumsy and therefore difficult to place in human body without disturbing the movements. The new 1,5 μm laser technology will anyway shrink the equipment in the future. Already today, small-scaled scanners are available for military applications.

The feasibility of human-slam has been tested with laser scanner supported with part of the PeNa hardware, which provides a transition estimate between scans. The scanner is located in the belt of a human, who walks normally through corridor. The matching algorithm uses raw data (after outlier rejection) and position estimate (integrated from IMU) as inputs. At the moment only the heading from gyro is used. The scans are matched by using particle type algorithm, where the particles, whose positions are tracked, are created on the basis of initial estimate.

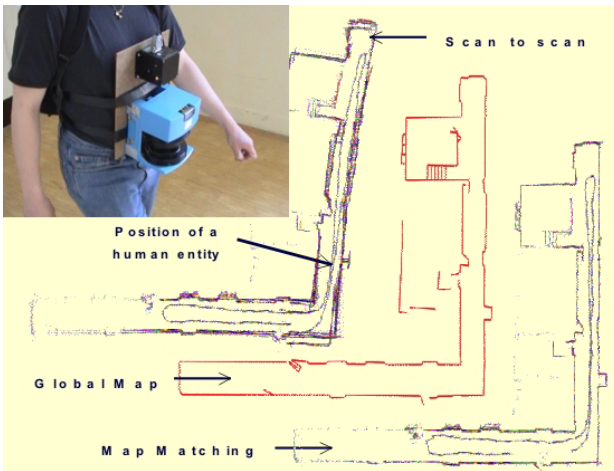


Fig. 4. Laser scanner and IMU, scan-to-scan matching and map matching.

At the moment the particles are evenly distributed and the number of particles for heading estimation depends on the value that gyro says is between scans. At the moment the algorithm also gives only the best estimate of position, but in the future it is intended to use probabilistic approach using EKF/Monte Carlo type algorithm [1], [2].

The algorithm calculates the distances to nearest neighbours according to one particle position. The neighbour is accepted if it is closer than threshold value. The winner is the particle that has the most neighbours. The algorithm in present uses 0.4 deg steps in heading and 5 cm steps in x and y direction. So far the algorithm runs in 0.3 – 0.8s / scan. The first test results are illustrated in fig. 4. The same data is also matched to a map. The algorithm is based on the optimization of point distances to lines (the global map has geometric presentation). The result is very good. However, it has to be stressed that the problem of this kind of algorithm is that, if the algorithm fails, it cannot recover.

2.2 HE – Operator Interface

Position data from Pena and mapping data from operator are transferred automatically via wireless interface. The actual communication between human entities and operator is mainly done by speech. It's assumed that especially in rescue operations men don't have time to play with computer interfaces but they can speak simultaneously they work. For example when a fireman notices that a remarkable error has accumulated in his position on the HMD, instead of correcting his position by using a trackball, he asks the operator: "I am front of the elevator facing to main door, please correct my position". Now the operator makes the correction and fireman can check the updated position in real time from his display. This is of course possible only when an a-priori map is available. When robots are using laser scanners and other sensors for accurate numerical mapping, humans use their ability to piece together wholeness. Again fireman acts as intelligent sensor and asks the operator to update the map. For example "the corridor in front of me is collapsed" and the operator marks the corridor as non-enter area in the map.

3 Robot Entities (RE)

The robots are used for accurate mapping and perception of the unknown/destructed areas. They will also explore dangerous areas. Robots are teleoperated by the operator, who points the zones to be explored or even takes a robot under direct control. Robots do simultaneous localization and mapping (SLAM) based on odometry, laser scanner and other possible sensors. External beacons like WLAN or GSM base stations can be used if they are available. Robots can also utilize the cooperative localization beacons if needed. The pre-processed perception data is transferred continuously to the operator and the updated map data is received from the operator.

3.1 Navigation and Map Building/Correction

The robots are sharing the knowledge of environment with humans and therefore the maps made by robots should be readable by humans. Speciality is that human is participating into the mapping process and also robots should understand the human's observations. The human observations are added to the common map shared among entities. The human observations are most likely topologically correct, but since human cannot measure absolute distances, it is likely that the observed objects are in wrong positions. Related and very challenging problem is that in rescue scenario the map of environment is dynamic and the a-priori map is most probably incorrect at least from some parts. The challenging issue is how to use the existing map and human observations although those are not completely correct.

The emphasis of the project is to study first whether it is possible to share this level information between robots and humans. Also dealing with

incomplete/inaccurate a-priori maps is studied. The earlier works in multi entity mapping such as [1] and [2] usually deals with mapping starting from unknown environment by using robots for environment perception. In [1] the observations and world model is based on detecting natural landmarks by using radar or ladar. This kind of world model is however not possible, when humans should use the same map as robots. The landmarks for human can be very different from those of robots. In [2] the robots are building an occupancy grid type of map in 2D (and even in 3D), which is more suitable for human perception.

The kind of maps that are readable for humans and for robots are e.g. occupancy grid maps and geometric maps. The Occupancy Grid maps (OCM) provide information based of the occupancy probability of some small area [3]. The geometric map is a map that presents the environment with polygons. The problem of occupancy grid map is the exchange of information between entities. The size of OCM is growing very fast when the environment is complex or/and large. The geometric map is in general smaller and easier for human to understand, but dealing with inaccuracies is more difficult.

The needed objects for localization in the different layers of map are fused to form the base localization map. This map is divided between different entities. The entities make from global map local occupancy grid sub maps according to their position. This occupancy grid map is based on probabilities in geometric maps. The robots use local occupancy grid maps for localization and map correction. In centralized solution, the robots are sending the local OC maps to command centre with some intervals. In command centre these grid maps are transformed back to geometric maps. The method for transforming OC maps to geometric maps is presented in [3].

4 Common Presence

The main topic of the project is to generate a presence from the data perceived by human and robotic entities. The presence should be common for both entities. The common presence consists of 2D or 3D map or model and an object database, which can be updated with any kind of object information. The model is composed from possible apriori map.

The model shows the navigable area and borders like walls, constant objects etc. All additional information and mapped objects are stored in the database, where they can be picked up and augmented in the model. The continuous mapping data from robotic and human explorers is added to the database.

The operator can afterwards include or correct “constant wall – type” objects as part of the model. In the database all objects have at least size, position, type, additional information and layer group. Additional information can be anything, even a photo or verbal description, which can be seen/hear by clicking the object. The layer group information is used to visualise only the needed information in order to avoid the information flood on the model.

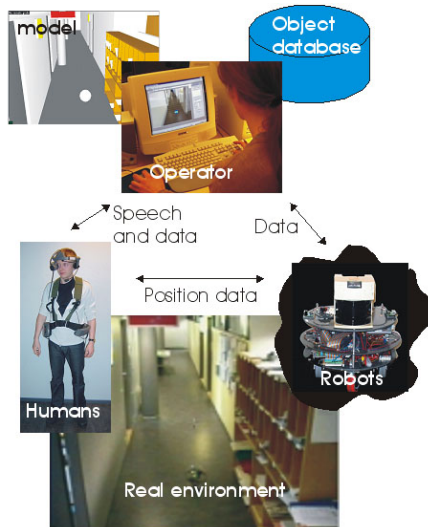


Fig. 5. Forming the common presence

As described before, robots do continuous SLAM in the environment. All obstacles differing from the model and database are mapped as polygons and transferred to the database. After operator has seen the data he will decide whether the obstacles are just objects or are they classified as constant structures and transferred to the model. To robots the model and the “robot visible” objects are shown only as a 2DoF occupancy grid allowing the localization based on laser/camera odometry.

Humans send both verbal and data information from the environment. The map data is handled like in the robot case but the verbal and possible photo information is always “translated” and filtered by the operator. To humans the model and needed layers of database are shown as full augmented model including his position (Fig 5). If the HE is not requesting any defined layers, they are shown simply as function of predefined importance and zoom in order to keep the model informative. The base of the modeling software is presented in [4].

5 Conclusions and Discussion

While this paper is written the Pelote project is still in start phase. Due to this fact the main part of the subtasks are only defined here and the work is still to be done. However, part of the defined methods like human dead reckoning, human SLAM and building augmented reality has been already tested and

found feasible for described cooperative mapping and presence building task. Especially laser scanner based human SLAM will be studied further as well as human dead reckoning will be improved with stride measurements and better performance gyro.

System integration, testing and final demonstrating will be carried out before the November 2004.

References

1. Stefan B. Williams, Gamini Dissanayake, Hugh Durrant-Whyte. Towards Multi-Vehicle Simultaneous Localisation and mapping. Australian Centre for Field Robotics, University of Sidney. *International conference on robotics and automation 2002*
2. Thrun, S. and Burgard, W. and Fox, D. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2000
3. Kulich, M. - Štěpán, P. - Přeučil, L. Knowledge Acquisition for Mobile Robot Environment Mapping. In: *Database and Expert Systems Applications*. Berlin: Springer, 1999, vol. 1, p123-134. ISBN 3-540-66448-3.
4. Harjo P., Halme A., Pitkänen H., Virekoski P., Halinen M., Suomela J., Moving Eye – Interactive Telepresence Over Internet With A Ball Shaped mobile Robot, *Proceedings of IFAC Conference on Telematics Applications in Automation and Robotics*, TA2001, 24-26 July 2001, Weingarten, Germany

Light Weight Autonomous Climbing Robot for Elderly and Disabled Persons' Services

Carlos Balaguer, Antonio Giménez, Alberto Jardón, Raúl Correal, Ramiro Cabas, and Pavel Staroverov

Robotics Lab
Department of System Engineering and Automation
University Carlos III of Madrid
balaguer@ing.uc3m.es
<http://www.uc3m.es/robotics>

Abstract. Human care and service demands an innovative robotic solution to make easier the everyday of elderly and disabled people in home and workplace environments. The EU project MATS has been developing a new concept of climbing robot for this type of service applications. The robot developed by University Carlos III of Madrid is a 5 DOF self-containing manipulator, that includes on-board all the control system. The main advantage of the robot is its light weight, about 11 kg for a 1.3 m reach. The robot is a symmetrical arm able to move between different points (Docking Stations) of the rooms and, if it is necessary, "jump" to (or from) the environment to the wheelchair. In this way the MATS robot should become a home companion and assistance for numerous persons.

1 Introduction

The role of the manufacturing industrial sector in the economy of industrialized countries is decreasing every year. The European Union agency Eurostat certificates that the role of services increases (52% in 1970 and 71% in 2001) and the role of manufacturing dramatically decreases (30% in 1970 and 18% in 2001). At the same time the ratio of elderly people is constantly increasing with the estimation to be 20-28% of the whole population in 2030. The elderly persons have an important reduction of their physical functions with aging. And unfortunately many elderly people get severe diseases with some troubles in physical functions [1]. This is why there is an urgent necessity to develop the new technology to support them in all their living and working environment.

During the last 15 years the rehabilitation technology has been developing towards more flexible and adaptable robotic systems. These robots try to assist persons, to support disabled and elderly people with special needs in their homes. The nowadays' rehabilitation robotics technology is focusing in three main development concepts:

- a** static systems that operate in a structured environment
- b** wheelchair mounted robotic systems for personal and care applications
- c** mobile manipulator companion following the person



(a) Handy 1



(b) MANUS

Fig. 1. Commercial assistive robots

First type of robotic systems are very useful when the persons need help in the same living environment and for the same application, such as eating, drinking, washing, shaving, etc. These robots have very good mechanical stability and adaptability for environment. The Handy 1 robot arm [2] is an excellent example of the static robot system, see figure 1(a). It is a low cost solution for personal care and assistance. Over 200 of these robots have been placed on the market to date. Nevertheless, the static robotic systems have one important limitation:

- a** the change of its location is very difficult (for example, to shave in the bathroom would be difficult if the robot must be carried upstairs each time and then back downstairs for eating)
- b** occupy valuable floor space
- c** limited manoeuvrability and dexterity due the robot's static base positioning

Other type of rehabilitation robots are wheelchair mounted. The current market leader of this type of robots is the MANUS system [3]. It is estimated that more than 60 of these systems have been place to date. This robot is used by the disabled to maintain independence in some structured environments in some tasks such as pouring a drink, drinking, meal preparation and washing (Figure 1(b)). Nevertheless, its stability in the wheelchair should be careful studied and controlled. Moreover, the arm is permanently fixed on either the left-or right-hand side of the wheelchair. Finally, the cost of the systems is very expensive.

The third concept in the rehabilitation robotics is the mobile manipulator following the wheelchair of the persons in structured environment. This concept has the similar advantages and disadvantages that the previous one. But it introduces new advantage; the robot has the ability to move independently from the wheelchair or the person. Example of this system is the KARES II robot [4]. Its main tasks are serving a meal and beverage, picking up an object from the floor, shaving, wiping a face with a wet towel, turning on or off a switch, etc. (Figure 2).



(a) Simulation



(b) Realtests

Fig. 2. KARES II robot


(a) Kitchen



(b) Bathroom

Fig. 3. MATS robot working environments (animations done by University of Staffordshire and University of Lund)

The European Union MATS project, with the participation, among others, of the University of Staffordshire (UK), Scuola Superiore Sant'Ana in Pisa (Italy), University of Lund (Sweden) and University Carlos III of Madrid (Spain), has the objective to develop the robotized system that joint both, the static and moving system into one climbing robot. The robot is able to be attached to the wheelchair and helps the disable person in his/her life domestic tasks. But at the same time the robot is able to "jump" from/to the wheelchair to/from the domestic environment and vice versa. In this way the MATS robot extends the human abilities and is able to perform a big variety of domestic operations: housekeeping, assistance, entertainment, etc. Figure 3 shows several working robot environments where two MATS robot cooperate for perform several domestic tasks.

This paper presents the new concept in the rehabilitation robotics [5]. The main advantage of the MATS robot concept is the light weight manipulator, 11 kg for 1.3 m reach, which includes on board all the control system. The communication

with the robot is performed wireless through friendly human-machine interface. The robot can be attached to the environment (wheelchair) by specially design low cost Docking Stations (DS). The locations of the DSs permit the robot to move from one location of the environment to another, and sometimes from one room to another.

2 Kinematics Design

The analysis of the MATS robot applications permitted to establish the length of the robot. Several typical applications in different environments were selected for this purpose: a) kitchen environment and manipulating plates, glasses, bottles, knives, etc., b) dinning room and eating, drinking, etc, c) bath room and tooth cleaning, washing, shaving, etc. d) bed room and dressing, manipulation of books, glasses, etc. Figure 4 shows the MATS robot in the kitchen environment during the eating application. This analysis leads to the 1.3 m length robot.

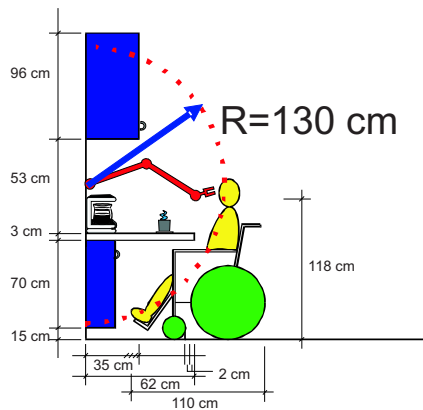


Fig. 4. MATS robot in the kitchen environment during the eating assistive application

From the kinematics' point of view the robot's structure needs to be symmetrical. The robot must work in the same manner when it has attached one tip or the other one. It means that the robot is a climbing device and needs to assume all the features of these systems [6]. For this type of robot one axis symmetry in the middle is necessary. At the same time the robot needs to be designed with the minimum necessary number of DOF in order to reduce the overall weight. This number of DOF should guarantee the sufficient mobility of the robot in the room environment, i.e. transitions from one DS to another (commonly vertical and perpendicular ones respect to another) and from the wall DS to wheelchair DS.

This analysis concludes with the selection of the 5 DOF configuration with the overall length of the robot equal to 1.3 m, including docking mechanisms, grippers and fingers. This number of DOF and length are enough to perform most

of the previously analyzed applications [7]. Figure 5 presents the robot's kinematics structure where C is the DS attachment device, A is the end of the robot, and B is the most critical robot axis.

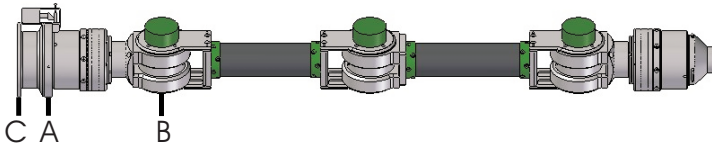


Fig. 5. MATS robot kinematics

3 Structural and Actuators Optimization

One of the key issues of the MATS robot design is the optimization of the overall robot structure. Two different approaches were performed. On the one hand, the robots structure must be stiff in order to support its weight and the dynamical reactions during the movements. The rigidity of the arm is very important in order to perform successfully the transition from one DS to another. It is well known that structural rigidity means high weight.

On the other hand, the lighter robot structure leads to the lighter actuators, which leads to lighter robot structure again and to the lighter actuators, etc. This iterative process has a minimum that represents the optimal structure weight [8]. Taking in mind that the weight of the actuator (motors and gearboxes) represents more than 30% of the overall weight of the robot, it is easy to understand their importance.

The finite elements' analysis (static and dynamic) together with the actuators selection will lead to the optimal robot design. For this purpose the whole robot stiffness and the force and torque analysis of the each of the robot's pieces were computed. Figure 6 shows the structural analysis of the arm in different critical positions and figure 7 shows the design of each robots piece in order to minimize the weight and at the same time satisfy the structural stiffness. The second approach is presented in figure 8 where the torque and temperature selection of the actuators are shown.

To minimize the overall weight of the robot the new technologies of the actuators, gearboxes and materials were used. The actuators are formed by motors, gearboxes and sometimes the transmission mechanisms. The conventional gearboxes, like planetary and others, are very heavy and have a big backlash. The most appropriate gearbox is the well known Harmonic Drive device. It has several advantages: light weight, high torque and speed, good torque transmission coefficient (efficiency), very small or zero backlash, etc. Nevertheless, the occupied space sometimes is very big. This is why the MATS robot uses the ultra flat Harmonic Drive gearbox. It reduces more that 50% of the length of the device for the same torque. At the same time the weight has some reduction also.

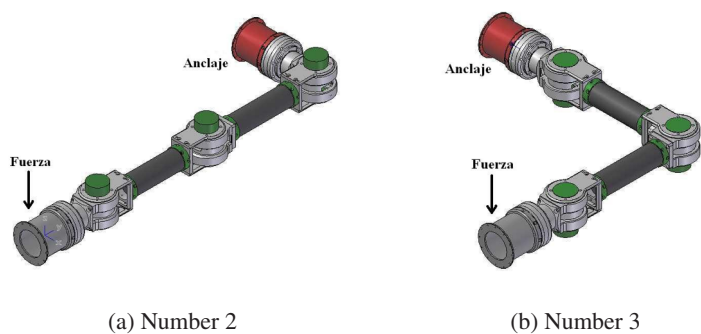


Fig. 6. Structural analysis of the robot in different critical positions for axis

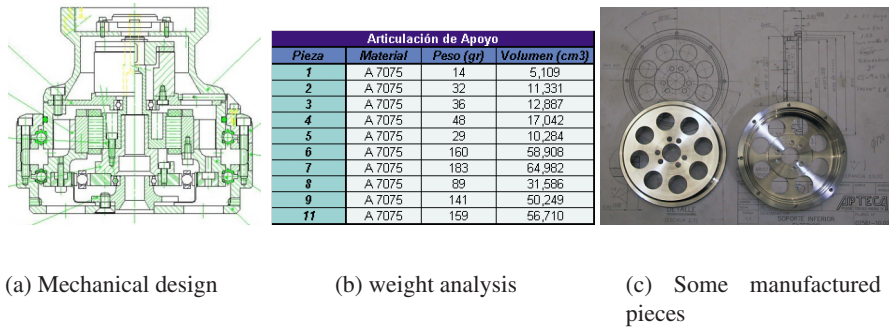


Fig. 7. Optimal structural/weight design of the robots pieces for the axis

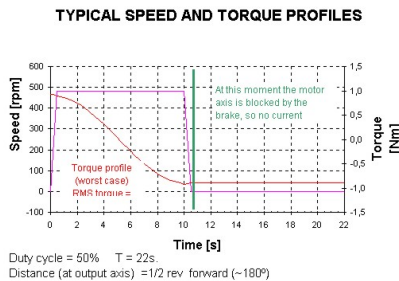


Fig. 8. Optimum slection of the electrical motor for the axis 1

Another important component is the motor. There are different well known technologies in the market like the DC brush or brushless motors with reduced weight and volume. More advanced technology is the torque motor (Figure 9(a)). Its main advantage is a constant high torque for a big range of velocities, including small ones. It has many other advantages: significant reduction of the length of the motor (more that twice), significant reduction of the weight (about twice), and possibility to custom manufacturing of the light weight hollow rotor axis. All these advantages

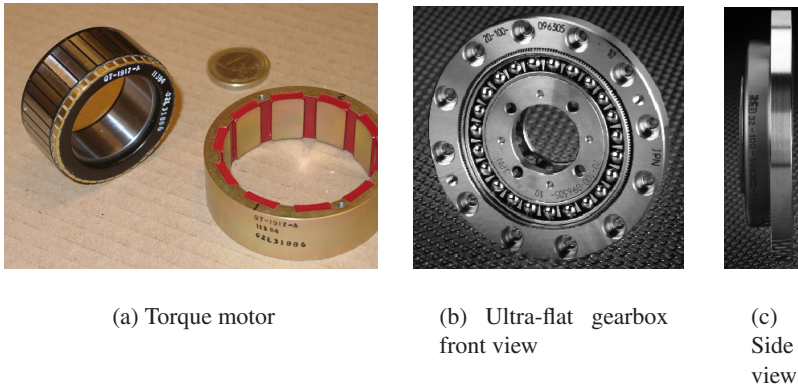


Fig. 9. Elements of MATS robot

result in that the overall weight of each axis has more than three times reduction. The same is for the length of the set gearbox-motor-break-encoder (Figures 9(b) and 9(c)). The critical stage of this set is the assembly process which can be performed manually in factory. For good assembly special assembly tools were developed.

The selection of appropriate materials is also part of the light weight robot design. The MATS robot uses the 7075 aluminium alloy and the custom produced carbon fiber tubes to serve as links between the joints. The diameter and the thickness of the tube are selected taking into account two issues: a) mechanical stiffness of the links and b) the necessary interior space to allocate the hardware control devices inside the tube.

The result of this careful design and actuators' selection is the MATS robot with the weight of about 11 kg, including the on-board control hardware. Figure 10 shows the robot easily supported by one person. In this way, the robot can be transported from one environment to another by assistive personnel. The same figure shows that the axis limits are very big which results in the high robot's maneuverability

4 Control Architecture

The on-board control system is formed by three different parts: a) CPU, b) actuators and c) communications. The innovation is centered in the used micro-design embedded elements. The robot's CPU is a StrongARM based board with the dimensions of 57 x 69 mm, which permits their easy installation inside the carbon fiber tubes of the arm. The same is for the motor drivers which are 65 x 58 in size and 10 g in weight, also installed inside the robots tube. The communication of the robot CPU with the "ground" central computer is performing wirelessly by IEEE 802-11b protocol. In this way the only necessary connection of the robot is the power supply (24 V).

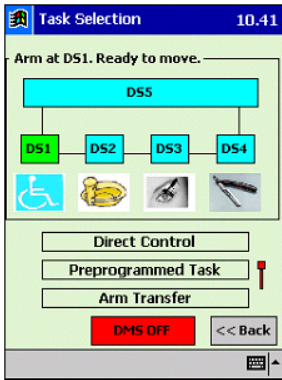


Fig. 10. MATS robot can easily supported by one person

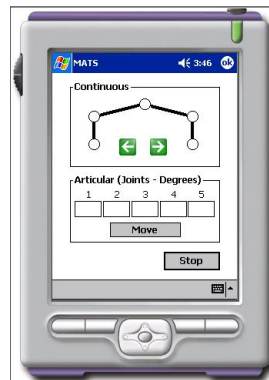
The "ground" computer is based on the PDA technology. Its main objective is to define the robot environment and act as the HMI between the user and the robot. Figure 11(a) shows the PDA screen where several DS are defined together with the some robots control commands. This HMI receives the different maps of the rooms of the house from the room controller which contains the whole information about the home environment, [9]. The screen 11(b) is for the direct robot control where both manual and automatic tasks can be performed.

In the "ground" computer different task levels are implemented: a) high level, b) learning, c) teleoperated, and d) direct control. In the first level, different automatic programs were generated previously. They are, for example, moving from one DS to another, opening the kitchen's door, etc. The learning level permits to generate a high level program or learns one specific position. In this way it is possible to learn the robot to perform specific tasks. During teleoperation of the robot the human user can use different interfaces manually (with a joystick), by voice, by simple On/Off switches, etc. Finally, direct control permits the debugging of the motions and creation of the numerical programs.

To transfer the MATS robot from the environment to the wheelchair, the so called "bird's-eye view" system had been implemented. It includes sensor integration (infrared system) in the DS to perform the docking operation manually with success and in a short time. This system is also includes a web-camera in the arm. The infrared and video data are transmitted to the "ground" PDA based computer where specially developed docking process guidance software was implemented. The docking operation is crucial for the good functioning of the MATS robot. Figure 12 shows the robot in the experimental test environment. Figure 12(a) shows the environment where the robot is attached to the wheelchair and carry out at the tip the spoon toolholder for eating tasks. Figure 12(b) shows the robot attached to the specially developed disable service table with several facilities for eating, make-up, washing, etc.



(a) PDA home environment interface (done by Scuola Superiore Santa' Ana in Pisa)

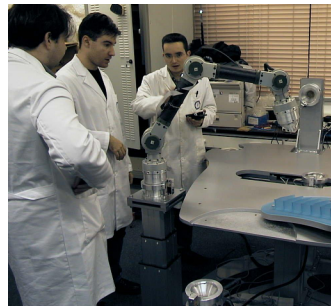


(b) Direct robot control

Fig. 11. MATS robot interfaces



(a) Robot attached to the wheelchair with the spoon tool-holder



(b) Robot attached to the disabled people service table

Fig. 12. MATS robot in work

5 Conclusions

The MATS robot, developed by University Carlos III of Madrid, presents excellent ratio weight/number of DOF/length. Moreover, given that all the control system is on-board, the robot can be very easily transported from one environment to another. In this way it can be used by several disabled and elderly people, reducing the cost of exploitation. The efficiency of the robot in the daily tasks is very high together with the extremely easy adaptation of the home environment to the MATS robot, by introducing low cost DSs and a unique power supply.

The robot is under testing actually. The adjustment of the control parameters that permit to work successfully in every situation is the main objective. The docking process must be robust for any position of the DS and the robot. The actual tests demonstrate that with the tolerances of the DS location in order of some mm and some degrees, it is possible to perform the docking process in an automatic way with any type of compliance.

Acknowledgement

The authors gratefully acknowledge the important contribution of Santiago Martinez, Raul García, José Antonio Campo, Angela Nombela and Carlos Perez. The MATS project is being funded by the European Union under the project n° 2001-32080 of the IST (Information Society Technology) program.

References

1. M. G. Fujie, "How to commercialize service robots in the 21st century", *Proceeding of the 1st IARP/IEEE Workshop on Technical Challenge for Dependable Robots in Human Environment*, Seoul, 2001.
2. M. J. Topping, "Handy 1, a robotic aid to independence for severely disabled people", in Mokhtari, M. (ed.), *Integration of Assistive Technology in the Information Age*, IOS Press, 2001.
3. H. Kwee, "Integrated control of MANUS and wheelchair", *Proceedings of the International Conference on Rehabilitation Robotics (ICORR'97)*, Bath (UK), 1997.
4. W.-K. Song, H.-Y. Lee, J.-S. Kim, Y.-S. Yoon, and Z. Bien, "Kares: intelligent rehabilitation robotic system for the disabled and the elderly", *IEEE International Conference on Engineering in Medicine and Biology Society*, 1998.
5. M. J. Tooping, "Flexibot - a multi-functional, general purpose service robot", *Industrial Robot*, vol. 28, n° 5, 2001.
6. C. Balaguer, A. Gimenez, M. Abderrahim, "A climbing autonomous robot for inspection applications in 3D complex environment", *Robotica*, vol. 18, September 2000.
7. C. Balaguer, A. Giménez, A. Jardón, "MATS: An assistive robotic climbing system for personal care & service applications", *1st Workshop on Advanced in Service Robotics (ASER'03)*, Bardolino (Italy), 2003.
8. A. Gimenez, "Metodología de diseño y control de robots escaladores. Aplicación a las tareas de inspección", (Methodology of design and control of climbing robots. Application to inspection tasks), Ph.D. Thesis, University Carlos III of Madrid, 2000.
9. A. Gimenez, C. Balaguer, A. Sabatini, V. Genovese, "The MATS system to assist disabled people in their home environments", *IEEE/RSJ International Conference on Intelligent Robots and Systems 2003 (IROS'03)*, Las Vegas (USA), 2003

Planning under Uncertainty for Reliable Health Care Robotics

Nicholas Roy¹, Geoffrey Gordon², and Sebastian Thrun³

¹ School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
nickr@ri.cmu.edu

² ggordon@cs.cmu.edu

³ thrun@cs.cmu.edu

Abstract. We describe a mobile robot system, designed to assist residents of an retirement facility. This system is being developed to respond to an aging population and a predicted shortage of nursing professionals. In this paper, we discuss the task of finding and escorting people from place to place in the facility, a task containing uncertainty throughout the problem.

Planning algorithms that model uncertainty well such as Partially Observable Markov Decision Processes (POMDPs) do not scale tractably to real world problems such as the health care domain. We demonstrate an algorithm for representing real world POMDP problems compactly, which allows us to find good policies in reasonable amounts of time. We show that our algorithm is able to find moving people in close to optimal time, where the optimal policy starts with knowledge of the person's location.

1 Introduction

We describe a mobile robot system, designed to assist residents of an retirement facility. This system is being developed to respond to an aging population and a predicted shortage of nursing professionals. Previously, we have reported on work focused on the task of reminding people of events (e.g., appointments) and accompanying them to these events [1], [2]. In this paper, we discuss the task of finding and escorting people from place to place in the facility. The problem of finding people is a challenging one because the state of the world is not completely known (the initial position of the person in the environment is unknown), the state changes (people are free to move around), and sensor noise can lead to perceptual errors. These substantial sources of uncertainty can lead to sub-optimal behaviour on the part of the robot.

Unfortunately, the kind of planning that is required for reliable robot operation is difficult to approximate with simple heuristics for handling the uncertainty, so we must use a planning methodology that explicitly models the real-world uncertainty. One such model is the Partially Observable Markov Decision Process (POMDP), but conventional approaches to finding policies for POMDPs are often intractable for the size of problems we wish to address.

We will take advantage of dimensionality reduction techniques to find low-dimensional representations that can be planned for much more easily, by using structure inherent in many real world domains. For example, Principal Components

Analysis (PCA) is well-suited to dimensionality reduction for data on or near a linear manifold in the higher-dimensional space. Unfortunately, POMDP belief manifolds are rarely linear; in particular, sparse beliefs are usually very non-linear. We therefore transform the data into a space where it does lie near a linear manifold; the algorithm which does so (while also correctly handling the transformed residual errors) is called Exponential Family PCA (E-PCA) [3,4]. E-PCA will allow us to represent POMDPs with only a handful of dimensions, even for belief spaces with thousands of dimensions. We will demonstrate the use of this planning technique on the problem of how to find a person whose location is initially unknown.

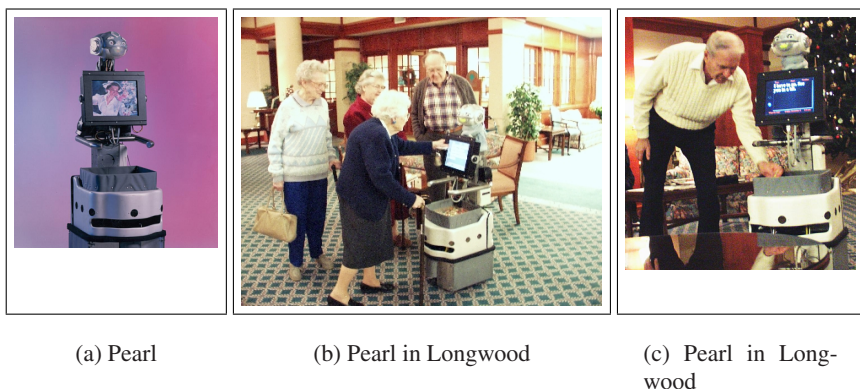


Fig. 1. (a) Pearl, the Nursebot (b) & (c) Pearl interacting with residents of Longwood at Oakmont.

2 Finding People

The problem we wish to solve is how to find people in a health care facility as quickly as possible. The robot is assumed to begin with a grid map of the environment, but no knowledge of where the person might be located, in which grid cell of the map. The robot can move about the environment to look for the person, and receives sensor information when the robot can and cannot see the person. Our implementation is based on a laser range-finder, but this work is independent of the particular sensing modality.

We assume a probabilistic state estimator that provides probability distributions over where people might be located. We will refer to this a distributions as a “belief” of the person’s location. The belief is updated over time after each action and observation from the robot according to a well-formed probabilistic rules [5]. The planning task can then be phrased as one of choosing the next action, based on the current belief, as depicted in figure 2. Not shown in this figure is the true state of the world, which is also not observable by the agent.

The Partially Observable Markov Decision Process is a decision-theoretic model for planning successfully with beliefs. The POMDP is solved by defining a “value

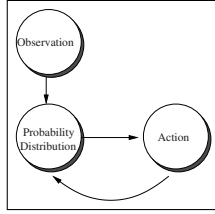


Fig. 2. The execution process for finding people. The observation is generated according to an emission probability model conditioned on the current state, that is hidden from the controller. The controller *only* has access to the observation, not the true state generating the observation.

function” over the space of beliefs, which assigns a value and action to each belief. By iteratively updating the value function appropriately, the value function can be made to converge to the greatest expected reward from each belief, and the action that will achieve that reward in expectation. The POMDP finds a policy that maximises the expected sum of future (possibly discounted) rewards of the agent executing the policy; for the problem of finding people, we can write a reward function for each possible configuration of the world such that the maximum reward is achieved for finding people fastest.

There are a large number of value function approaches [6] [7] that explicitly compute the expected reward of every belief. Such approaches produce complete policies (the optimal action for every belief), and can guarantee this optimality under a wide range of conditions. However, *finding* a value function this way is usually computationally intractable [6,8].

Large POMDPs are generally very difficult to solve, especially with standard value iteration techniques. Maintaining a full value function over the high-dimensional belief space entails finding the expected reward of every possible belief under the optimal policy. In reality, most POMDP policies generate only a small percentage of possible beliefs. For example, a mobile robot tracking a person is extremely unlikely to ever encounter a belief about the person’s pose that resembles a checkerboard. If the execution of a POMDP is viewed as a trajectory inside the belief space, trajectories for most large, real world POMDPs lie on low-dimensional manifolds embedded in the belief space. So, POMDP algorithms that compute a value function over the full belief space do a lot of unnecessary work.

3 Dimensionality Reduction

In order to find the low-dimensional manifold for representing our belief space, we take advantage of dimensionality reduction techniques. One possible technique that we could consider is Principal Component Analysis¹ (PCA). We collect a data set of beliefs X , and use PCA to find a low-dimensional representation; so long as the collected data set is representative of the beliefs we will encounter during the

¹ Also known as Singular Value Decomposition

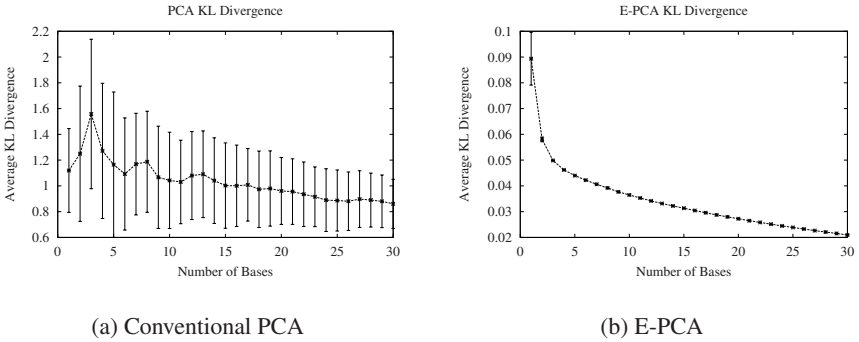


Fig. 3. A comparison of the reconstruction quality of conventional PCA and E-PCA, using a probabilistic distance measure, Kullback-Leibler divergence, on a people-tracking data set. Notice that even with 30 bases, the PCA performs poorly and has very high variance in reconstruction quality. The E-PCA error falls rapidly initially, and the variance in error is low, indicating consistent performance across the entire data set. (Note the different scales on the Y axes.)

execution of the people-finding plan, then we should be able to track the current belief on the low-dimensional manifold accurately.

PCA operates by finding a set of feature vectors $U = \{u_1, \dots, u_n\}$ that minimise the loss function

$$L(U, V) = \|X - UV\|^2 \quad (1)$$

where X is the original data and V is the matrix of low-dimensional coordinates of X . This particular loss function assumes that the data lie near a linear manifold, and that displacements from this manifold are symmetric and have the same variance everywhere. (For example, i.i.d. Gaussian errors satisfy these requirements.) Unfortunately, probability distributions for POMDPs rarely form a linear subspace. In addition, squared error loss is inappropriate for modelling probability distributions: it does not enforce positive probability predictions.

We use exponential family PCA to address this problem. Other nonlinear dimensionality-reduction techniques [9,10] could also work for this purpose, but would have different domains of applicability. Exponential family Principal Component Analysis [3] (E-PCA) varies from conventional PCA by adding a link function, in analogy to generalised linear models, and modifying the loss function appropriately. As long as we choose a link function that corresponds to an exponential family distribution log likelihood, and as long as the link and loss functions to match each other, there will exist efficient algorithms for finding U and V given X . By picking particular link functions (with their matching losses), we can reduce the model to an SVD.

In our case the entries of X are non-negative, and we wish to ensure accurate representation of low-probability events. Consequently, a link and loss function that

correspond to the Poisson distribution are most appropriate.² The corresponding link function is

$$\bar{X} = f(UV) = \exp(UV) \quad (2)$$

(taken component-wise) and its associated loss function is

$$L(U, V) = \exp(UV) - X \circ UV \quad (3)$$

where the “matrix dot product” $A \circ B$ is the sum of products of corresponding elements. It is worth noting that using the Poisson loss for dimensionality reduction is related to Lee and Seung’s non-negative matrix factorisation [11]. Gordon [12,4] has a Newton’s Method solution for computing U and V quickly.

In figures 3 and 4 we compare the ability of PCA and E-PCA to represent belief states for person tracking. We collected 500 sample beliefs from the environment shown in figure 4, then tried to compress them using both PCA and E-PCA. Figure 3(a) shows the average Kullback-Leibler divergence (a distance metric for probability distributions) between the high-dimensional belief and its reconstruction for conventional PCA. We see that the distance is large, does not improve quickly with more dimensions, and the representation quality is inconsistent (as shown by the wide error bars). Figure 3(b) shows the same evaluation (average KL divergence) where E-PCA was used to find the low-dimensional representation. In this case, the error is small, improves quickly initially, and is consistent across the entire data set, in all ways outperforming conventional PCA.

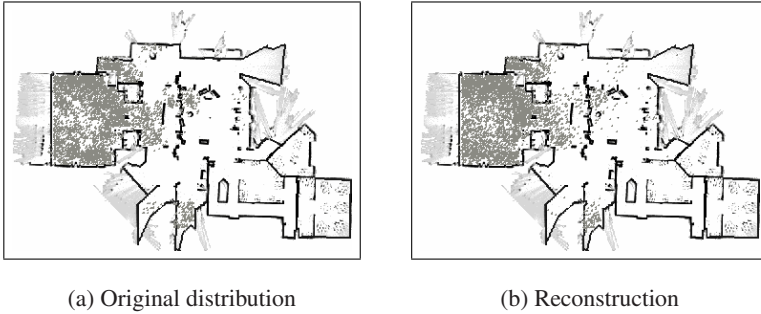


Fig. 4. Examples of distributions in the Longwood at Oakmont retirement facility. The small grey dots show particles drawn from the original distribution; the higher the probability, the denser the particles. (a) An example distribution of potential positions of the person being searched for. This distribution is represented using 1961 dimensions. (b) The same distribution, reconstructed using only 6 dimensions. The true position of the person is not observable by the robot at a distance.

² Examples of other choices are the Exponential distribution, the multinomial, the Beta, etc. The Gaussian is also an Exponential family distribution, but a Gaussian link and loss function reduce to conventional PCA.

Figure 4 shows an example of the tracking process in progress. The true position of the person is unknown, and the robot instead maintains a probability distribution over possible poses of the person. The small grey dots show particles drawn from the original distribution. As the robot moves around the environment, sensor information is integrated into the distribution. The space of possible distributions is 1961-dimensional: the environment $53 \times 37m$ discretised into $1m$ grid cells, and the belief has 1 dimension for each grid cell. However, by taking advantage of the E-PCA decomposition we can generate a faithful representation of the space of actual distributions in only 6 dimensions. Figure 4(b) shows the original distribution projected to the low-dimensional space and then reconstructed. Although this is a lossy projection, the reconstruction is accurate for planning purposes. Remember that the task is not to reconstruct only the distribution shown in figure 4(a), but to be able to represent all of the distributions that we expect to see as points in the 6-dimensional space.

4 Planning

Given the belief features acquired through E-PCA, it remains to compute the policy. Unfortunately, the non-linearity of the E-PCA projection prevents any guarantees of value function convexity over the low-dimensional space, which means that standard POMDP value iteration techniques cannot be used to find policies on the low-dimensional manifold directly. Instead, we approximate the low-dimensional space discretely, converting the POMDP into a belief space MDP. During execution, the action taken at each time step is taken from the discrete belief state that is closest to the current actual belief.

Our conversion algorithm from POMDP to MDP is a variant of the Augmented MDP, or Coastal Navigation algorithm [13], using belief features instead of entropy. We can compute the model reward function $\mathcal{R}(s_i)$ easily from the reconstructed beliefs, using $\mathcal{R}(b) = b \cdot \mathcal{R}(s)$. To learn the transition function $p(b_i|a, b_j)$, we can sample states from the reconstructed beliefs, sample observations from those states, and incorporate those observations to produce new belief states. Table 1 outlines the steps of this algorithm.

Table 1. Algorithm for planning in low-dimensional belief space.

1. Collect sample beliefs
2. Use E-PCA to generate low-dimensional belief features
3. Convert low-dimensional space into discrete space \mathcal{S}
4. Learn belief transition probabilities $\mathcal{T}(s_i, a, s_j)$, and reward function $\mathcal{R}(s_i)$.
5. Perform value iteration on new model, using states \mathcal{S} , transition probabilities \mathcal{T} and \mathcal{R} .

The state space can be discretised in a number of ways, such as laying a grid over the belief features or using distance to the closest training beliefs to divide feature space into Voronoi regions. Thrun [14] has proposed nearest-neighbour discretisation in high-dimensional belief space; we propose instead to use nearest-neighbour in a low-dimensional feature space, where neighbours should be more closely related.

In order to find a good policy, we must be sure to discretise carefully. In some regions of the low-dimensional manifold, beliefs that are close together we can cluster into the same, large discrete cell without hurting performance. In other regions of the belief space, the cells must be much smaller, in order to distinguish different beliefs that require different actions. This leads to a variable resolution representation of the low-dimensional manifold.

We typically do not have enough belief samples initially to determine the full discretisation across the entire space; in places the discretisation will be insufficiently fine. We compensate by periodically re-evaluating the model at each grid cell, and splitting the grid-cell into smaller discrete cells where the model disagrees with some statistics of the real world. A number of different statistics have been suggested for testing the model against data from the real world [15], such as reduction in reward variance, or value function disagreement. We have opted instead for a simpler criterion of transition probability disagreement, although one improvement we are exploring is to use the Kolmogorov-Smirnov criterion for reducing expected reward disagreement [16].

5 Performance

Figure 5(a) shows an example trajectory for a simple environment³. Even for this very simple problem, the trajectory is relatively complicated. The robot starts at the far end of the corridor, with the person's position completely unknown (the initial belief is uniform over the entire space). The robot travels past the open door on the right, part way down the corridor, returns to explore the room, and then finishes the corridor. This trajectory ensures that by the time the robot is finished exploring the room, the person must either have been found, or be at the far end of the corridor – there is no possibility for the person to escape into already-explored sections of the environment. This is an example of the kind of planning we hope to see – our planner has found a strategy that is not obvious, nor easy to capture using simple heuristics. Figure 5(b) shows a more obvious but sub-optimal trajectory in mid-execution. Notice the probability mass that appears in the already-explored region near the robot start location, causing the robot eventually to retrace its steps. The optimal strategy in figure 5(a) explicitly avoids this problem.

Figure 6 shows a quantitative comparison of our technique and other possible heuristics. The horizontal line is the baseline, “True MDP” situation where the position of the person is always known correctly, that is, there is no hidden state. This algorithm is essentially cheating, but serves as a useful lower bound in that the robot find the person as quickly as possible every iteration. The “Closest” heuristic takes the robot to the nearest grid cell where the person might be. The “Densest” heuristic takes the robot to the location where the most particles are visible. The “MDP” heuristic takes the robot to the maximum-likelihood location (the single grid cell with the most particles). The “E-PCA 72” and “E-PCA 260” is a comparison of the E-PCA plans before state splitting (with 72 low-dimensional belief states) and

³ For this environment, the original space was $47m \times 17m$ with a $0.2m$ resolution, for 20,230 grid cells, reduced to 6 dimensions.

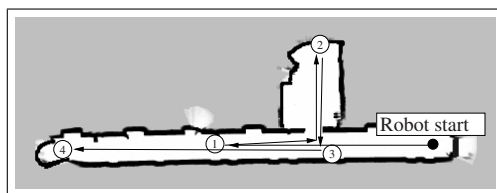
after iterative refinement of the manifold (to 260 low-dimensional belief states). The “E-PCA 260” is clearly the best performing algorithm, able to find the person almost as quickly as the fully-observable planner.

6 Related Work

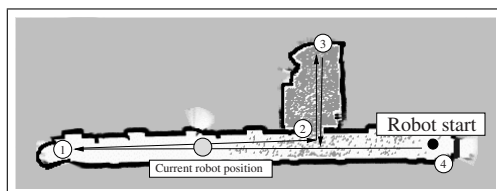
There have been a number of recent advances in solving large POMDPs. Poupart & Boutilier [17] make use of a similar dimensionality reduction technique, however, their representation requires a linear combination of bases to represent arbitrary data, which is a strong limitation on the compression they can achieve. (Figure 3(a) demonstrates the limitation of linear representations.) Pineau et al. [18] have had success in finding approximate value functions quickly, but again their approach has not scaled to the size of the problems discussed in this paper.

Policy search algorithms [8,19] have addressed some large problems. We suggest that a large part of the success of policy search is due to the fact that it focuses computation on relevant belief states. A disadvantage of policy search, however, is that can be data-inefficient across problems: many policy search techniques have trouble reusing sample trajectories generated from old policies. Our approach focuses computation on relevant belief states, but also allows us to use all relevant training data to estimate the effect of any policy.

Related research has developed heuristics which reduce the belief space representation. In particular, entropy-based representations for heuristic control [20] and



(a) Optimal trajectory



(b) Sub-optimal trajectory

Fig. 5. Example trajectory. (a) Even for this very simple environment, in order to maximise the likelihood of finding the person, the trajectory is relatively complicated. (b) The more obvious, sub-optimal trajectory allows some probability mass to “leak” into already-explored regions.

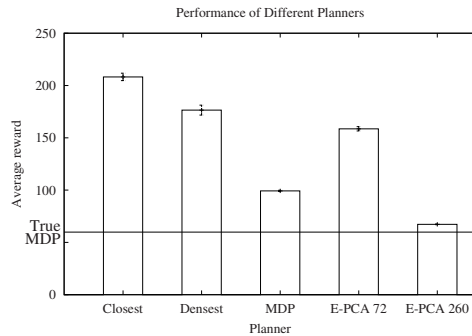


Fig. 6. A comparison of different planning methods, including some simple heuristic planners. The “True MDP” method is the lower-bound “cheating” solution which assumes that the true position of the robot is always known. The optimal method under uncertainty is the “E-PCA-260” method, which also learns the optimal state decomposition.

full value-function planning [13] have been tried with some success. However, these approaches make strong assumptions about the kind of uncertainties that a POMDP generates. By performing principled dimensionality reduction of the belief space, our technique should be applicable to a wider range of problems.

7 Conclusion

We have demonstrated a system for finding and tracking people in the health care setting. The problem of finding people is computationally difficult in many environments, because of the high degree of uncertainty. Planners that do not reason intelligently about this uncertainty can take arbitrarily long to perform such real world tasks. The Partially Observable Markov Decision process is a planner that can reason about uncertainty, but is typically held not to scale to large problems.

We have shown that by taking advantage of dimensionality reduction techniques, we can represent POMDP problems compactly, and therefore generate good plans. We used a variant of PCA called Exponential family PCA (E-PCA) to find a low-dimensional manifold one which typical beliefs lie, and compute a value function over that manifold using a function approximator. We have also shown that naive function approximation is not sufficient for finding good plans. Our experimental results indicate that the optimal plan can be sensitive to small changes to the function approximation in different regions of the low-dimensional manifold. By appropriate use of statistical tests, we are able to find good variable resolution representations for the value function that lead to good policies.

Acknowledgements

Nicholas Roy was funded by the National Science Foundation under ITR grant # IIS-0121426. Geoff Gordon was supported by AFRL contract F30602-01-C-0219, DARPA’s MICA program.

References

1. Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialog management for robots. In *The Proceedings of the Association for Computational Linguistics*, 2000.
2. Michael Montemerlo, Joelle Pineau, Nicholas Roy, Sebastian Thrun, and Vandana Verma. Experiences with a mobile robotic guide for the elderly. In *Proceedings of the National Conference of Artificial Intelligence (AAAI 02)*, Edmonton, AB, July 2002.
3. M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal components analysis to the exponential family. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, 2002.
4. Nicholas Roy and Geoffrey Gordon. Exponential family PCA for POMDPs. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, 2003.
5. Matthew Rosencrantz, Geoffrey Gordon, and Sebastian Thrun. Locating moving entities in indoor environments with teams of mobile robots. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 2003.
6. L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.
7. Milos Hauskrecht. Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, 13:33–94, 2000.
8. Andrew Ng and Michael Jordan. PEGASUS: A policy search method for large MDPs and POMDPs. In *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, 2000.
9. Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.
10. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.
11. Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
12. Geoffrey Gordon. Generalized² linear² models. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.
13. Nicholas Roy and Sebastian Thrun. Coastal navigation with mobile robots. In *Advances in Neural Processing Systems 12*, pages 1043–1049, 1999.
14. Sebastian Thrun. Monte Carlo POMDPs. In *Advances in Neural Information Processing Systems 12*, 1999.
15. Remi Munos and Andrew Moore. Variable resolution discretization for high-accuracy solutions of optimal control problems. In *International Joint Conference on Artificial Intelligence (IJCAI 99)*, 1999.
16. A. McCallum. Instance-based utile distinctions for reinforcement learning. In *Proceedings of the 12th International Machine Learning Conference*, Lake Tahoe, CA, 1995.
17. Pascal Poupart and Craig Boutilier. Value-directed compression of POMDPs. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, 2003.
18. Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2003)*, Acapulco, Mexico, August 2003.
19. J. Andrew Bagnell and Jeff Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *Proceedings of the International Conference on Robotics and Automation*, 2001.
20. Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS)*, 1996.

Development of a Personal Service Robot with User-Friendly Interfaces

Jun Miura, Yoshiaki Shirai, Nobutaka Shimada,
Yasushi Makihara, Masao Takizawa, and Yoshio Yano

Dept. of Computer-Controlled Mechanical Systems,
Osaka University, Suita, Osaka 565-0871, Japan
{jun,shirai,shimada,makihara,takizawa,yano}@cv.mech.eng.osaka-u.ac.jp
<http://www-cv.mech.eng.osaka-u.ac.jp/>

Abstract. This paper describes a personal service robot developed for assisting a user in his/her daily life. One of the important aspects of such robots is the user-friendliness in communication; especially, the easiness of user's assistance to a robot is important in making the robot perform various kinds of tasks. Our robot has the following three features: (1) interactive object recognition, (2) robust speech recognition, and (3) easy teaching of mobile manipulation. The robot is applied to the task of fetching a can from a distant refrigerator.

1 Introduction

Personal service robot is one of the promising areas to which robotic technologies can be applied. As we are facing the "aging society", the need for robots which can help human in various everyday situations is increasing. Possible tasks of such robots are: bringing a user-specified object to the user in the bed, cleaning a room, mobile aid, social interaction.

Recently several projects on personal service robots are going on. HERMES [2,3] is a humanoid robot that can perform service tasks such as delivery using vision- and conversation-based interfaces. MORPHA project [1] aims to develop two types of service robot: robot assistant for household and elderly care and manufacturing assistant, by integrating various robotics technologies such as human-machine communications, teaching methodologies, motion planning, and image analysis. CMU's Nursebot project [10] has been developing a personal service robot for assisting elderly people in their daily activities based on communication skills; a probabilistic algorithm is used for generating a timely and user-friendly robot behaviors [9].

One of the important aspects of such robots is the user-friendliness. Since personal service robots are usually used by a novice, they are required to provide easy interaction methods to users. Moreover, personal service robots are expected work in various environments and, therefore, it is difficult to give a robot a complete set of required skills and knowledge in advance; so teaching the robot *on the job* is indispensable. In other words, *user's assistance to a robot* is necessary and should be done easily.

We are developing a personal service robot which has the following three features (see Fig. 1):



Fig. 1. Features of our personal service robot.

1. Interactive object recognition.
2. Robust speech recognition.
3. Easy teaching of mobile manipulation.

The following sections will describe these features and experimental results.

The current target task of our robot is fetching a can or a bottle from a distant refrigerator. The task is roughly divided into the following: (1) movement to/from the refrigerator, (2) manipulation of the refrigerator and a can, (3) recognition of a can in the refrigerator. This paper focuses on the last two subtasks. In the third task, a verbal interaction between the user and the robot is essential to the robustness of the recognition process.

2 Interactive Object Recognition

This section explains our interactive object recognition method which actively utilizes the dialog with a user [6].

2.1 Registration of Object Models

The robot registers object models in advance. A model consists of the size, representative colors (primary features), and secondary features (the color, the position, and the area of uniform regions other than primary features). Secondary features are used only when there are multiple objects with the same primary feature. For model registration, the robot makes a “developed image” by mosaicing images captured from eight directions while the robot rotates an object (see Fig. 2).

Since primary and secondary features depend on the viewing direction, we determine intervals of directions where similar features are observed. In the case of Fig. 3(a), for example, two intervals, I_1 (white) and I_2 (blue), are determined. Fig. 3(b) shows an extraction result of features, which are indicated in the lines below the image; the top line indicates the intervals of the primary feature and the middle and the bottom line indicate those of two secondary features. Secondary features of an object are incrementally registered to its model every time another object is added to the database which has a similar feature and is undistinguishable in several viewing directions.



Fig. 2. Procedure for constructing a developed image for a can.

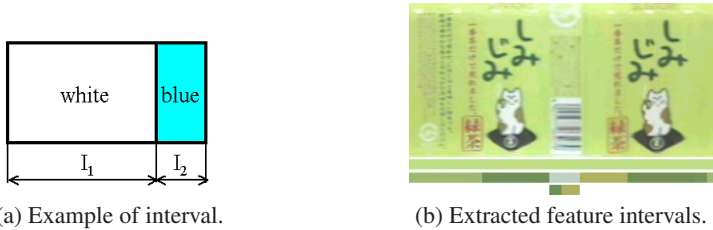


Fig. 3. Extraction of features

2.2 Object Recognition

The robot first extracts candidate regions for objects based on the object color which is specified by a user or is determined from a user-specified object name. Then the robot determines the type of each candidate from its shape; for example, a can has a rectangular shape in an image.

For each candidate, the robot checks if its size is comparable with that of the corresponding object model. If no secondary features are registered in the model, the recognition finishes with success. Otherwise, the robot tries matching using secondary features. Fig. 4 shows an example matching process. Fig. 4(b) shows two candidates are found using only the primary feature (representative color). Using a secondary feature (black region in Fig. 4(c)), the two candidates are distinguished.

Since the lighting condition in the recognition phase may differ from that in the learning phase, we have developed a method for adjusting colors based on the observed color of a reference object such as the door of a refrigerator [7].

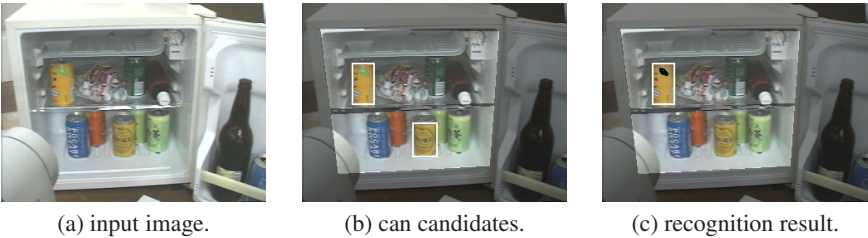


Fig. 4. Recognition of a can.



Fig. 5. Recognition of an occluded object.

2.3 Recognition Supported by Dialog

If the robot failed to find a target object, it tries to obtain additional information by a dialog with the user. Currently, the user is supposed to be able to see the refrigerator through a remote display. We consider the following failure cases: (1) multiple objects are found; (2) no objects are found but candidate regions are found; (3) no candidate regions are found due to (a) partial occlusion or (b) color change.

In this dialog, it is important for the robot to generate *good* questions which can retrieve an informative answer from the user. We here explain case (3)-(a) in detail. In this case, the robot asks the user an approximate position of the target like: “I have not found it. Where is it ?” Then the user may answer: “It is behind A” (A is the name of an occluding object). Using this advice, the robot first searches for object A in the refrigerator (see Fig. 5(b)). Then the robot searches both sides of the occluding object for the target object and determine its position (see Fig. 5(c)).

3 Robust Speech Recognition

Many existing dialog-based interface systems assume that a speech recognition (sub)system always works well. However, since the dialog with a robot is usually held in environments where various noises exist, such an assumption is difficult to be made. There is another problem that a user, who is usually not an expert of robot operations, most probably uses words which are not registered in the robot’s database. Therefore, the dialog system has to be able to cope with speech recognition failure and unknown words [11].

3.1 Overview of the Speech Recognition

We use IBM’s ViaVoice as a speech recognition engine. Fig. 6 shows the overview of our speech recognition system. We first apply a context-free grammar (CFG)-based recognition engine to the voice input. If it succeeds, the recognition result is sent to the image recognition module. If it fails to identify some words due to, for example, noise or unknown words, the input is then processed by a dictation-oriented engine, which generates a set of probable candidate texts. Usually in a candidate text, some words are identified (i.e., determined to be registered ones) and the others are not.

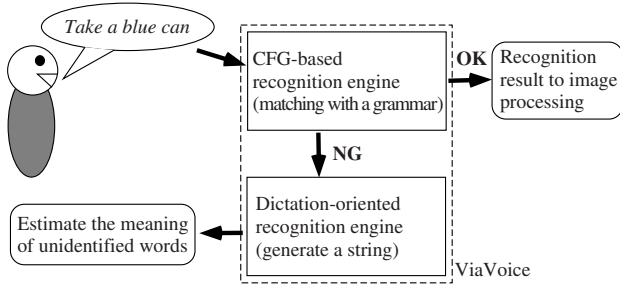


Fig. 6. Speech recognition system.

So the unidentified words are analyzed to estimate their meanings, by considering the relation to the other identified words. Then the robot generates a query to the user to verify the estimation. The robot uses probabilistic models of possible word sequences and updates the model through the dialog with each specific user.

3.2 Estimating the Meaning of Unidentified Words

We consider that an unidentified word arises in the following three cases: (1) a known word is erroneously recognized; (2) an unknown word is uttered which is a synonym of a known word; (3) noise is erroneously recognized as a word. In addition, we only consider the case where one or two consecutive unidentified word(s) exist in an utterance. The robot evaluates the first two cases and selects the estimation with the highest evaluation value, if the value is above a certain threshold. Otherwise, the unidentified word is considered to come from noise.

This estimation problem is formulated as finding the registered word W with the maximum probability, given state S , context γ , and a text string R generated by the dictation-oriented engine. State S indicates a possible state in the dialog such as the one where the robot is waiting for the user's first utterance or the one where it is waiting for an answer to its previous question like "which one shall I take ?" Context γ is *identified* words before and after an unidentified one under consideration.

Fig. 7 illustrates the estimation of category C and word W using the following probabilistic models:

- $P_{c-p}(C_p|C)$ is the probability that C_p is uttered just before C .
- $P_{c-n}(C_n|C)$ is the probability that C_n is uttered just after C .
- $P_{w-p}(W_p|W)$ is the probability that W_p is uttered just before W .
- $P_{w-n}(W_n|W)$ is the probability that W_n is uttered just after W .

For case (1) (i.e., erroneous recognition of a registered word), we search for the word \hat{W} which is:

$$\hat{W} = \arg \max_W P(W|S, \gamma, R). \quad (1)$$

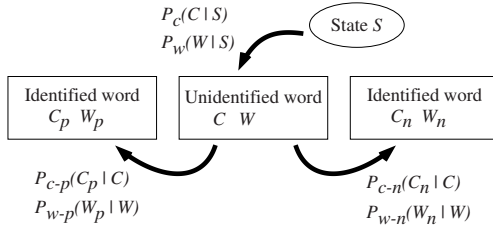


Fig. 7. Estimation of category C and word W

For case (2) (i.e., use of a synonym of a registered word), we search for the word \hat{W} which is:

$$\hat{W} = \arg \max_W P(W|S, \gamma). \quad (2)$$

We here further examine only eq. (1) due to space limitation. Eq. (1) is rewritten as:

$$\begin{aligned} P(W|S, \gamma, R) &= \frac{\sum_C \{P(W|S, \gamma, C)P(C|S, \gamma)\}P(R|W, S, \gamma)}{\sum_W P(W, R|S, \gamma)} \\ &\simeq \frac{\sum_C \{P(W|S, \gamma, C)P(C|S, \gamma)\}P(R|W)}{\sum_W P(W, R|S, \gamma)}, \end{aligned} \quad (3)$$

where \sum_C indicates the summation for categories C whose probability $P(C|S, \gamma)$ is larger than a threshold, and \sum_W indicates the summation for words W belonging to the categories. Eq. (3) is obtained by considering that a recognized text R depends almost only on word W ; $P(R|W)$ is called a *pronunciation similarity*.

An example of recognizing an unidentified word is as follows. A user asked the robot to take a blue PET bottle, by uttering “AOI (blue) PETTO BOTORU (PET bottle) WO TOTTE (take)”. But the robot first recognized the utterance as “OMOI KUU TORABURU WO TOTTE”. Since this includes unidentified words, the robot estimates their meanings using the proposed method, and reached the conclusion that “OMOI” means “AOI” and “KUU TORABURU” means “PETTO BOTORU”.

The recognition result of unidentified words are fed back to the system to update the database and the probabilistic models [11].

4 Easy Teaching of Mobile Manipulation

Usually service robots have to deal with much wider range of tasks (i.e., operations and environments) than industrial ones. An easy, user-friendly teaching method is, therefore, desirable for such service robots. Among previous teaching methods, direct methods (e.g., the one using a teaching box) are intuitive and practical but requires much user’s effort, while indirect methods (e.g., teaching by demonstration [5,4]) are easy but still needs further improvement of the robot’s ability for deployment.

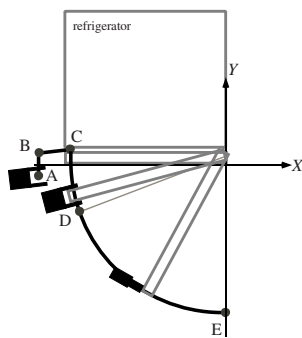


Fig. 8. A nominal trajectory.

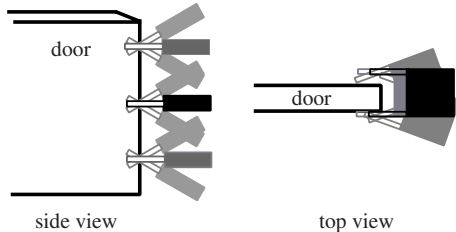


Fig. 9. Tolerance for segment CD in Fig. 8.

We, therefore, develop a novel teaching method for a mobile manipulator which exists in between the above two approaches. In the method, the user teaches the robot a nominal trajectory of the hand and its *tolerance* to achieve the current target task. The given nominal trajectory may be infeasible due to the structural limitation of the robot; so the robot searches for a feasible one within the tolerance. Only when the robot fails to find any feasible trajectory, it plans a movement of the mobile base, by using the redundancy provided by the mobile base as another tolerance. The teaching method is well intuitive and does not require much user's effort because the user does not have to consider the structural limitation of the robot in teaching. At the same time, the method does not assume a high recognition and inference ability of the robot because the given nominal trajectory has much information for motion planning; the robot does not need to generate a feasible trajectory from scratch.

The following subsections explain the teaching method, using the task of opening the door of a refrigerator as an example.

4.1 Nominal Trajectory

A nominal trajectory is the trajectory of the hand pose (position and orientation) in a 3D object-centered coordinate system. To simplify the trajectory teaching, we currently set a limitation that a trajectory of hand position is composed of circular and/or straight line segments. Fig. 8 shows a nominal trajectory for opening a door, composed of straight and circular segments on some horizontal planes; on segment CD, the robot roughly holds the door, while on segment DE, the robot pushes it at a different height. The hand orientation is also specified as shown in the figure.

4.2 Tolerance

A tolerance indicates acceptable deviations from a nominal trajectory to perform a task; if the hand exists within the tolerance over the entire trajectory, the task is achievable. A user teaches a tolerance without explicitly considering the structural limitation of the robot. Given a nominal trajectory and its tolerance, the robot searches for a feasible trajectory.

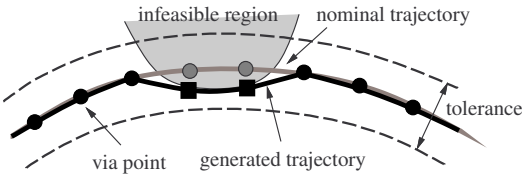


Fig. 10. Example feasible regions.

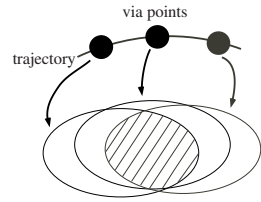


Fig. 11. A feasible region.

The user sets a tolerance to each straight or circular trajectory using a coordinate system attached to each point on the segment. In these coordinate systems, a user can teach a tolerance of positions relatively intuitively as a kind of the *width* of the nominal trajectory. Fig. 9 shows an example of setting a tolerance for circular segment CD in Fig. 8, which is for opening the door.

4.3 Generating Feasible Trajectories

The robot first tries to generate a feasible trajectory within a given tolerance. When the robot fails to find a feasible one, it divides the trajectory into sub-trajectories such that each sub-trajectory can be performed without movement of the base; it also plans the movement between performing sub-trajectories.

On-line Trajectory Generation The robot sets via points on the trajectory with a certain interval, and generates a feasible trajectory by iteratively searching for feasible hand poses for the sequence of via points. This trajectory generation is performed on-line because the relative position between the robot and the manipulated objects may vary from time to time. The robot estimates the relative position before trajectory generation. The previously calculated trajectories are used as guides for efficiently calculating the current trajectory.

Fig. 10 illustrates how a feasible trajectory is generated; small circles indicate via points on the given nominal trajectory, two dashed lines indicate the boundary of the tolerance, the hatched region indicates the outside of the range of possible hand poses. A feasible trajectory is generated by searching for a sequence of hand poses which are in the tolerance and near to the given via points. In the actual trajectory generation, the robot searches the six dimensional space of hand pose.

During executing the generated trajectory, it is sometimes necessary to estimate the object position. Currently, we manually give the robot a set of necessary sensing operations for the estimation.

Trajectory Division Based on Feasible Regions The division of a trajectory is done as follows. For each via point, the robot calculates a region on the floor in the object coordinates such that if the mobile base is in the region, there is at least one feasible hand pose. By calculating the intersection of the regions, the robot determines the region on the floor where the robot can make the hand follow the

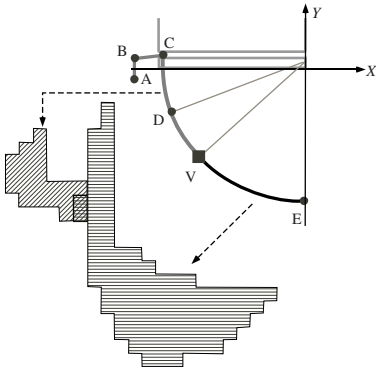


Fig. 12. Example feasible regions.

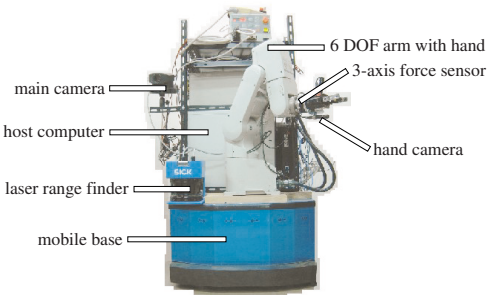


Fig. 13. Our service robot.

entire trajectory. Such an intersection is called a *feasible region* of the task (see Fig. 11). The robot continuously updates the feasible region, and if its size becomes less than a certain threshold, the trajectory is divided at the corresponding via point. Fig. 12 shows example feasible regions of the trajectory of opening the door shown in Fig. 8. The entire trajectory is divided into two parts at point *V*; two corresponding feasible regions are generated.

5 Prototype System and Experiments

Fig. 13 shows our personal service robot. The robot is a self-contained mobile manipulator with various sensors. In addition to the above-mentioned functions, the robot needs an ability to move between a user and a refrigerator. The robot uses the laser range finder (LRF) for detecting obstacles and estimating the ego-motion [8]. It uses the LRF and vision for detecting and locating refrigerators and users. Fig. 14 shows snapshots of the operation of fetching a can from a refrigerator to a user.

6 Summary

This paper has described our personal service robot. The robot has user-friendly human-robot interfaces including interactive object recognition, robust speech recognition, and easy teaching of mobile manipulation.

Currently the two subsystems, object and speech recognition and teaching of mobile manipulation, are implemented separately. We are now integrating these two subsystems into one prototype system for more intensive experimental evaluation.

Acknowledgment

This research is supported in part by Grant-in-Aid for Scientific Research from Ministry of Education, Culture, Sports, Science and Technology, and by the Kayamori Foundation of Informational Science Advancement.

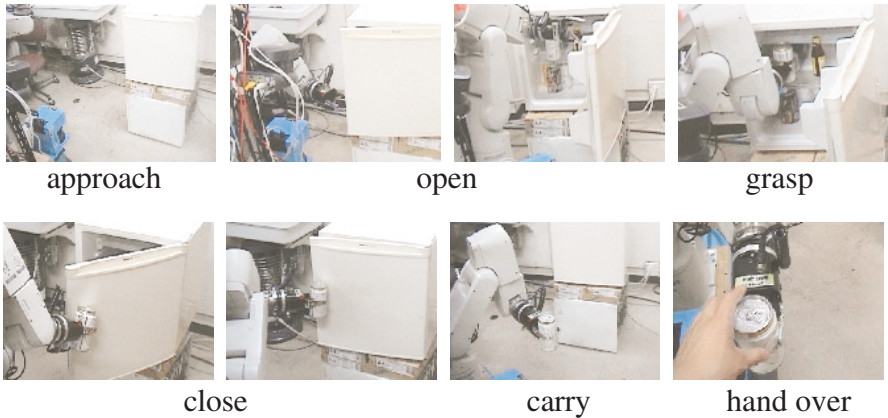


Fig. 14. Fetch a can from a refrigerator.

References

1. Morpha project, <http://www.morpha.de/>.
2. R. Bischoff. Hermes – a humanoid mobile manipulator for service tasks. In *Proc. of FSR-97*, pp. 508–515, 1997.
3. R. Bischoff and V. Graefe. Dependable multimodal communication and interaction with robotic assistants. In *Proc. of ROMAN-2002*, pp. 300–305, 2002.
4. M. Ehrenmann et al. Teaching service robots complex tasks: Programming by demonstration for workshop and household environments. In *Proc. of FSR-2001*, pp. 397–402, 2001.
5. K. Ikeuchi and T. Suehiro. Toward an assembly plan from observation part i: Task recognition with polyhedral objects. *IEEE Trans. on Robotics and Automat.*, Vol. 10, No. 3, pp. 368–385, 1994.
6. Y. Makihara et al. Object recognition supported by user interaction for service robots. In *Proc. of ICPR-2002*, pp. 561–564, 2002.
7. Y. Makihara et al. Object recognition in various lighting conditions. In *Proc. of SCIA-2003*, pp. 899–906, 2003.
8. J. Miura, Y. Negishi, and Y. Shirai. Mobile robot map generation by integrating omnidirectional stereo and laser range finder. In *Proc. of IROS-2002*, pp. 250–255, 2002.
9. J. Pineau et al. Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems*, Vol. 42, No. 3–4, pp. 271–281, 2003.
10. N. Roy et al. Towards personal service robots for the elderly. In *Proc. of WIRE-2000*, 2000.
11. M. Takizawa et al. A service robot with interactive vision – object recognition using dialog with user –. In *Proc. of Workshop on Language Understanding and Agents for Real World Interaction*, pp. 16–23, 2003.

An Enhanced Robotic Library System for an Off-Site Shelving Facility

Jackrit Suthakorn¹, Sangyoon Lee², Yu Zhou², Sayeed Choudhury³,
and Gregory S. Chirikjian²

¹ Department of Mechanical Engineering,
Mahidol University, Bangkok, Thailand
song@jhu.edu or jackrit@trs.ac.th

² Department of Mechanical Engineering
The Johns Hopkins University, Baltimore, Maryland 21218 USA
slee@konkuk.ac.kr, yuzhou@titan.me.jhu.edu, gregc@jhu.edu

³ Digital Knowledge Center of the Sheridan Libraries
The Johns Hopkins University, Baltimore, Maryland 21218 USA
sayeed@jhu.edu

Abstract. This paper describes our continued work of a unique robotics project, Comprehensive Access to Printed Materials (CAPM), within the context of libraries. As libraries provide a growing array of digital library services and resources, they continue to acquire large quantities of printed material. This combined pressure of providing electronic and print-based resources and services has led to severe space constraints for many libraries, especially academic research libraries. Consequently, many libraries have built or plan to build off-site shelving facilities to accommodate printed materials. An autonomous mobile robotic library system has been developed to retrieve items from bookshelves and carry them to scanning stations located in the off-site shelving facility. This paper reviews the overall design of the robot system and control systems, and reports the new improvement in the accuracy of the robot performance; in particular, the pick-up process.

1 Introduction

As libraries provide a growing array of digital library services and resources, they continue to acquire large quantities of printed material. This combined pressure of providing electronic and print-based resources and services has led to severe space constraints for many libraries, especially academic research libraries. Consequently, many libraries have built or plan to build off-site shelving facilities to accommodate printed materials. However, given that these locations are not usually within walking distance of the main library, access to these materials, specifically the ability to browse, is greatly reduced. Libraries with such facilities offer extensive physical delivery options from these facilities, sometimes offering multiple deliveries per day. Even with such delivery options, the ability to browse in real-time remains absent. The goal of the CAPM Project is to build a robotic, on-demand and batch scanning system that will allow for real-time browsing of printed materials through a web interface. We envisage the system will work as follows: an end user will identify that a monograph is located in an off-site facility. The user will engage the CAPM system that, in turn, will initiate a robot that will retrieve the requested item. The robot will deliver this item to another robotic system that will open the item and turn the pages automatically. By using existing scanners, optical character recognition (OCR) software, and indexing software developed by the Digital Knowledge Center (a research and development unit of the Sheridan Libraries at Johns Hopkins), the

CAPM system will not only allow for browsing of images of text, but also for searching and analyzing of full-text generated from the images.

The details of the mechanical structure, the navigation system, control and software, simulations, experiments and results of the robot were previously described in [1]. This paper focuses on improvement in the accuracy of entire delivery procedure of the robot system, in particular, the pick-up of bookcases, while the future work, will be concentrated on developing the robotic system to complete the remaining processes.

Since the CAPM robot is designed to work in an off-site shelving facility that belong to the Johns Hopkins University, several assumptions in the design are made based on the actual environment of this facility. All the paths in the facility are assumed to be smooth and flat. Each book is assumed to be stored in a special case, which has a pair of wing-like handles for engaging with a passive gripper. It is assumed that each item is stored in a specifically designed case and arranged side-by-side with a small in-between gap. Finally, a barcode is attached to each case.

Currently at the Moravia Park shelving facility, after receiving a request, a library officer at the facility will drive a portable personnel lift to retrieve the requested item to its location, and then bring it to a waiting area for the next scheduled transportation. Then, a batch of requested items is transferred to the main library. In the same manner, our robot will be initially parked at the docking station until an item is requested. The robot is equipped with a database system of book locations and a global map of the off-site shelving facility. After receiving a request, the robot will autonomously run along a known path to the book location and retrieve the requested item from the shelf. Then, the robot will carry the item back to the scanning station and then return to the docking station.

This is not the first time a robot has been built to perform a specific service function. In 1995, Hansson introduced an industrial robot in a Swedish library [2]. Safaric [3] presented an example of a telerobot controlled via Internet [4]. Byrd introduced a successful service robot used to survey and inspect drums containing low-level radioactive waste stored in warehouses at Department of Energy facilities [5].

The CAPM system differs from other existing systems in the following ways. First, the system retrieves individual items, as opposed to boxes of items, such as the system at the California State University at Northridge [6]. Second, the CAPM system does not assume an existing or fixed shelving and space arrangement. This flexibility will allow it to work in many diverse environments. Third, the CAPM retrieval robot is an autonomous system. Fourth, the economic analysis by a collaborating research group in the Department of Economics at Johns Hopkins University has verified that a *relatively* inexpensive robotic system is cost-effective, especially in comparison to potential benefits. Finally, the page-turning system, to be built in the future, will accommodate a wide variety of paper types and materials.

In subsequent sections of this paper, we report the design, control systems, experiments and results of an autonomous robotic library system for an off-site shelving facility. Sections 2 and 3 briefly review the robot design, the robot control systems and software and navigation system (detailed descriptions can be found in [1, 6].) Section 4 explains improvement in the accuracy of entire delivery procedure of the robot system. We then report experiments and results in Section 5.

2 Hardware of the Robotic System

2.1 Mechanical Structure

This section presents designs and descriptions of two major components of the CAPM library robot: the manipulator arm and the locomotion device.

2.1.1 Manipulator arm system

In order to retrieve books from bookshelves and carry them to the scanning station, a specific manipulator arm system was designed. Since each bookshelf is 10-foot-high, a vertical translation system (VTS) was used to move the robot manipulator to different altitudes. The VTS is a sliding rod with an electric motor for driving a lead-screw rod. An enhanced commercial 6-DOF robot manipulator, the F3 made by CRS Robotics, Inc., is affixed to a platform which is a part of the vertical translation system (See Figure 1.)

We built and installed a passive gripper to the end-effector of the robot manipulator. The gripper is used to passively grasp the bookcase. The structures of the gripper and bookcases were designed to fit to each other. A barcode scanner is installed inside the gripper in order to recognize and ensure the precision of picking a requested item.

2.1.2 Locomotion device

The locomotion device is responsible for the gross motion of the robot. We have modified a commercial servo-controlled mobile robot platform, the Labmate made by Helpmate Inc. An aluminum-alloy cart is built and attached to the Labmate mobile platform. This cart is used to store the robot manipulator controller and the power source while the Labmate mobile platform is used as the base of the manipulator arm system. A ranging sensor system was installed on the mobile platform to collaborate and improve the navigation system. All electronic devices used to control the vertical translation system and sensor systems were installed on the mobile platform. Because of the installation of a power source onboard, the robot does not require an external power line while working. Figure 1 shows the overall mechanical structure of the library robot.

In our work, due to limits of sensor performance, 8 sensors are used: 4 sonar sensors and 4 infrared sensors. One sonar and one infrared sensor are paired together to get each of the 4 sensor readings needed. This is done because of the distance measuring limits which each have. The Polaroid 6500 sonar sensor has a range of 15-1067 cm while the Sharp GP2D02 infrared sensor has a range of 10-80 cm. It can be observed that by using these two sensors combined, we can achieve a range of 10-1067 cm of reliable distance measurement. Each sensor is controlled and interfaced to the main computer via a micro-controller (BASIC Stamp II).

3 Control and Software

All the processes and activities of the system are controlled by an onboard Intel Pentium II laptop. The control systems of the library robot consist of several sub-controllers: the control system of the VTS, the control system of the robot

manipulator, the control system of the mobile platform, the high-level control system of the library robot, and the control software.

3.1 Control of the VTS

The VTS is required in the library robot to move the robot manipulator to different altitudes. A lead screw lift system was selected and modified to be a major component of the VTS. To enhance the lift system so that it could successfully be implemented as part of an autonomous robotic library system, a feedback control system was integrated. This feedback control system functions to: 1) determine the altitude of the VTS platform, 2) send the real-time ranging information to the high-level controller, and 3) receive a command from the high-level controller to control the motion of the VTS platform in order to reach a desired altitude. The feedback control system of the VTS consists of a BASIC Stamp II micro-controller, an input/output serial communication port, a range sensor system, and a VTS directional controller.

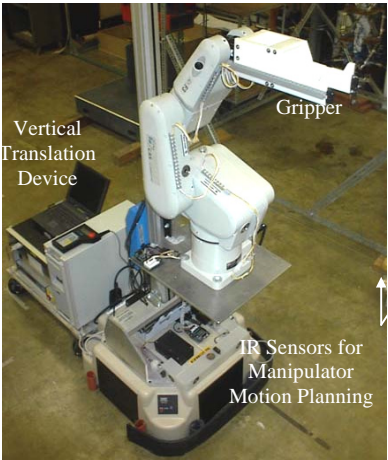


Fig.1. The Robotic Library System

3.2 Control of the Robot Manipulator

The six-axis F3 robot system manufactured by CRS Robotics Inc is used as the control system of the robot manipulator. Articulated joints provide the F3 arm with six degrees of freedom, and absolute encoders mounted on the motor shaft in each joint provide positional feedback to the controller. The F3 robot arm uses the Cartesian coordinate system.

Control programs were written in the C++ language and downloaded to the controller. Control programs use the ActiveRobot interface developed by CRS robotics and include two object classes of the ActiveRobot interface: one provides the application with the main interface to the robot system, and the other enables the application to create and modify robot locations. The input variables to the control programs are the speed and the position and orientation of the final location of the end-effector. The controller provides the computation of inverse kinematics.

3.3 Control of the Mobile Platform

The locomotive device, or the mobile platform Labmate, has the drive system microprocessor that the user's host computer communicates with through an RS-232 serial port. The host computer always initiates communications between the host computer and the Labmate. The Labmate uses a Cartesian coordinate system for position control. The coordinate system is a global reference that is initialized at power up and reset. Odometry (dead reckoning) is the practice of calculating position from wheel displacements. The Labmate control system depends on encoders mounted on each wheel to keep track of position. Control programs are basically composed of commands that direct the Labmate to a particular location. We assume here that the entire map of the workspace is stored in the form of a look-up table in the memory of the Labmate. If a destination is given, the Labmate computes the direction from the current location by referring to the look-up table. To compensate for the errors, we used two kinds of sensors: ultrasonic ranging sensors and infrared sensors. For more successful navigation of an autonomous vehicle over extended distances, references to the external world at regular intervals are necessary. Figure 2 illustrates the diagram of the library robot controls.

3.4 Control Systems of the Library Robot

We call the main control system of the library robot the "high-level control system". This control system consists of a Pentium II 233 MHz computer notebook and a serial port splitter hub. The computer notebook functions as the central processing unit of the library robot, and it communicates to every subsystem through the serial ports.

3.5 Control Software

The control software is designed based on the idea of event driven programming. Principally, the main control program controls the mobile platform, the vertical translation device, and the arm through serial ports. When the main control program begins to run, it initializes the serial ports of the computer at first and starts the event listeners for all the serial ports. Then the main control program leaves the control to the listeners. It is actually these event listeners that control the movement of the platform, the vertical translation device, and the arm. Basically, each event listener will monitor the status of one serial port. Once the status of that port changes, the listener will judge what kind of event happens and execute a corresponding function. We use the word 'lift' interchangeably with 'the vertical translation device'.

An important property of event driven programming is that the execution order of the functions is not fixed, it depends on the need to execute. This property is suitable for the sensor driven system of the library robot. In total four event listeners are created. They monitor the status of the platform, the lift, the arm, and the sensors respectively. Figure 3 shows the software structure of the robot.

4 Improvement in the Accuracy of the Book Pick-Up Process

To assist the motion planning of the library robot, the complete working process of the library robot was simulated using 3DSMAX. Based on the simulation, a complete path was generated, and experiments were executed to test and adjust the performance of the robot. To simplify the implementation, a map-based scheme was employed to

control the mobile platform. A fixed global coordinate system is defined with its origin at the docking spot. The positions of the intermediate stops and the destination were defined in the global system. An optimal path was chosen to connect the current stop and the next stop. The major problem appearing in the experiments is positioning error. It was found that the positioning error was closely related to the moving speed of the platform. If the speed was low, the motor may lose some steps because of the certain heavy overall load. If the speed was high, the platform may deviate from the desired path at the turning corners because of the inertia. After a few adjustments of the speed setting, the positioning was improved considerably.

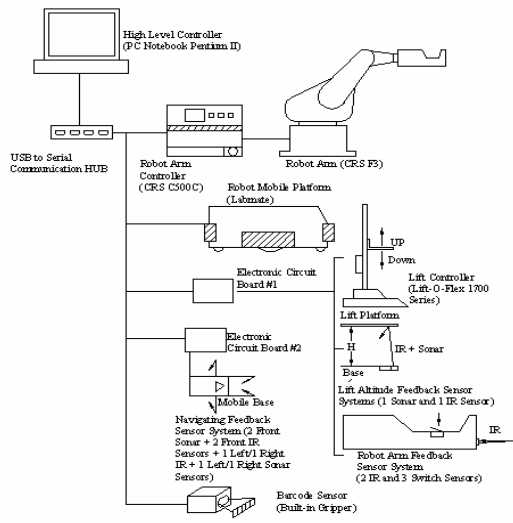


Fig.2. Diagram of the library robot controls.

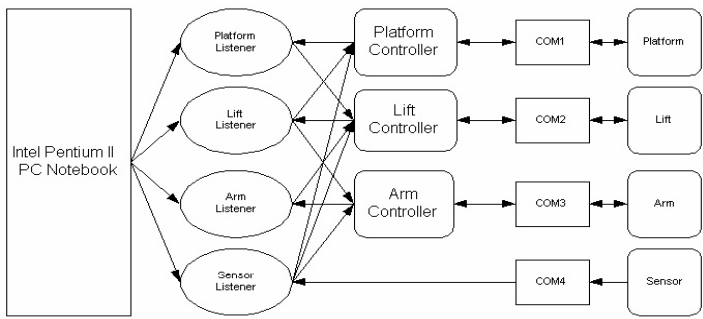


Fig.3. Software structure of the library robot.

To enhance the accuracy of pick-up process in the robot manipulator controls, we integrated an infrared sensing system to the manipulator. The infrared sensing system consists of two infrared sensors (Sharp GP2D02), the sensor controller, and an input/output serial COM port to communicate with the high-level controller. The two infrared sensors were attached to the fingertips of the end-effector, and are controlled by a circuit controller made of a BASIC Stamp II microchip and other required

circuitry components. Figure 4 shows a picture of the IR sensors installed on the gripper. The high-level controller communicates with the infrared sensing system through the I/O serial com port. The high-level controller receives the ranging information from the sensor and converts this information to an updated data file.

In the pick-up process, once the library robot reaches the position in front of the desired bookshelf, the robot (which already has the desired book's coordinates) starts the pick-up process. However, to eliminate the error that may occur by slightly miss-positioning the book, the book-position-scanning process was added.

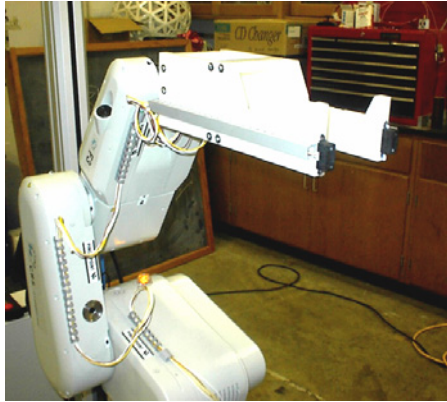


Fig.4. IR sensors installed on the gripper.

Because the manipulator picks up a book in the same manner as a forklift, the positioning of the book is critical. Figure 5 shows the motion planning of the robotic manipulator in the pick-up process. The book-position-scanning process is a process to correct the book-positioning errors. This allows the book to be placed in a specific ranging position instead of only at an exact position. The control architecture of the book-position-scanning process is illustrated in Figure 6.

The book-position-scanning process normally begins with the scanning a certain range of the bookshelf by the IR sensors. The robot uses the known book position stored in its database as the reference position. The manipulator begins scanning from left to right in the horizontal direction relative to the reference position. The scanning path starts at the position located at 40 mm to the left of the reference position, and then moves to the position located at 40 mm to the right of the reference position, stopping at every 5 mm increment. This scanning creates 17 positions along the path. All of these positions are located in the same X-Z plane, which means that only the X-coordinate values vary. At each stop, the median of distance data is calculated and stored in the computer. After the scanning is finished, a set of median distance values is used to find the exact range of the bookcase.

We developed a computer algorithm for the process and obtained remarkable improvement in accuracy from tests. The algorithm was implemented in C++ to control the scanning path of the robot manipulator. The code picks up the data information from infrared sensing system and analyzes these data to generate an actual book location at the end. Once the actual book location is generated, the robot updates the pick-up book position and executes picking-up the book.

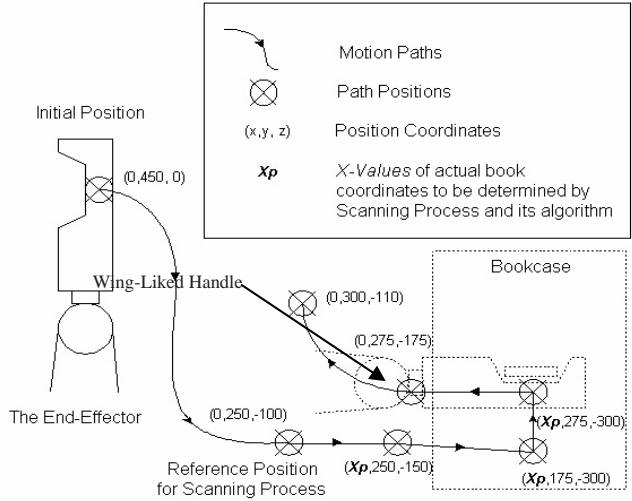


Fig.5. Motion planning of the robotic manipulator in the pick-up process.

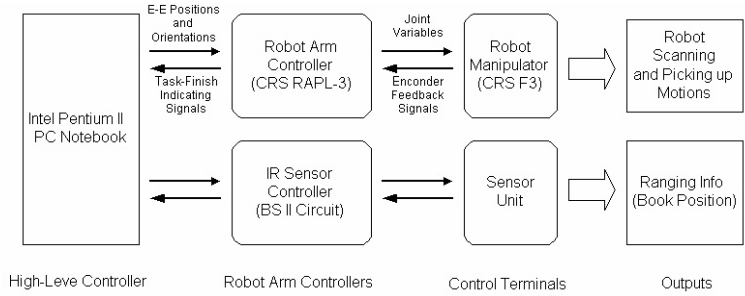


Fig.6. Control architecture of the manipulator's sensing system.

5 Experiment and Results of the New Improving Pick-Up Process

Experiments on the robot manipulator were conducted to determine the accuracy of the pick-up process performed by the manipulator and its feedback control system. The experimental set-up, results, and error analysis are described below.

To verify the accuracy of the pick-up process, we set up the experiment by placing the library robot in the perpendicular direction to the bookshelf (see Figure 7.) The robotic manipulator and its feedback control are designed to be able to scan and pick up a bookcase in a certain range of -40 mm to +40 mm from the reference point, where this range is called “active range.” The active range is located on a bookshelf by marking the far-left, reference, and far-right positions, along with a set of fine measurements in increments of 5 mm.

In each experiment, a bookcase was randomly placed in the active range. We recorded the number of successful and failed trials in attempting to pick up a bookcase at each position. We performed the experiments three times at each position,

without order of positioning. Figure 7 shows the experiment set-up of the robotic manipulator tests, and in the following part the experimental results will be described.

There are no failed trials (out of 51 trials) in this experiment. Figure 8 shows the step-by-step picking up process of the experiment. The robot manipulator started at its initial position and begins scanning from far-left to far-right positions. After the algorithm determines and finds the actual book position, the robot manipulator executed the picking-up process.

6 Conclusion

An autonomous robotic library system was built as a prototype. The robot design, control systems, simulations, experiments and results were presented. An implementation using IR sensors and a new algorithm to enhance the accuracy of an operation process, book pick-up process, was described, and reported its successful experimental results.

While these points outline the specific benefits and qualities of CAPM, it is important to note a ultimate goal of this project. The CAPM Project will introduce robotics into the library and, perhaps more importantly, digital library context. As robotics have provided great impact and utility within manufacturing and, increasingly, computer-assisted surgery, it is possible that similar gains will be achieved in developing digital libraries. Already, some cultural heritage faculty and librarians have identified CAPM as the best option for digitizing the vast amounts of knowledge retained in print format. Additionally, through batch scanning, CAPM will produce automatically and systematically preserved copies of printed materials.

Future work includes the improvement of the robot navigation system, and the development of the page-turning robotic system for the next stage of the CAPM project.



Fig.7. Picture shows the experiment set up for testing the robotic manipulator.

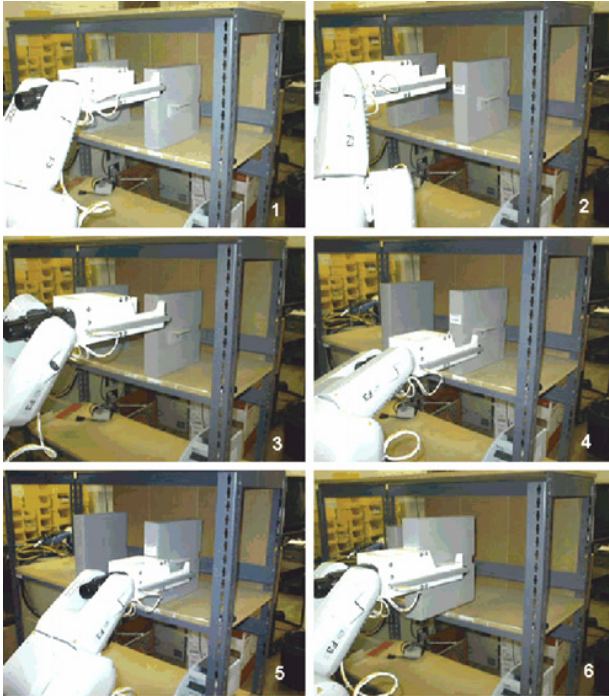


Fig.8. Picture shows step-by-step in pick-up experiment: 1) Manipulator starts at the process's initial position, 2) Manipulator starts the scanning process from the far-left position, 3) Manipulator finishes the scanning process at the far-right position, 4) Manipulator begins picking up the bookcase 5) During the pick-up process, and 6) Manipulator successfully picked up the bookcase.

Reference

1. J. Suthakorn, S. Lee, Y. Zhou, R. Thomas, S. Choudhury, G.S. Chirikjian, "A Robotic Library System for an Off-Site Shelving Facility," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) 2002*, Washington DC, May 2002.
2. R. Hansson, "Industrial robot lends a hand in a Swedish library", *ABB Review*, No. 3, pp.16-18, 1995.
3. R. Safaric, K. Jezernik, D.W. Calkin, and R.M. Parkin, "Telerobot control via Internet," *Proceedings of the IEEE International Symposium on Industrial Electronics 1999*, Vol. 1, 1999. pp. 298-303.
4. J.S. Byrd, "ARIES: A Mobile Robot Inspector," *Proc. of ANS 6th Topical on Robotics & Remote Systems*, Monterey, CA, Feb. 1995.
5. <http://library.csun.edu/>
6. J. Suthakorn, "Paradigms for Service Robotics," *PhD Dissertation*, Department of Mechanical Engineering, The Johns Hopkins University, April 2003.

International Contest for Cleaning Robots: Fun Event or a First Step towards Benchmarking Service Robots

Erwin Prassler¹, Martin Hägele², and Roland Siegwart³

¹ Gesellschaft für Produktionssysteme
Nobelstr. 12, 70569 Stuttgart, Germany
prassler gps-stuttgart.de

² Fraunhofer Institute for Manufacturing Engineering and Automation
Nobelstr. 12, 70569 Stuttgart, Germany
haegele ipa.fhg.de

³ Ecole Polytechnique Federale de Lausanne, Autonomous Systems Lab
CH-1015 Lausanne, Switzerland
roland.siegwart epfl.ch

Abstract. In this paper we report on the First International Contest for Cleaning Robots, which took place jointly with IROS 2002 in Lausanne, Switzerland. The event had two primary objectives. As an educational event with a significant fun factor it was supposed to attract the brainpower and activate the creativeness of students and young researchers for an application of service robotics, which has a significant economic potential. The cost for commercial cleaning services is estimated at around US\$ 50 billion per year only in Europe. A fair contest, of course, required that all contestants had equal race conditions, This in turn required to have a well-define set up, which could be reproduced for every contest team and for any single run. With that, the second major objective of the event, which was to define a benchmark for robotic cleaning, was a natural byproduct of organizing a fair contest.

1 Introduction

Are the times, when one had to spend tedious hours cleaning the kitchen or the bathroom or the mess in the children's bedroom finally over? It sounds like a dream, but there seems to be some hope at the horizon. In the beginning of October 2002, teams of students and young researchers from all over the world gathered in Lausanne in Switzerland at the beautiful lake of Geneva to compete in the first world championship for cleaning robots. The championship was held at the Ecole Polytechnique Federale de Lausanne (EPFL) jointly with the 2002 International Conference on Intelligence Robots and Systems IROS 2002 one of the largest annual robotics conferences world wide. In the beginning it was intended by the organizers as a trial to attract the brain power of students and young researcher to an application of robotics technology which seemed to be far less funny and entertaining than soccer playing robots but has a significant economic potential: cleaning robots. The market for cleaning services in Europe alone is estimated to be US\$ 50 billion per year.

The organizers were everything but sure that what they called the "Robo-cup effect" could be transferred to an application such as cleaning and that a cleaning robot contest could attract only roughly as many students as a contest of soccer

playing robots. Their worries were unjustified. The contest became a big success. Fifteen teams from 10 countries worldwide participated in the contest. The event received overwhelming media-coverage and attracted not only public and academic audience but also a significant number of industrial representatives.

The contest, however, was not only a remarkable event from a technology marketing point of view. There was a second very important aspect tied with the event. The set-up of the contest held in Lausanne was a first step towards a benchmark for service robots. At large the task and the conditions under which it had to be solved were clearly defined by an independent entity in such a way that it can be fully reproduced at any time by anybody. In Lausanne the final evaluation of the performance of the competing system for various reasons was still left to the subjective assessment of jurors introducing some uncertainty in the performance measurement. However, it will be only a cost issue, to replace the human jurors by a technical measuring system such guaranteeing a performance measure and comparison, which is free of any subjective assessment.

Of course, it may not be straightforward to transfer the set-up of contest in Lausanne to an application other than cleaning. At the same time is evident, however, that it will be extremely difficult for any service robot, let it be a cleaning robot or any other robot assisting the human, to mature to a state ready for production and to find its way into the market without clearly defined performance measures and figures. Having a Formula 1 in service robotics does not only mirror the state of the available technology. It may also significantly reduce the duration of innovation cycles and the time to market of this new generation of robots. Benchmarks will not only push the engineering part of the development of those systems. They will also push the development of robust and sound algorithms. A lesson, which could be learnt from the contest in Lausanne, was that even seemingly trivial everyday task always leaves enough room not only for ad-hoc heuristics but also for sound algorithmic solutions.

In the following section we will first describe the set-up of the first international championship for cleaning robots. This includes details such as the set-up of the contest area, the contest rules or the measurement and scoring procedure. This section is followed by an overview of the teams which competed in the contest together with an short overview of their systems and the employed cleaning strategy. In Section 3, we further present the results of the contest including the qualification and the semi-finals. In the final section we subject the event in Lausanne a critical review. We will discuss some problems, which could hardly be avoided in such a first event but may be circumvented in the future. These problems are of particular interest if the contest is to be further developed towards a true benchmark for cleaning robots.

2 The Set-Up of the Contest

The contest was arranged in of three sub-contest:

- a contest robotic floor cleaning in a “regular” living room
- a contest robotic window cleaning
- an idea contest “household robot”.

Due to the lack of space we will focus in this paper on the floor cleaning sub-contest. This was the event which attracted the most attention amongst the teams as well as amongst the audience and press.

2.1 Overall Structure of the Contest

The contest for robotic floor cleaning took place in two adjacent identical contest areas each 5×5 square meters in size and furnished as a typical living room with a black linoleum surface floor. The two adjacent contest areas are shown in Fig. 1. Wall-like barriers approx. 80 in high surrounded these areas to prevent the audience from entering and the robots from escaping the contest areas. The model layout of the contest area was published about two weeks before the contest.

The contest was organized very much like a soccer championship. In total, 12 teams from seven countries accepted the challenge and ran for the floor cleaning contest. These 12 competing teams were divided into four groups of equal size, each group consisting of three teams. In the qualifications, the best team was determined in each of the four groups. This was done in a pair wise direct competition. Within the groups each team had to compete with the two other teams, which means that each group had to participate in two runs. This resulted in three runs per group and 12 runs in total in the qualifications.

Two competing teams run their cleaning robots in parallel in the two adjacent contest areas. This schedule enabled the audience and jury to observe the performance of the robots in direct comparison and also significantly contributed to the entertainment value of the contest. A single run was limited to 10 minutes. During this period, the two robots had to operate autonomously in the contest areas and clean an area as large as possible. The robots were placed and switched on upon a signal by a member of the jury.

According to the rules published before the contest, the teams were not permitted to enter the contest area during a run, otherwise they would be disqualified. For pragmatic reasons, this rule could not be sustained. Too many robots got stuck in corners or underneath furniture. So it was decided by the jury and the organizers to allow interventions but to punish each intervention with a loss of 33% of the scores.



Fig. 1. Contest areas for robotic floor cleaning

As mentioned above each team in a group had to pass two runs, in which it directly competed with the two other teams in the group. The scores of these two runs were summed up, yielding the final scores of each team in the qualifications. The team with the highest scores entered the semi-finals. From the semi-finals on the contest was conducted in a double-elimination mode, which means a team had to compete only with one other and not with all other teams. Accordingly the semi-finals consisted only of two runs. The opponents in the semi-finals were chosen by lot. The winners of the two runs entered the finals, while the losing team would compete for the third place.

2.2 Layout and Assembly of the Contest Areas

The two contest areas were laid out as two typical living rooms with a couch, a table, sideboards two stools, and a lamp. Additionally a little doll was placed in the living room, representing a baby which had to be treated very gently during the race. The floor in each area consisted of a total of 100 wooden tiles with a side-length of 0.5×0.5 sqm covered with black linoleum. These tiles were arranged in a rectangle of 5×5 sqm. As mentioned above each contest area was surrounded by a wall-like fence preventing the audience to enter and the robots to leave the contest area.

The living rooms were equipped with the following IKEA furniture (from the summer catalogue 2002):

- couch “Klippan” ($h = 69/41$; $w = 188$; $d = 88$; $cfg = 13$)
- stool “Knoppe” ($h = 39$; $\emptyset 48$; $cfg = 16$)
- table “Krokshult” ($h = 54$; $w = 120$; $d = 68$; $cfg = 27$)
- shelf “Visdalen” ($h = 87$; $w = 78$; $d = 42$; $cfg = 15$)
- sideboard “Visdalen” ($h = 72$; $w = 92$; $d = 48$; $cfg = 9$)
- lamp “Skyar” ($h = 175$; $\emptyset = 25$; $cfg = 0$)

where h , w , d , \emptyset , and cfg stand for height, width, depth, diameter and clearance from ground. The exact layout of the contest areas is shown in Fig. 2. The doll (not shown in the layout) had approximately the size and weight of a baby (50 cm and 4 kg). In the contest it was usually placed next to the table.

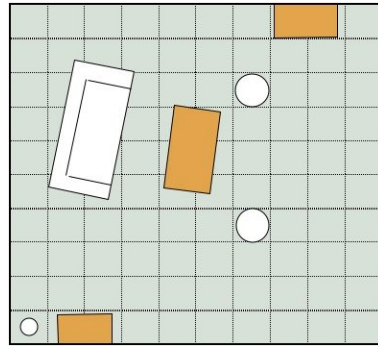


Fig. 2. Contest areas for robotic floor cleaning

2.3 Pollution of the Contest Area

In order to assure equal race conditions for each team, it was necessary to cover the contest area with dirt in an even and reproducible fashion. This was everything but a

trivial task. As dirt white sugar was chosen for several reasons. Sugar can be removed by vacuuming, sweeping, and wiping e.g. by a wet towel. In the contest this was a good compromise, since not all robots employed the same cleaning technology. For a benchmark, which aims at a specific type of cleaning technology, some other type of dirt might be more appropriate. Another reason for choosing sugar was the fact that it was well visible on the black linoleum floor. This was important for measuring the “cleaning performance”. The distribution of the dirt in an even and reproducible fashion proved to be a challenging problem. After a number of devices and approaches had failed miserably, a tool for lawn fertilizing was finally used for distributing the dirt. Although this device still left some room for improvement it showed by far the best results.

2.4 Hardware Requirements and Auxiliary Devices

The cleaning robots had to obey a few hardware requirements. The weight of a robot must not exceed 30kg and its size must not exceed 60cm in diameter or side-length and 50 cm in height. The cleaning robots did not necessarily have to be completely self-designed. The use of commercially available experimental robots was permitted as well as the use of commercially available cleaning technology, e.g. battery powered small vacuum cleaners, brushes or alike. Cleaning robots built from off-the-shelf components were scored differently, however (see Section “Handicaps”).

The following auxiliary devices were allowed:

- docking stations and
- external movable and removable reference stations and
- navigation support systems.

Non-movable installations were not allowed.

2.5 Rules

Besides the above requirements the teams and robots had to obey a number of rules to assure a fair contest:

- one run was limited to 10 minutes,
- during these 10 minutes the robots had to clean as much space in the contest area as possible,
- teams could place their robots at any place in the contest area
- one team member had to turn on the robot and then leave contest area immediately
- upon a signal from a juror, one team member had to enter the contest area and turn off the robot after the run
- interference with the robot during the race was considered as a foul (see Section “Fouls”)
- pushing the doll away from its position was considered as a foul (see Section “Fouls”)

2.6 Performance Measurement

In order not to kill the entertainment value of the event a trade-off had to be found for measuring the cleaning performance of the robots during the contest. A fair and sound measure of the cleaning performance of each robot would have required a significant amount of time. During the contest this time was not available. To still assure a minimum level of correctness the following measurement procedure was employed:

- due to its construction the floor in the contest area was divided into 100 tiles of equal size,
- each tile counted as “cleaned” when it was covered by the robot to more than 50% during a run,
- a “cleaned” tile counted as one point,
- the number of “cleaned” tiles were counted after the run by a group of two independent jurors.

These points were set against the fouls committed during the race and against a handicap factor, which was assigned to each robot to yield the final scores.

2.7 Fouls

During a run, the robots were allowed to touch the objects in the contest area, but not to damage them in any kind. The following actions by the robots or the teams were considered as fouls and penalized accordingly:

- pushing the doll away from its place
- damaging (e.g. scratching) a piece of furniture
- interference of a team member with the robot during the race

Each single foul lead to a loss of 33% of the achieved scores.

2.8 Handicaps

It was in the spirit of this first contest to allow a maximum level of creativity in the design of the robots. According to the contest rules self- designed robots with self-designed cleaning technology were admissible as much as off-the-shelve experimental platforms equipped with some off-the-shelve cleaning mechanism. High-tech platforms with expensive sensor equipment was accepted as much as low-cost designs with simple and cheap sensor technology. For a fair evaluation of such a variety of designs it was either necessary to introduce classes of similar designs and compare only the performance of robots in the same class or to introduce some sort of handicap factor. Since it was foreseeable that the number of robots participating in the contest was not sufficient to establish classes of similar designs, it was decided to introduce a handicap factor to make the performance of the robots comparable. The handicaps factors are shown in the table below.

These handicap factors were used as multipliers to calculate the final scores. So, a self-designed robot with no range sensing using some self-designed cleaning device that cleaned 20 tiles, would have achieved a score of 33.8. A commercial robot platform with some commercial cleaning device using a high- cost range sensor such as a laser range finder would have had to clean 56 tiles to achieve the same score.

Table 1. Handicap factors to account for design variations

<i>Design</i>	<i>Handicap</i>
self-designed robots	1.3
self-designed cleaning technology	1.3
commercial robot platforms	1.0
commercial cleaning technology	1.0
low cost range sensors (sonar, IR)	0.8
low cost external positioning aids	0.8
high cost range sensors	0.6
high cost external positioning aids	0.6

3 The Teams, the Robots, and the Results

3.1 The Teams and the Robots

While preparing the event one of the biggest worries of the organizers was whether they would be able to attract a sufficient number of teams. Luckily these worries were completely unnecessary. In total 15 teams from 10 countries registered for the three sub-contests, 12 for the floor cleaning contest, 3 for the window cleaning contest and 2 for the idea contest. Two teams actually registered for two out of three contests. It is not overstated to say that the designs of the robots participating in Lausanne were spectacular without exceptions. More than 80% of the robots were self-designed.

In the following we give a very brief overview of the robot designs participating in the floor cleaning contest:

Carnegie Mellon University, USA

commercial platform Nomadic Scout with differential drive and ring of Polaroid US sensors, broom mounted in front of robot, optimal coverage algorithm for unknown environments (boustrophedon decomposition)

Charles Univ. Prague, Czech Rep.

self-design, differential drive consisting of rubber tracks, range sensing using color CMOS camera and structured light, position sensing by odometry, coverage through random motion, obstacle avoidance based on a local map created from range data.

EPFL, Autonomous Systems Lab, Switzerland

self-design platform using two SICK 2D laser range finder, differential drive mechanism, micro-fibre cleaning towel mounted at the bottom of vehicle, optimal coverage by locomotion on parallel tracks in a partially mapped environment.

FAW Ulm, Germany

self-design robot with differential drive and tactile sensing, self-designed wet-cleaning unit, position estimation based on odometry, coverage through a combination of random motion and fixed motion pattern.

Fraunhofer Institute AIS, Germany

modified commercial platform KURT2 with differential six-wheeled drive system,

10 ultrasonic range sensors, commercial electric broom, coverage by locomotion on parallel tracks and random motion.

Short Circuits Robotics Club, Ireland/Czech Rep.

self-designed platform with differential drive, self-designed vacuum cleaning unit, collision detection by measuring motor current, additional tactile sensing, position estimation through odometry, coverage based on random motion.

Univ. of Guelph, Canada: modified commercial platform from Evolution Robotics with a commercial broom attached to the vehicle, differential drive, vision system for obstacle detection and avoidance, position sensing by matching long straight edges while moving, coverage based on random motion.

University of Kaiserslautern, Germany

self-designed platform with self-designed vacuum cleaning unit, differential drive system, infrared range sensing and tactile sensing, position estimation through external laser positioning system, coverage through motion on parallel tracks.

University of Lübeck, Germany

self-designed differentially driven platform with a self-designed vacuum cleaning unit (1000W), equipped with 15 infrared range sensors, 3 ultrasonic sensors and 20 micro-switches, makes a U-turn once an obstacle is detected, coverage by moving on parallel tracks.

Universidad de Salamanca, Spain self-designed robot platform with an electric broom attached in front of vehicle, differential drive, 8 Polaroid range sensors, 5 infrared sensors, position estimation through shaft-encoders, coverage by contour-following and locomotion on a spiral-shaped track.

Azaiz Ridha (student), Germany

self-designed differentially driven vehicle, high resolution range sensor allows differentiating between carpet, vase and other objects.

Hertwig Andre (private inventor), Germany

self-designed vehicle with differential drive, self-designed brushing and vacuuming unit, reflecting light sensor of collision detection, coverage by combination of random motion and locomotion along a meander.

3.2 The Results

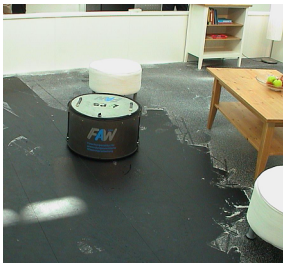
As described above the floor cleaning contest took place in three stages: the qualifications, the semi-finals and the finals. The results of these three stages are summarized in the tables below. It should be mentioned that the team from Carnegie Mellon University due to technical problems could get its robot started and therefore could not enter the contest. In Fig. 3, pictures of the best four floor cleaning robots are shown.



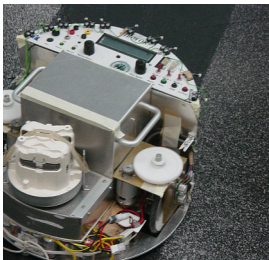
Short Cicut Robotics Club (winner)



Andre Hertwig (second place)



FAW Ulm (third place)



Univ. of Lübeck (fourth place)

Fig. 3. The top four

Table 2. Results of qualifications in floor cleaning

Group	Team	1. Run	2. Run	TOTAL	Raking
A	Short Circuit Robotics	46	45	91	1.
	Fraunhofer AIS	16	15	31	2.
	Univ. of Salamanca	4	17	21	3.
B	FAW Ulm	42	35	78	1.
	Univ. of Kaiserslautern	27	12	39	2.
	Charles Univ. Prague	16	13	29	3.
C	Univ. of Luebeck	57	48	104	1.
	EPFL	7	22	29	2.
	Univ. of Guelph	3	1	4	3.
D	Andre Hertwig	39	51	91	1.
	Ridha Azaiz	8	16	24	2.
	Carnegie Mellon Univ.	0	0	0	3.

Table 3. Results of semi-finals in floor cleaning

Group	Team	1. Run	Raking
A	Short Circuit Robotics	51	1.
	FAW Ulm	29	2.
B	Andre Hertwig	43	1.
	Univ. of Luebeck	15	2.

Table 4. Results of finals in floor cleaning

Ranking	Team	Scores
1	Short Circuit Robotics	51
2	Andre Hertwig	29
3	FAW Ulm	43
4	Univ. of Luebeck	14

4 Insights and Conclusions

The contest in Lausanne was the first of this kind. Obviously, many decisions had to be made spontaneously and on a gut level. There were certainly some elements in the contest, which leave plenty of room for improvements, some of them were mentioned already. This is particularly true, if the contest is developed further towards a true benchmark.

First, in the contest two aspects, which may actually be separated, were treated as a single one: cleaning performance and coverage. More precisely, coverage was implicitly treated as an element of cleaning performance. This does not always make sense, particularly, if we are not really interested in the suction performance of a vacuum cleaning unit which is powered by a battery. So, in a benchmark these two aspects should definitely separated and measured separately.

Although it was desired, the large design variety of the robots caused a crucial problem, namely that of the comparability of the systems. In Lausanne the organizers worked around this problem by introducing handicap factors. But it is obvious that this is not the best solution. The organizers of future events should think about organizing the contest in performance categories or leagues. This seems to be a fairer and sounder approach, but requires a sufficient number of participants per league.

Two further aspects, which left room for improvement in Lausanne, were the measurement and evaluation procedure and the method for distributing the dirt. For the first aspect, the performance measurement, an automatic measuring system would be desirable. Such a system should be able to precisely measure the area covered by the cleaning robot in a certain time interval, and, if desired, the amount of dirt removed in that time. Also the second aspect, the reproducible distribution of dirt, would ideally be left to some automatic device. However, it is obvious, that these issues are not only a matter of scientific insights and pioneering spirit but also of available finances.

Acknowledgement

This work was partially supported by the German Ministry for Education and Research (BMB+F) under grant no. 01 IL 902 F6 and was conducted in the context of the MORPHA project as an activity towards benchmarking service robots.

Dragline Automation: Experimental Evaluation through Productivity Trial

Peter Corke, Graeme Winstanley, Matthew Dunbabin, and Jonathan Roberts

CSIRO Manufacturing and Infrastructure Technology

firstname.lastname@csiro.au

www.mining-robotics.com

Abstract. Since 1993 we have been working on the automation of dragline excavators, the largest earthmoving machines that exist. Recently we completed a large-scale experimental program where the automation system was used for production purposes over a two week period and moved over 200,000 tonnes of overburden. This is a landmark achievement in the history of automated excavation. In this paper we briefly describe the robotic system and how it works cooperatively with the machine operator. We then describe our methodology for gauging machine performance, analyze results from the production trial and comment on the effectiveness of the system that we have created.

1 Introduction

Automated excavation is a potentially important robotic application and various research groups have investigated control of different types of machines such as excavators[1,2], bulldozers and draglines[3]. An excellent review of the field is given by Singh[4].

Draglines, see Figure 1, are large electrically powered machines used to remove the blasted rock (overburden) to uncover coal seams in open-cut mines. They are large (weight 3000-4000 tonnes) and expensive (\$50-100M) and generally in production 24 hours per day, every day of the year. The machine fills its bucket by dragging it through the ground; lifts and swings it to the spoil pile, dumps it, and then returns. Each cycle moves up to 100 tonnes of overburden, and typically takes 60 seconds. The work is cyclic and very repetitive, yet requires considerable operator skill. The *only* performance metric is tonnes of overburden moved per hour (without breaking the machine).

We commenced this project with Phase 1 (1993-1995) which demonstrated control of a 1/10th scale model dragline[5] using computer vision to measure load position. This was followed by Phase 2 (1996-1998) which commissioned the control system on a full-scale production dragline and demonstrated all aspects of operator skill[3,6]. We have just completed Phase 3 (2000-2002) on a different full-scale machine and performed a two week production test where we ran over 3,000 automated cycles and moved nearly 200,000 tonnes of dirt. To the best of our knowledge this is the first such comprehensive test of an automated excavator in a working mine environment and exceeds by an order of magnitude any earlier work in the field.



Fig. 1. The Bucyrus-Erie 1350 dragline at Boundary Hill mine, Queensland, which was used for our testing. The boom is 100m long.

Section 2 describes briefly the salient points of the automation system. Section 3 describes the production trial, the analysis methodology and results. We draw conclusions in Section 4.

2 The Automation System

From a robotics point of view the machine can be considered as a 3DOF robot (drag rope, hoist rope, slew angle) with a flexible, or un-actuated, final link, and the task can be thought of as comprising:

constrained motion (digging, interaction between the bucket and ground),
unconstrained position control (moving the bucket in free space). In our project we chose to automate the second task since it is simpler and also comprises the bulk (80%) of the cycle time, and control is *traded* between operator and computer. Our automation system does in fact perform some simple constrained motion, since it is activated while the bucket is full and engaged in the bank.

Technical details of our system are given in [7,8,3] while [6] describes how our thinking changed over the duration of the project as we became aware of other, generally non-technological, issues.

In normal manual operation the human operator visually senses the state of the bucket, its rigging and the local terrain and actuates the drag, hoist (joystick) and slew

(pedal) controls in a coordinated fashion. A good deal of skill is required to control, and exploit, the bucket's natural tendency to swing. This skill is not easily learnt and a novice operator typically requires 6 months training to become proficient. Performance can vary by up to 20% across the operator population, and for an individual will vary over a shift (up to 12 hours). It has been estimated that a 1% improvement in machine productivity is valued at around AUD \$1M per machine per year.

Our dragline swing automation (DSA) system is activated by the operator once the bucket is filled. It automatically hoists, swings and dumps the bucket (at an operator defined reference point) before returning the bucket to a predefined digging point. Training of dig and dump points is done during a normal working cycle using the touch screen to enter training mode, and the joystick mounted button to mark points. The automation system then computes a path to move the bucket from dig point to dump point. The operator may also specify intermediate, or via, points so that the bucket path can will avoid obstacles such as the edge of the pit or a spoil pile.

Critical technology elements include:

Load position sensing provides instantaneous bucket position and velocity, under all visibility conditions, day and night. The system uses a PLS scanner mounted in a weatherproof enclosure mounted near the boom tip. Robust target tracking and data association is used to detect each of the two hoist ropes in the scanning plane, the centroid is computed and from this the rope angle. A Kalman filter provides smooth angle and angular rate estimates, as well as covariance which we use a means to detect tracking failure.

Operator interface The automation system 'drives' the dragline by physically moving the control joysticks and pedals; like the cruise control in a car or an aircraft autopilot. To achieve this we modified the control sticks and pedals by adding brushless servo motors and an electromagnetic clutch. A button, mounted in the ball of the stick, is used to engage the automation system and to mark training points. These 'active' controls facilitate the smooth transfer of drive motor set points in the transitions between automatic and manual control modes. Sensing in servo loops allows the automation system to sense if the operator is opposing the motion of the joysticks — in which case it smoothly transfers control back to the operator.

Touch screen display provides a simple display of status and allows the operator to perform tasks such as adjusting digging and dumping points, or enable automatic bucket return point indexing.

The control system is hierarchical, and in top down order comprises;

1. Three coupled state machines, one per axis, which sequence through the states: DISENGAGE, TODUMP, DUMPING, DUMP, RECOVER, TODIG, APPROACH.
2. The motion planner attempts to move all machine axes so as to reach the goal state by a mutually *agreed* deadline which is dynamically adjusted. This minimum time strategy requires that one axis is always kept at its performance limit.

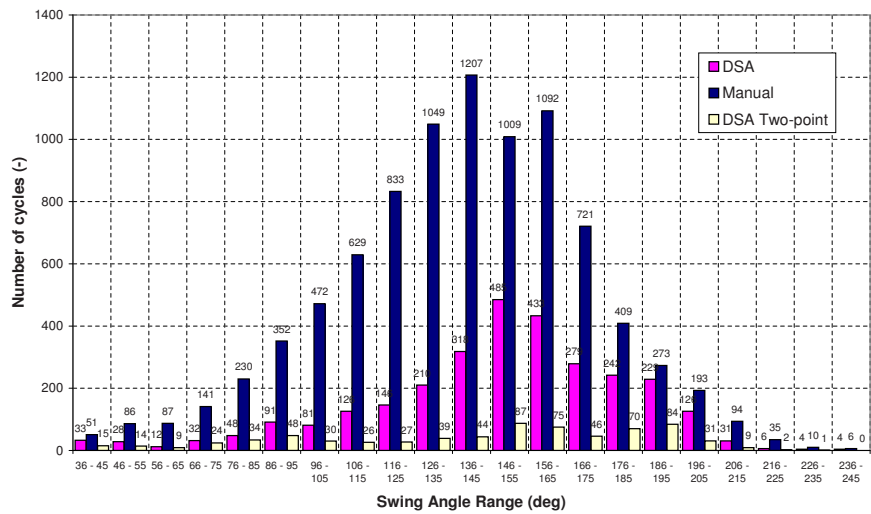


Fig. 2. Swing angle histogram showing the number of cycles performed manually, by DSA and by DSA with 2-point paths (see Section 3.3).

3. The control layer takes the setpoints from the planner (desired rope length and slew angle), sensed machine state and computes the control demands to the dragline. This is essentially a position control loop, though some force control is also implemented. Control loop parameters and modes are a function of the main state machine.
4. The axis demand is communicated to embedded microprocessor controllers which implement the servo loop that actuates each active control.
5. The joysticks and pedals are mechanically connected to rheostats which control current to the amplidyne based Ward-Leonard regulator that powers the main traction motors.

3 The Production Trial

The performance measure of interest to the client is cubic metres moved per hour. Given that the operator fills the bucket consistently in automatic and manual mode, then the cycle time statistics of manual versus automatic operation are of great interest. A total of 12,235 cycles were recorded during the trial: 3,042 with DSA and 9,193 manual cycles. Our target was in fact 10,000 automated cycles but maintenance and operational issues precluded this. Nevertheless this number of automated cycles far exceeds, to our knowledge, any previous operational testing of a robotic excavation machine.

3.1 Methodology

Since we cannot dig the same block of dirt manually and automatically we ran the automation system for one hour on, one hour off for two shifts per day over the two week period. We relied on the digging conditions being uniform, ie. the same type of cycle for the entire period, in order to allow meaningful comparison but this turned out not to be the case.

Cycle time for almost all Australian draglines is continually measured by a Tritronics performance monitor. This is an industry standard device whose results are believed. We downloaded all cycle data for the two week period and separated manual and automatic cycles. Next we applied an industry standard methodology which is based on dividing the cycles according to the swing angle range, grouping them into 'bins' that are 10 degrees wide. A histogram of swing angle range for manual and computer cycles achieved over the trial is shown in Figure 2. Although the technique is 'standard' within the industry it has at least two limitations:

- Narrow bins will have too few, or zero, cycles to be useful. Wide bins will have a large distribution of cycle times due to the maximum swing rate. For example, a 10deg bin will have a spread of at least 1.5s if the maximum swing rate is 6deg/s.
- The cycles are binned purely on the basis of the swing angle, and may have occurred at widely separated times over the two weeks of the trial. This may also correspond to widely different digging conditions, ie. digging and dumping depth.

3.2 Analysis

Figure 3 shows a histogram of the cycle times for all cycles whose swing angle falls within a particular binned range. The calculated mean and peak cycle times for the binned data are indicated by arrows. It can be seen that for a particular swing angle there is a best possible time (due to motor performance limits). However, we notice that in practice, both the operator and DSA have some considerable scatter in the cycle times actually achieved. For the manual cycles this may reflect different operators, normal operator variability or different digging conditions (perhaps hoist drive limited). For DSA we also notice scatter and in particular a 'long tail' on the distribution.

This 'long tail' reflects the 'tune-in' process, whereby for each time DSA is started we need to adjust some parameters in order to optimize cycle time, as well as to finesse the dump and recovery performance. From this it can be seen that some DSA cycles are competitive and some are not. Ideally we would like to express the performance within the bin using some compact statistic. A mean of the times would penalize the DSA results because of the long times recorded in the tune-in process. The mode statistic reports the time corresponding to the peak, that is, the most common time for this swing angle range. However, computing the mode is difficult, particularly for some bins where there are very few DSA cycles.

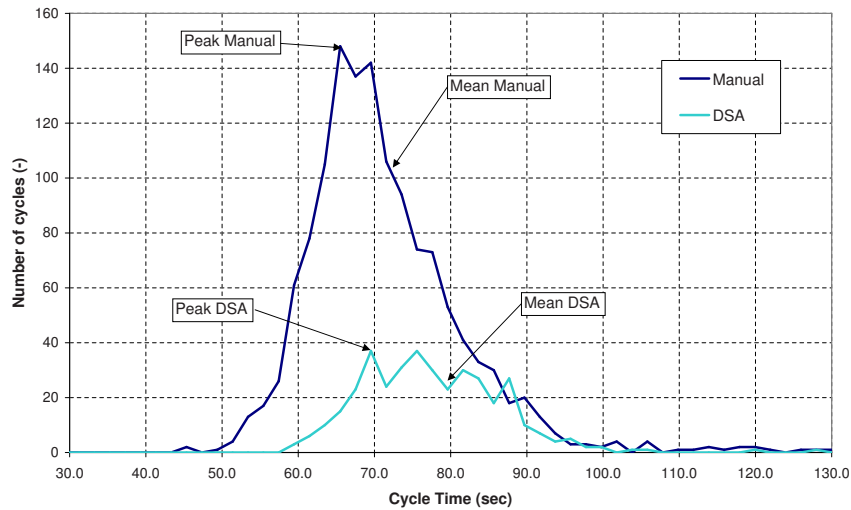


Fig. 3. Cycle time distribution within the bin 136-145deg.

In Figure 4 we plot the mean and mode cycle time for each swing angle range. The mean times show that the automation system “wins” for small and large swing angles, but in between there is an almost constant time penalty in the range 5 – 10s. The peak times paint a somewhat different story, so the choice of statistic used is clearly critical.

A perhaps better statistic is to compute the percentage of DSA cycles that are completed in the same or less time than the mean manual time. This statistic does not involve trying to estimate the peak of a distribution, which is a difficult task for a small number of samples where the distribution is far from normal as shown by the histogram of Figure 3. The calculated percentage of DSA cycles which are completed quicker or as quick as the mean manual cycle time for all swing angles is shown in Figure 5(top).

3.3 Hoist Limiting

Early in the analysis we became aware that the performance monitor flagged an unusually high number of the automated cycles as “hoist limited”, but not those for an operator digging and dumping at the same points. We isolated this effect to those automated cycles which involved a multi-point path from dig to dump. Simple paths, without a via point, are not marked as hoist limited. The exact problem with multi-point paths is not yet known, the segments are each timewise optimal and the transition is fast and smooth. The problem may be due to the placement of the intermediate points. Generally for multi-point paths, the first via point is set near the edge of the bench. The DSA system will therefore be hoist limited for the first segment, and then swing limited for the remaining segments. Perhaps the long initial

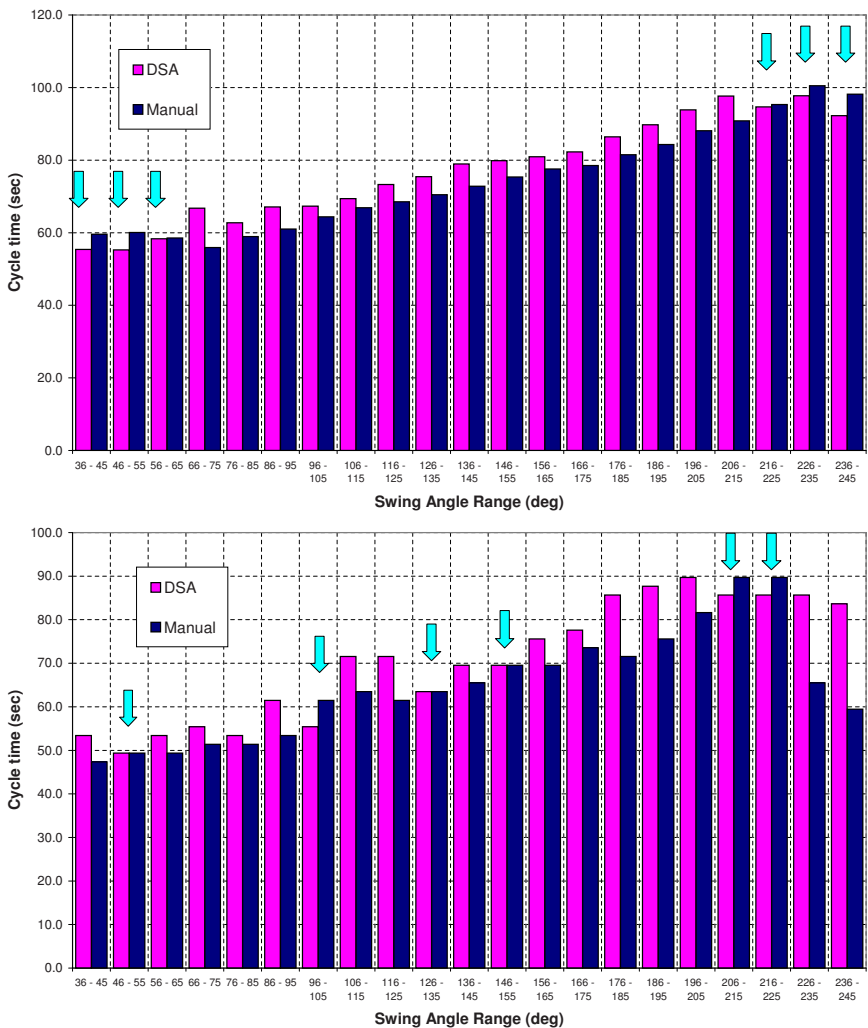


Fig. 4. Comparison of mean (top) and mode (bottom) cycle times versus swing angle. Arrows indicated bins for which DSA performance exceeds operator.

period of hoist limited motion is causing the Tritonics to categorize the entire cycle as being hoist limited.

Figure 2 shows the distribution of two-point DSA cycle swing angles compared to all DSA and manual cycles — there are significantly fewer of such cycles. The calculated percentage of two-point DSA cycles which are completed quicker or as quick as the mean manual cycle time for all swing angles is shown in Figure 5. We see that the performance of the automated system is now considerably better than the operator for the middle swing angle range where most of the cycles occurred.

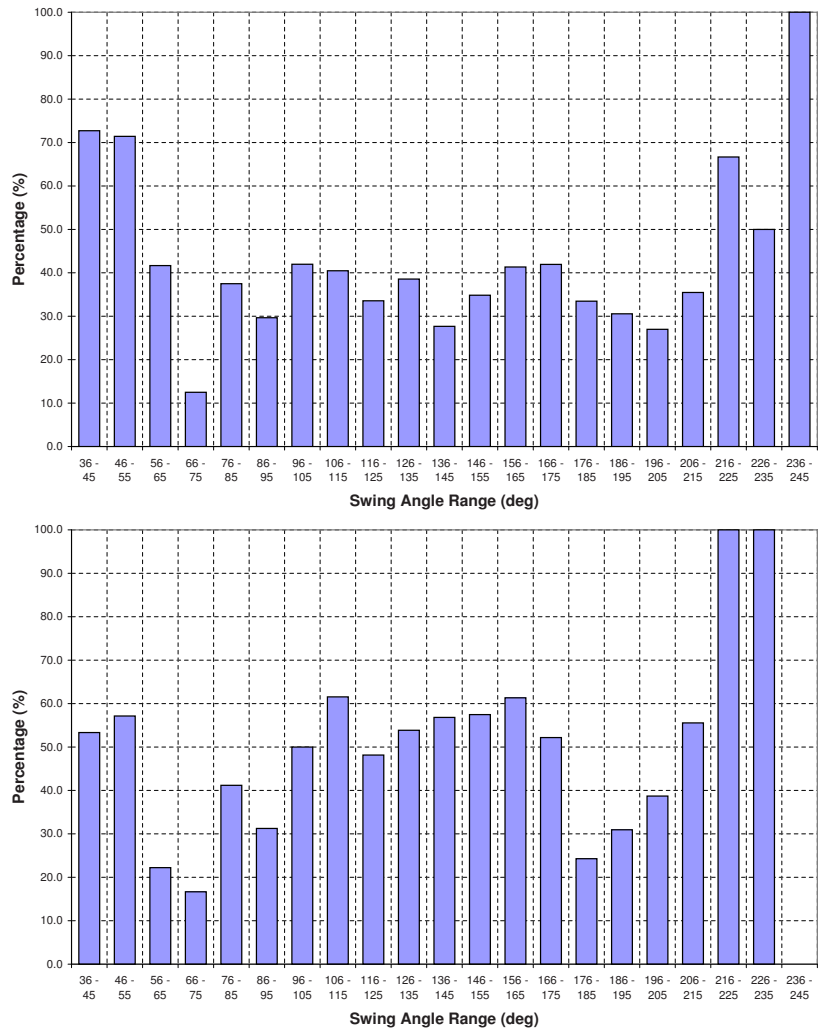


Fig. 5. Histogram of percentage of DSA cycles better than mean operator cycle times. Top, all cycles; bottom, two-point cycles.

3.4 Uniform Digging Analysis

As mentioned above, a limitation of the methodology is that each swing angle bin contains cycles that may correspond to quite different operating conditions such as dig depth and dump height. To obtain uniformity we choose to analyze cycles from a chronological block on 12th November 2002: 1 hour manual, 1 hour DSA, then 1 hour manual. Figure 6 shows the distribution of mean cycle times versus swing angle for this small dataset. It can be seen that the mean DSA times are in general better than the operator times.

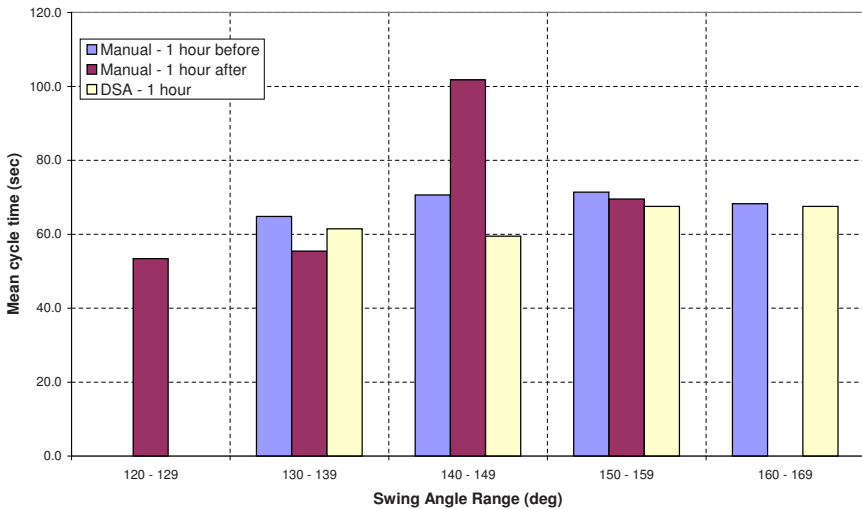


Fig. 6. Adjacent digging block, comparison of mean cycle times.

These results show that when the DSA system is “tuned” correctly, and is not hindered by excessive conservatism in hoisting, the system is as good, if not better than manual operation of the dragline. Again we must caution that the number of automated cycles in this analysis, around 50, is very low.

4 Conclusions

We have twice now demonstrated the feasibility of automating a full-scale production dragline, and developed a computer control system that has all the essential ‘skills’ of a dragline operator: disengaging from the bank, dumping and bucket recovery are performed consistently well. The computer has thus mastered what is considered a very difficult skill, albeit at the level of a novice of perhaps 6 months training. Also significant was that the system was highly reliable, there were no hardware or software failures over the duration of the trial, and that operator interface was intuitive and readily accepted by the operators. With further research the skills can only improve.

Our production trial is unprecedented in the history of robotic excavation and we moved nearly 200,000 thousand tonnes of material. Analysis of the results is not straightforward and is dependent on the statistical methodology chosen. The analysis showed a defect in the motion planner for a particular class of paths, a defect that was not apparent to any of the people onboard the dragline during the trial.

If we discount the multi-point cycles, or take data from uniform and consistent digging, the DSA equals or outperforms the operator. This is despite the fact that throughout the trial, for machine safety, we chose conservative bucket paths to avoid

bucket/spoil collisions. The resultant over-hoisting, average 6m, imposes a clear time penalty on the DSA system (at 3m/s hoisting speed this corresponds to 2s).

A significant remaining problem is that the automation system lacks knowledge of its spatial environment, making it incapable of planning its own paths. In practice the human operator takes the bucket *very close* to the spoil piles, whereas the automation system must plan conservative paths that have considerable clearance from potential objects. For the DSA system to achieve this it would need to be integrated with a real-time terrain mapping system such as the one we have developed separately[9]. Further time loss is attributable to delay in the operator handing control to the DSA system after the bucket is filled, and the very gentle bucket disengage at the start of each cycle.

Acknowledgement

The authors gratefully acknowledge the help of their colleagues Lesley Overs, Stephen Brosnan, Pavan Sikka, Craig Worthington and Stuart Wolfe. The work described was funded by the Australian Coal Association Research Programme as project C9028. The authors appreciate the support of Callide Coalfields for allowing the automated swing system to be installed on their BE 1350.

References

1. S.Singh, A. Stentz, J. Bares, and P. Rowe, "A robotic excavator for autonomous truck loading," *Autonomous Robots*, vol. 7, pp. 175–186, Sept. 1999.
2. P. Lawrence, S. Salcudean, N. Sepehri, D. Chan, S. Bachmann, N. Parker, M. Zhu, and R. Frenette, "Coordinated and force-feedback control of hydraulic excavators," in *Proc. ISER*, July 1995.
3. J. Roberts, P. Corke, and G. Winstanley, "Development of a 3,500 tonne field robot," *The International Journal of Robotics Research*, vol. 18, pp. 739–752, July 1999.
4. S. Singh, "The state of the art in automation of earthmoving," *ASCE Journal of Aerospace Engineering*, vol. 10, Oct. 1997.
5. D. W. Hainsworth, P. I. Corke, and G. J. Winstanley, "Location of a dragline bucket in space using machine vision techniques," in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP-94)*, vol. 6, (Adelaide), pp. 161–164, April 1994.
6. P. I. Corke, J. M. Roberts, and G. J. Winstanley, "Experiments and experiences in developing a mining robot system," in *Experimental Robotics VI* (P. Corke and J. Trevelyan, eds.), no. 232 in Lecture Notes in Control and Information Sciences, pp. 183–192, Sydney: Springer Verlag, June 2000.
7. P. I. Corke, J. M. Roberts, and G. J. Winstanley, "Modelling and control of a 3500 tonne mining machine," in *Experimental Robotics V* (A. Casals and A. de Almeida, eds.), no. 232 in Lecture Notes in Control and Information Sciences, pp. 262–274, Barcelona: Springer Verlag, June 1998.
8. J. Roberts, F. Pennerath, P. Corke, and G. Winstanley, "Robust sensing for a 3,500 tonne field robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, (Detroit), pp. 2723–2728, May 1999.
9. J. Roberts, G. Winstanley, and P. Corke, "3d imaging for a very large excavator," *Int. J. Robot. Res.*, vol. 22, pp. 467–478, July 2003.

Shearer Guidance: A Major Advance in Longwall Mining

David C. Reid, David W. Hainsworth, Jonathon C. Ralston, and Ronald J. McPhee

Mining Automation
CSIRO Exploration and Mining
Technology Court, Pullenvale, Q 4069, Australia.
david.reid@csiro.au

Abstract. This paper describes recent advances in the development of an integrated inertial guidance system for automation of the longwall coal mining process. Significant advances in longwall automation are being achieved through an industry sponsored project which targets productivity and safety benefits. Stabilised inertial navigation techniques are being successfully employed to accurately measure the three-dimensional path of the longwall shearer. This enabling technology represents a breakthrough in achieving practical and reliable automated face alignment. This paper also describes a specification for the interconnection of underground mining equipment based on the newly developed EtherNet/IP control and information protocol which ensures equipment compatibility across multi-vendor components of the automation system.

1 Introduction

Recent advances in the development of an integrated guidance system for automation of the longwall coal mining process using inertial navigation techniques have been made by the CSIRO Mining Automation group. Longwall mining is used extensively worldwide and accounts for over 80% and 50% of underground coal production in Australia and the USA respectively.

Modern longwall mining operations have been optimized to the point where manual operation of the process limits the future potential of the mining method. In addition, the necessary hands-on approach to the operation exposes the operators to hazardous working conditions. In 2001 the Australian coal industry provided major Landmark funding to the CSIRO Australia to develop a longwall mining automation system to the level of “on-face observation”. This project builds on previous success in highwall mining automation [1]. More details on the Landmark project can be found at the project website [2].

The first major component of this development is the demonstration of automated longwall face alignment. This achievement alone yields significant productivity and safety benefits across the industry. An international patent covering this technology development has been granted [3].

Details of the longwall mining process and the need for automation are presented in Section 2. Section 3 describes the automated face alignment system and the use of stabilized inertial navigation techniques. A Landmark specification for the interconnection of multi-vendor mining equipment is presented in Section 4.

2 The Longwall Mining Process

Longwall mining is a full extraction mining process in which large panels of a coal seam up to 5m thick are completely mined. An indicative longwall panel is 250m wide by 2000m long. A longwall mining system used in this process is comprised of three main components: a shearer, an armoured face conveyor (AFC) and a roof support system.

A longwall shearer as shown in Figure 1, is up to 15 metres long, weighs 90 tonnes and typically extracts a one metre slice of the coal seam as it travels back and forth across the panel along rails attached to the AFC. Portions of the roof support system and AFC can also be seen in Figure 1. The roof support system can have over 200 individual hydraulic support modules which collectively provide temporary support of the roof material above the extracted coal seam. The load capacity of each support can exceed 1000 tonnes.

As the shearer moves across the coal seam, large hydraulic push rams attached to the roof support modules are used to progressively advance the AFC, and thereby the shearer rails, behind the shearer in a snake-like manner. A shuffle is required at each end of travel to advance the end portions of the AFC. The snake and shuffle are indicated in Figure 2. As the longwall equipment progresses in this manner, the roof material collapses into the void left behind the advancing system.

The complete longwall system is a mobile semi-autonomous underground mining machine weighing in excess of 700 tonnes with each of the three main components operating under largely independent and proprietary control systems.

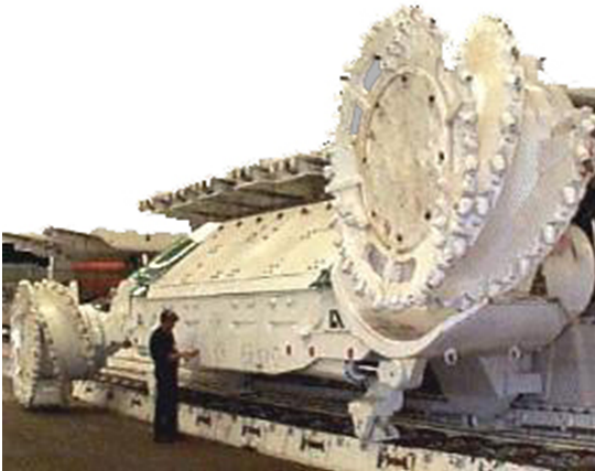


Fig. 1. A longwall shearer showing the leading and trailing drum. A portion of the roof support canopy and AFC are also visible.

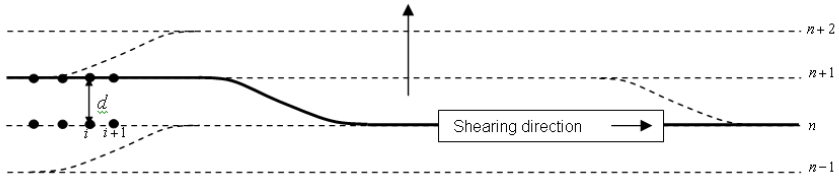


Fig. 2. Schematic of shearer path and AFC profile in plan view for a typical cutting mode. The advancing AFC (solid line) bends to form a snake. The shearer path (dashed line) shuffles at each end of travel

2.1 The Need for Automation

Automation of the longwall mining process has always held the lure of increased productivity but more recently is being driven by issues of occupational health and safety. The presence of hazardous gases, respirable dust and the inherent danger of personnel working in close proximity to large mobile mining equipment is becoming increasingly unacceptable.

There have been many attempts worldwide over a number of decades to achieve full automation of the longwall mining process [4]. Equipment manufacturers have invested heavily in ongoing development of their respective proprietary control systems and yet, to date, personnel are still required to routinely work in hazardous production areas and to manually control the mining process.

Previous automation attempts have in large part been stymied by the inability to accurately determine the three-dimensional path of the longwall shearer as it systematically progresses through the coal panel. Without this information there is no absolute reference for controlling the motion of the equipment and reliable, sustained automation can not be achieved.

Automated face alignment is a major deliverable of the Landmark project. Face alignment refers to the process of maintaining a desired path for the shearer in the horizontal plane as it “slices” across the coal face. In order to minimize mechanical stresses on the mining equipment and maximize production, the face is generally required to be straight and at a geodetic heading nominally orthogonal to the direction of panel progression.

Face alignment is presently achieved by manually aligning the position of the AFC and each roof support with reference to a string line deployed across the face for the purpose. This adjustment is typically required about every eight hours of operation and is both time consuming and non-productive.

3 Automated Face Alignment

Automated face alignment is achieved using the horizontal position information from the shearer-mounted inertial navigation system (INS) as it travels along the

AFC. As represented in Figure 2, at any particular time, the AFC at the i th support is moved from the n th to the $n + 1$ th shear cycle. The distance d is computed from INS information which is gathered during the $n - 1$ th pass so as to achieve a desired face profile.

In control theory terms, the desired face profile, which includes the absolute geodetic heading, is the system set point. The desired face profile is typically a straight line but other non-straight profiles could be advantageous under certain geological conditions. The control system output is the proportional control of the AFC movement via the roof support system. Negative feedback is provided by the shearer-mounted INS which measures the three-dimensional position of the shearer at closely sampled points across the face. Position error in the AFC proportional control is represented as a system disturbance.

Due to INS processing requirements, shearer position data is batch processed at the end of each full face traverse so that the profile corrections made during the $n + 1$ th shear cycle are computed from data gathered throughout the n th cycle. Profile correction values are calculated as shown in Figure 3. The correction values (solid arrows) at positions corresponding to each roof support module are normalized to be zero at points where no correction is required (point D) and negative valued elsewhere. For each increment of panel progression the required advance distance at each roof support module is then computed by the roof support control system as the addition of the correction value and a constant default advance distance (typically 1m). A correction of zero at all points across the face will result in the longwall progressing the default distance. This strategy ensures that the mining process can continue under open loop control during periods where the correction information is unavailable. An example of the shearer path under open-loop control as measured by the shearer mounted INS is shown in Figure 4. The vertical path projected onto the vertical plane correctly follows the natural undulations in the coal seam and is consistent across the multiple shear cycles. The horizontal component as projected

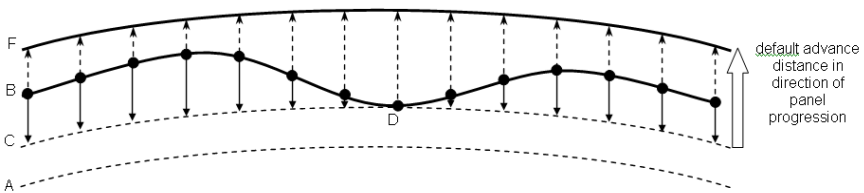


Fig. 3. Diagrammatic representation of the relationship between the desired face profile (dashed line A or C), actual face profile (solid line B), normalised position correction values (solid arrows), required advance distance (dashed arrows) and the resulting face profile (solid line F).

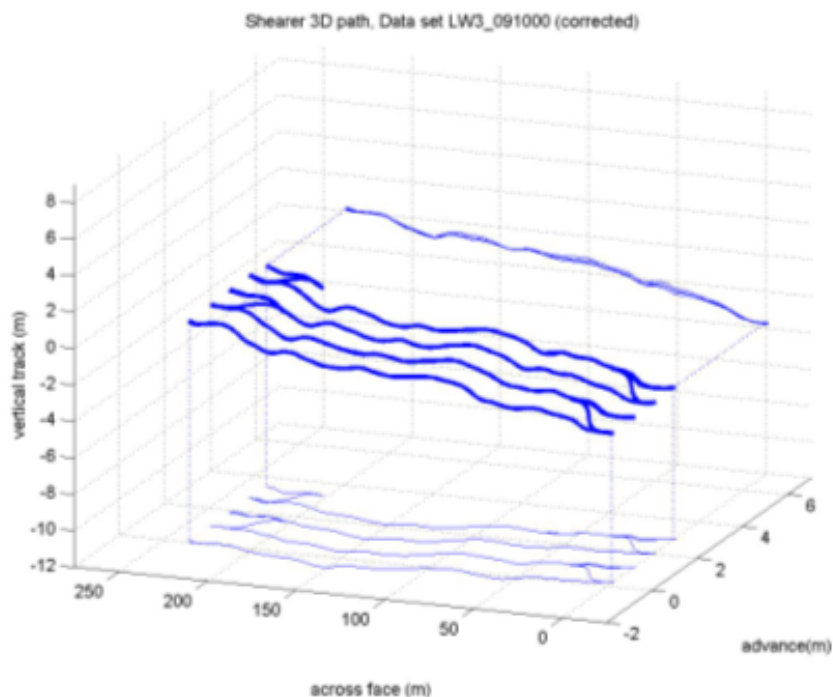


Fig. 4. Three-dimensional path of the moving shearer throughout a number of shear cycles as measured by the INS.

onto the horizontal plane highlights the departure in the face profile from the desired straight line due to accumulated position errors in the open-loop face alignment control system. It is interesting to note that at the time this data was collected, the longwall operators determined by visual inspection that the face profile was straight.

In the automated face alignment system these position errors are minimized by systematically adjusting the AFC movement at each roof support module. Full underground trials of the automated face alignment system are planned for second half of 2003 at Beltana Colliery, NSW Australia.

The performance of the automated face alignment system is critically dependent on the accuracy and precision of the INS.

3.1 Stabalised Inertial Navigation System

Inertial navigation systems are subject to position drift with time mainly as a result of the numerical double integration required to compute three-dimensional position

from three axis acceleration. Dead-reckoning techniques using external odometry can be used to improve short term position stability but systematic drift can still occur if the incremental motion of the INS is not exactly along the measured geodetic heading. High performance INS, such as the military grade units used in this project, typically use GPS aiding to correct this inherent drift. This integrated approach combines the short term accuracy of the INS with the long term stability of GPS.

In the underground mining application GPS is not available and so other bias correction strategies were developed. Without effective bias correction the INS derived shearer path may diverge (or converge) in both the horizontal and vertical components [5]. An example of this divergence is apparent in Figure 5 which represents the uncorrected data of Figure 4.

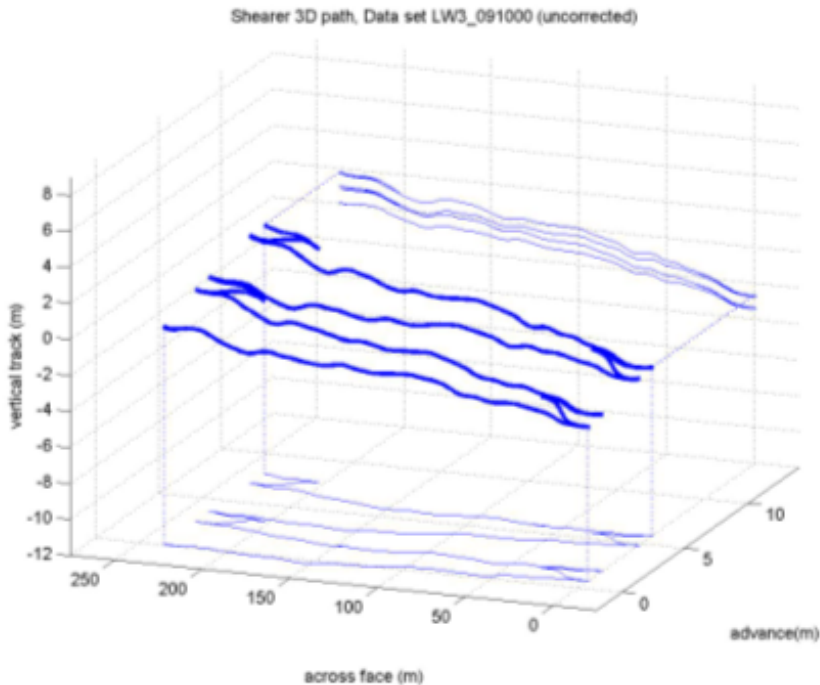


Fig. 5. Three-dimensional path of the moving shearer throughout a number of shear cycles as measured by the INS without bias correction. Divergence in the shearer path due to this bias is apparent.

INS stabilisation techniques generally rely on externally available position or velocity information such as GPS, vehicle odometry or zero velocity updates (ZUPTs).

In the Landmark project INS stability has been achieved by recognising the (almost) closed-path of shearer travel throughout each shear cycle. In normal mining operations the horizontal closing distance for each cycle is either fixed or can be independently determined. This information is used in the automated face alignment system to back-correct the shearer path at the completion of each shear cycle. Similarly, back-correction in the vertical plane can be achieved based on independently surveyed levels which are generally available at the panel boundaries.

4 Equipment Interconnection Standard

The Landmark automation strategy combines new enabling technologies with existing proprietary control systems from the major international equipment manufacturers. These manufacturers are working closely with the Landmark project to integrate their proprietary control systems while maintaining market differentiation and protecting proprietary knowledge. The practical success of the Landmark automation project therefore depended heavily on establishing an industry acceptable data and control standard across the various equipment components. This standard needed to:

- Take advantage of the existing Ethernet cabling and network infrastructure available in many mines
- Allow mine operators to mix and match mining equipment from various vendors
- Be non-proprietary and easily maintained
- Support future development and system expansion.

A Landmark specification has now been developed and accepted by the industry for each of the major equipment components. This specification is based on the newly developed EtherNet/IP control and information protocol managed and promoted by the Open DeviceNet Vendor Association (ODVA). EtherNet/IP combines the proven and popular application layer protocol of DeviceNet and ControlNet with the convenience, bandwidth and flexibility of Ethernet hardware and internet protocols.

The choice of EtherNet/IP gives the mining industry the ability to leverage the rapid advances being made in Ethernet technology driven by the vast enterprise market and increasingly by the industrial control market.

This ability was demonstrated in the Landmark project by using inexpensive commercial off-the-shelf wireless Ethernet hardware to provide a relatively high bandwidth data link to the moving shearer. The link was established using a number of wireless access points distributed at fixed locations across the longwall face and a workgroup bridge installed on the shearer. These units required very little modification for the underground environment and featured channel diversity and hand-off mechanisms for increased link reliability. These units also offered the convenience of web-based remote administration and configuration.

5 Summary and Conclusions

Longwall mining accounts for a large portion of underground coal production worldwide. The industry is seeking ways to improve productivity and safety for mining personnel. Significant advances in longwall automation are being achieved through the industry sponsored Landmark project. A major deliverable of this project is automated face alignment which promises productivity and safety benefits to the industry. INS-based techniques are being successfully employed in this project to accurately measure the three-dimensional path of the longwall shearer. INS provides the enabling technology for automated face alignment that is paving the way towards full automation of the longwall mining process. Techniques have been developed to ensure the long-term position stability of the INS. A specification for the interconnection of underground mining equipment has been published as part of the Landmark project. This specification is based on the newly developed EtherNet/IP control and information protocol and well positions the mining industry to benefit from rapid advances in network and industrial control technology.

References

1. D. C. Reid, D. W. Hainsworth and R. J. McPhee, "Lateral Guidance of Highwall Mining Machinery Using Inertial Navigation", *4th International Symposium on Mine Mechanisation and Automation*, pp B6-1 B6-10, Brisbane, Australia, 1997.
2. <http://www.longwallautomation.org>
3. D. W. Hainsworth and D. C. Reid (2000), *Mining Machine and Method*, Australian Patent PQ7131, April 26, 2000 and US Patent, May 12, 2000.
4. A. L. Craven and I. R. Muirhead, "Horizon Control Technology for Selective Mining in Underground Coal Mines", *The Canadian Mining and Metallurgical Bulletin*, Volume 93, Number 1040, May, 2000.
5. D. C. Reid, D. W. Hainsworth, J. C. Ralston, and R. J. McPhee, "Longwall Shearer Guidance using Inertial Navigation", Australian Coal Association Research Project C9015 report, June 2001.

Development of an Autonomous Conveyor-Bolting Machine for the Underground Coal Mining Industry

Jonathon C. Ralston, Chad O. Hargrave, and David W. Hainsworth

Mining Automation
CSIRO Exploration and Mining
Technology Court, Pullenvale, Q 4069, Australia.
jonathon.ralston@csiro.au

Abstract. This paper describes the development of a new autonomous conveyor and bolting machine (ACBM) used for the rapid development of roadways in underground coal mines. The ACBM is a mobile platform fitted with four independent bolting rigs, bolt storage and delivery carousel, coal receiving hopper and through-conveyor for coal transport. The ACBM is designed to operate in concert with a standard continuous mining machine during the roadway development process to automatically insert roof and wall bolts for securing the roadway. This innovative machine offers significant benefits for increasing personnel safety and improving productivity. The paper describes the core sensing and processing technologies involved in realizing the level of automation required by the ACBM, which includes online roof monitoring, roadway profiling, navigation, and automatic control of drilling and bolting processes.

1 Introduction

The CSIRO Mining Automation is a group that concentrates on developing and applying modern automation technology to mining equipment and systems. Automation technology has significant potential to meet the mining industry's ongoing need to improve productivity and safety. This is achieved by developing new machines and mining processes, creating predictive maintenance and hazard monitoring systems, adding intelligent sensing and processing systems to existing equipment, and by removing personnel from hazardous environments.

One of the key areas for automation in underground coal mining is the development of the core roadway infrastructure. Roadway development is a complex, expensive and time-consuming process using a combination of different mining machinery to cut a lattice network. The main performance bottleneck in roadway development is the need to constantly halt mining to allow the installation of supporting bolts to prevent the roadway from collapsing. Moreover, the current practice of manually drilling and bolting is one of the most dangerous tasks in underground coal mining, involving significant safety concerns for mine personnel. A real need therefore exists for a rapid roadway development system to minimize personnel exposure to hazardous areas of unsupported roof, as well as to improve the overall production rate of this vital mining activity.

In an effort aimed at addressing this roadway development problem, a new mining machine, known as the Autonomous Conveyor-Bolting Machine (ACBM), has been designed. The ACBM is a mobile platform fitted with independent bolting rigs, coal receiving hopper, bolt storage and delivery system, and a through-conveyor for coal transport. It is designed to follow the path of a continuous miner as it drives a new roadway, automatically inserting roof and wall bolts. Figure 1 shows the ACBM during factory testing. Figure 2 shows the placement of machinery associated the rapid roadway development process, with a leading continuous miner, the ACBM and a shuttle car for coal transport.



Fig. 1. The ACBM showing tramming platform, automatic bolting rigs, receiving hopper, bolt storage and delivery system.

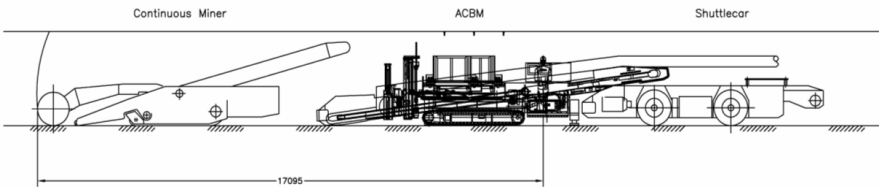


Fig. 2. Placement of machinery associated the rapid roadway development process, with a leading continuous miner, the ACBM and a shuttle car for coal transport. The ACBM showing tramming platform, automatic bolting rigs, receiving hopper, bolt storage and delivery system.

2 ACBM Functional Overview

The ACBM is a mobile platform fitted with independent bolting rigs, coal receiving hopper, bolt storage and delivery system, and a through-conveyor. It is designed to follow the path of a continuous miner as it drives a new roadway, automatically inserting roof and wall bolts. While creating the roadway, coal cut from the tunnel is also transferred via a through-conveyor belt which leads to the shuttle-car which transports the coal to the surface. The cycle time for placement of a row of four bolts is approximately five minutes, allowing a machine advancement rate of approximately 15 m/hour with a 1.2-metre row spacing. The ACBM uses a combination of processing systems in order to provide online roof monitoring, roadway profiling, navigation, and control of drilling and bolting processes. The ACBM control system thus has two fundamental operating modes, namely tramming and bolting.

2.1 ACBM Processing and Control

As the ACBM may be interposed between various production and coal haulage machines, the system has been designed to work either independently or in concert with a modified remote controlled miner with remote controlled bolting capabilities. The ACBM uses a centralized unit for the intelligent control and coordination of a set of distributed computing and sensing modules. The central unit is responsible for ACBM tramming, bolting, and conveyor tasks, as well as generic supervisory tasks such as link/device integrity monitoring and system-wide safety. The aim of the control system is to execute the necessary control over the robotic bolting and motion systems in order to implement the required bolting pattern. The bolting control system is designed to place up to six bolts in a row, oriented from a vertical placement to an outward angle of 15 degrees with a maximum vertical reach of 3.7m from the floor. The block diagram in Figure 3 shows the control hierarchy between the central control unit and associated signal processing components to achieve this goal. Although the ACBM is designed with fully automatic drilling and bolting capabilities, the system can be set into semi-automatic or manual modes. This permits the operator to elect the operational mode. A graphical user interface allows operators to interact with the system.

2.2 Software Architecture

The integrated signal processing component technologies are implemented at three orthogonal layers: Validation, execution, and functional. The validation layer has the highest priority and is responsible for top-level intersystem and inter-machine coordination, system mode resolution, integrity monitoring and other safety related logic decisions. The execution layer controls and coordinates the dynamic execution of main functions such as drilling and bolting sequencing, profiling and drill monitoring. The functional layer contains modules that encapsulate all device specific interface and control details (such as drivers and communication protocols) for the sensors and actuators of the machine.

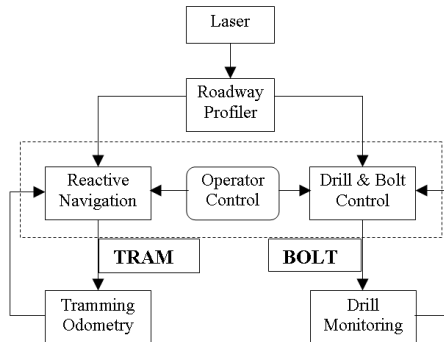


Fig. 3. Block diagram of the signal processing components associated with the ACBM.

The software design thus serves to effectively decouple high priority safety functions from the real-time signal processing activities. The architecture also greatly facilitates the incorporation of new devices or new machine behaviour.

2.3 Software Framework

The complex algorithms needed for the real-time process and control of the four asynchronous drilling and bolting rig sequences present an interesting challenge. This led to the development of a high-level scripting language and execution engine specifically designed for codifying the behaviour of a human operator. The script-based language is derived from the notion of a virtualised programmable logic controller (PLC), and is thus known as the VPLC. The VPLC is a state-based processing framework for the implementation of generic industrial automation and control tasks.

VPLC scripts are used to describe the desired system behaviour. The VPLC run-time engine implements an indeterminate finite-state Moore machine. A Moore-based state machine associates an output whereas a Mealy-machine associates an output to a state transition. Using the Moore-based machine thus simplifies behaviour codification and run-time validation. The VPLC language employs constructs that would be typically expected of modern automation-oriented languages such as locally and globally scoped variables, timers, temporal and persistent data objects, fast IO access, conditional evaluations, assignment operators, transitions and state definitions. The VPLC engine is based entirely on the C++ standard template library, the implementation is clean, efficient, cross platform and readily scalable.

One of the key benefits of the framework is that it allows for rapid prototyping and development in an environment suitable for deploying the ACBM processing and control algorithms. The scripts are in plain-text format and of arbitrary length. At program run time, the scripts are read, verified and evaluated and thus no source code recompilation is required when a new behaviour or feature is added. Using this approach, the system behaviour is created entirely through scripted configuration

data, rather than through source code. This has significant benefits in terms of the speed at which system modifications and enhancements can be implemented. The system thus provides a rapid prototyping environment for monitoring and control processes, and enables dynamic reconfiguration of system behaviour – an important aspect in industrial contexts where system specifications are frequently modified.

3 Laser Profiling and Navigation

Four independent laser measurement sensors are used to provide cross sectional roadway profiling and navigational information [3]. A fifth laser sensor is reserved for analysis of coal-flow on the through-conveyor and is discussed elsewhere. The laser sensor data is augmented with independent tramming (odometer) inputs for secondary platform motion validation. The machine does not rely on additional infrastructure such as waypoints or reflective tape for the profiling and navigation tasks.

3.1 Roadway Profiling

Cross sectional profiles of the roadway are required at prospective bolting locations to ensure that the drilling rigs are optimally orientated for bolt placement. Ideally, the roof and rib (side wall) surfaces should be perpendicular to the respective bolting rigs and the distance to the surface must be within the limits of the rig stroke. If these conditions are out of tolerance, the bolting process may halt. The profiling process can thus warn the operator and also search for a better location for bolt placement. Figure 4 shows a series of typical tunnel cross-sectional profiles acquired as the ACBM progresses through the roadway.

3.2 Navigation

Laser sensors also provide important information for navigational purposes. It is necessary to maintain a suitable separation between the ACBM and the miner for coal flow management, and to provide a collision avoidance mechanism when the ACBM-miner separation is too small. Optimal bolt placement also requires that the orientation of the ACBM be positioned along the centreline of the roadway, i.e., equally displaced from the ribs. Given the relatively slow velocity of the platform and the constrained tunnel environment, a conventional reactive navigation algorithm provides a simple and robust method for both collision alerts and ACBM orientation. Figure 5 shows the physical arrangement of the ACBM in the roadway.

4 Drill Monitoring System

4.1 The Need for Drill Monitoring

It is critical for safety that the supporting bolts are securely anchored in solid rock. This means that not only must the bolt be appropriately torqued when fastened, but

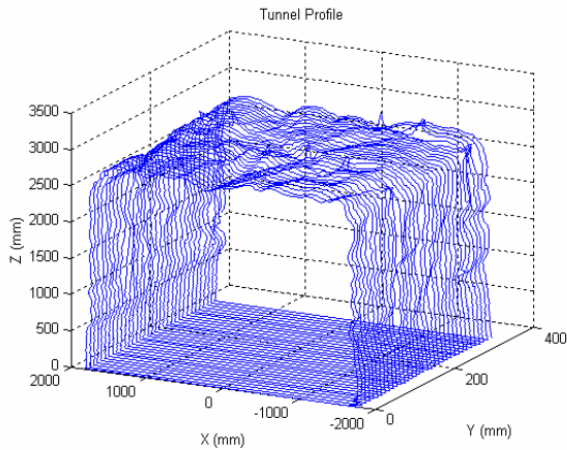


Fig. 4. Typical laser-generated roof and wall profile formed as the ACBM trams along the underground roadway.

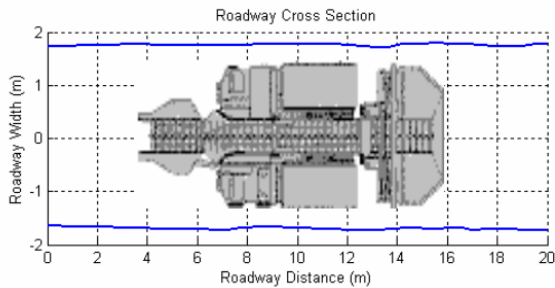


Fig. 5. Roadway data derived from the laser profile used for generating ACBM navigation proximity alerts.

also that the basic composition of the strata be known. Although predictive coal interface detectors have been considered for this class of problem [4], the need to drill and bolt for roadway integrity supports a more direct approach.

The need for drill monitoring is particularly important as a machine is replicating a function normally fulfilled by an experienced underground operator. To this end, an online in-situ drill monitoring system is needed in order to assess the quality of the bolting process and provide information on rib and roof integrity.

4.2 Neural Network Classifier

The drill monitoring system is designed to detect layers, cracks and discontinuities in the drilled strata. The roof drilling rigs on the ACBM are instrumented to provide

sensor feedback during each drilling and bolting phase. The key physical parameters measured for drill monitoring purposes are torque, rotational rate, thrust, and penetration rate. These signals are shown in Figure 6.

An important parameter used in drill monitoring is the specific energy of drilling, SED , which expresses the linear and rotational energy needed to drill a given volume of material, i.e.,

$$SED = \frac{F}{A} + \frac{\omega\tau}{k A d}$$

where F is the thrust force, A is the area of the drill hole, ω is the drill rotary speed (RPM), τ is the drill torque, d is the drill displacement and k is a normalisation constant [6]. SED is of special interest for strata characterization problems as it can be used to determine the relative strengths of strata and geological features.

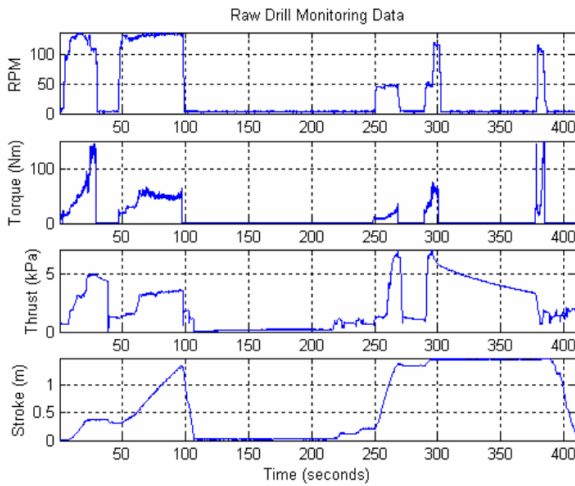


Fig. 6. The parameters measured for drill monitoring: RPM, torque, thrust, and stroke.

Other metrics such as torque versus thrust can also be used as feature inputs [5]. The drill monitoring process consists of three major components: Data acquisition, feature conversion, and strata classification. A neural network based classifier is used to estimate the characteristics of rock strata, where the SED derived from the drill monitoring data provides an additional feature for the classifier. The neural network architecture is particularly well suited to this classification problem due to the highly nonlinear and time-varying characteristics of the drilling process. A detailed survey of the ACBM neural network classifier implementation can be found in [5].

5 Field Implementation

5.1 Operator Console

Special design and construction considerations were necessary to make the control and monitoring components of the system suitable for use in an underground coal mining environment. For example, Figure 7 shows the explosion-proof operator control interface with push buttons, operator visualisation, and remote video display. Two such consoles are used on the left and right hand sides of the ACBM to enable control from either side.



Fig. 7. The flameproof enclosure housing the operator display, input controls and computing hardware.

5.2 Operator Interface

Due to the potentially explosive gases present in an underground coal mine, all electronic equipment that is not rated as intrinsically safe must be housed in a flameproof enclosure to protect the external environment from any sparks, high temperatures, or flames that the equipment could generate under fault conditions. The relatively tight space considerations, and concerns for a simple and physically robust interface, meant that an intrinsically safe keyboard (connected to the computing equipment inside the flameproof) was unsuitable for the application. Instead, external push buttons were mounted on the flameproof console with the button contacts wired to discrete inputs on a data acquisition card in the main control computer.

The operator uses these buttons to control the ACBM by navigating through a graphical user interface (GUI) displayed on the system monitor which is housed within the flameproof cabinet. The GUI is designed to provide a simple and meaningful interface for an operator with little or no computing experience. Due to the limited number of pushbuttons on the flameproof, almost all of the control commands must be implemented using the primary navigational inputs, i.e., up, down,

left, right, and enter. To manage this limited interface, the GUI screens were designed to optimise functionality while minimizing the number of button presses required. Figure 8 shows a typical screenshot from the operator user interface.

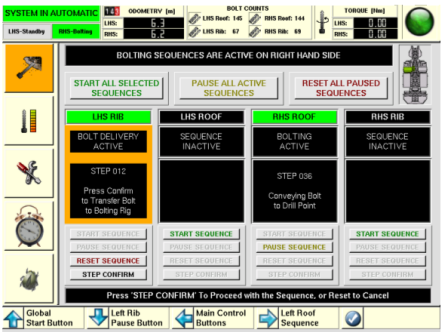


Fig. 8. Screenshot showing operator GUI for controlling and monitoring all ACBM drilling, bolting and traming functions.

5.3 Embedded Computing Hardware

There are many challenges developing electronic hardware that can withstand the hostile conditions of the underground coal mining environment: Water, vibration, dust intrusion, shock, and heat all impact on system reliability issues. As a result, the ACBM employs ruggedised PC104-based industrial modules for implementing the high-level computing tasks associated with the ACBM control. These modules have proven to be an effective and reliable platform for controlling the ACBM.

The computing systems also need to operate in presence of potentially explosive gases and thus need to be housed in a special manner. This requires that the ruggedised processing modules be placed into flameproof enclosures to ensure that they present no explosion risk.

6 Summary

This paper presents the design of a novel application of robotics and automation technology for the mining industry, featuring an autonomous mobile bolting platform designed to work with a conventional continuous miner. The immediate implications of the ACBM include the removal of personnel from hazardous areas of unsupported roof, as well as the potential to significantly improve the rate of roadway development. The key to the successful deployment of the automation task machine lies in the hardware and software system design, which integrates the disparate components of the system to realise the overall automation task.

References

1. M. Kelly, "Improving roadway development systems through automated roof bolting technology," *Proc. Australia Japan Technology Exchange Workshop: Coal Resources and Coal Mining Technology*, 12pp. Brisbane, Australia, 1999.
2. J. C. Ralston and D. W. Hainsworth, "An Autonomous Bolting Machine for Rapid Roadway Development," *IEEE Int. Conf. Mech. and Mach Vision in Practice (M2VIP '2001)*, Hong Kong, August 2001.
3. C. Ye and J. Borenstein, "Characterization of a 2-D Laser Scanner for Mobile Robot Obstacle Negotiation," *Proc Inter. Conf. Robotics and Automation*, Washington, pp. 2512-2518, May 2002.
4. J. C. Ralston and D. W. Hainsworth, "Application of Ground Penetrating Radar for Coal Depth Measurement," *ICASSP '99*, Vol. 4, pp. 2275-78, (Arizona), March 1999.
5. K. Itakura, K. Sato, Y. Ichihara, G. Deguchi, H. Matsumoto, and H. Eguchi, "Development of a roof logging system by rock bolt drilling", *Trans. Inst Min. Metal*, Vol.106, Sect. A, (1997), pp.118-123.
6. R. Teale, "The concept of specific energy in rock drilling," *Int. J. Rock Mech. Min. Sci. & Geo. Abs.*, vol. 2 pp. 757-764. 1965.

A Case Study in Robotic Mapping of Abandoned Mines

Christopher Baker, Zachary Omohundro, Scott Thayer, William Whittaker, Mike Montemerlo, and Sebastian Thrun

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
<http://www.ri.cmu.edu>

Abstract. Mining operations depend on current, accurate maps of adjacent mine works to limit the risks of encroachment and breaching. Adjacent mines may be decades or centuries old with missing, inaccurate, or ambiguous maps. Dangers such as flooding, roof-fall, rotten support timbers, and poor ventilation preclude human entry to survey these spaces. Only robots may enter and directly observe these otherwise inaccessible underground voids, providing incontrovertible evidence of the mine's existence and extent. This presents the configuration of a mobile mine mapping robot, *Groundhog*, and results from three deployments into coal mines.

1 Introduction

On July 24, 2002, nine miners working in the Quecreek coal mine accidentally breached the adjacent Saxman mine, unleashing millions of gallons of water contained therein [1]. Trapped by the rising water, the nine miners endured three days underground prior to their rescue. "The root cause of the accident was the unavailability of a certified final map of [the Saxman] mine in the State of Pennsylvania's mine map repository"[2].

There exist surface mapping techniques[3] such as ground penetrating radar and seismic surveying that rely on geophysical models and assumptions to infer the existence of underground voids. The accuracy and reliability of these systems, however, are highly dependent on the depth of the mine and the structure and composition of the surrounding strata.

Robots, on the other hand, may operate safely and reliably inside mines independent of depth or geology. Motivated by the Quecreek accident, Carnegie Mellon has developed *Groundhog*, a 700kg mobile mine mapping robot. Built to traverse the rough, unpredictable terrain of abandoned mine corridors, *Groundhog* provides a means of conclusively establishing the extents of an abandoned mine by direct observation.

2 Chassis and Electronics

Groundhog's chassis is based on the union of two front halves (steering columns and differentials) from recreational all-terrain vehicles, allowing all four of its wheels

to be both driven and steered(Fig.1). The frame has been reinforced to accommodate a 500kg payload, and the two steering columns are cooperatively linked, reducing *Groundhog*'s outside turning radius to 2.5 meters. The steering linkage is driven by a single hydraulic cylinder with potentiometer feedback providing closed-loop control of steering direction. Emphasizing mobility over speed, hydraulic motors drive the front and rear ATV differentials through 3:1 chain reduction, resulting in a maximum velocity of 15 cm/s. When in motion, *Groundhog* consumes roughly one kilowatt of power, and at such low speeds generates the torque necessary to overcome the fallen timbers and other rubble commonly found in abandoned mines. Processing and sensing draw less than 100 Watts, so the operational range of the robot is dominated by its locomotive load. Equipped with six deep-cycle lead-acid batteries providing more than 6 kWh of energy, *Groundhog* has a locomotive range greater than 3 km.

Groundhog is just under 1 meter tall and is 1.2 meters wide, originally intended to navigate the breach between the Quecreek and Saxman mine, which was 1.2 meters tall and 2 meters wide. The investigation and subsequent sealing of the breach prevented *Groundhog* from exploring Saxman, but its configuration and dimensions have proven effective for operation in most mines.



Fig. 1. Left: The base chassis, composed of two connected ATV front-ends. Right: *Groundhog*, 43 days later, just before its first deployment into an abandoned mine.

Within mines, pockets of gas accumulate, forming explosion hazards. Mine safety regulations[4] for permissible equipment require that all electronics either be intrinsically safe¹ or be encased in an explosion-proof enclosure.² While complete and rigorous certification is not appropriate for a prototype robot, *Groundhog* incorporates many features of a permissible device, including mine-certified batteries,

¹ An intrinsically safe device cannot, through capacitance or inductance, discharge enough energy into a spark to ignite an explosive atmosphere

² Explosion-proof enclosures are designed to control the release of energy from an explosion within the enclosure to prevent the ignition of an explosive external environment.

a hydraulic locomotive system and a 225kg steel enclosure for the majority of its electronics.

The explosion-proof enclosure houses an industrial DC motor that drives the hydraulic system. A 300 MHz PC/104+ CPU and associated I/O electronics also occupy the enclosure along with a hydraulic manifold with six solenoid actuators. All outgoing power lines are computer-controlled and individually fused to limit the risk of an explosion from an electrical short. The CPU continuously monitors two Industrial Scientific TMX412 combustible gas sensors and may cut power to external devices if a hazardous environment is detected. A hardware watchdog timer automatically stops the robot and disables outgoing power in the event of a computer failure.

3 First Field Deployment

Groundhog was first deployed into the Florence mine in Burgettstown, PA, USA. Florence was a conventional deep mine that had been abandoned for more than 80 years. A local mining company encountered Florence while strip-mining the area and offered an opportunity to explore the mine.

3.1 Configuration

For the experiments at Florence, *Groundhog* was teleoperated over a 100 meter ethernet connection and was tethered to the surface by a steel cable in case of emergency. Additionally, two lengths of coaxial video cable fed signals from forward- and rear-facing cameras to the operations center. *Groundhog's* mapping payload consisted of two SICK LMS-200 scanning laser range-finders. One faced forward to generate a two-dimensional map of the mine via scan matching as outlined in [5]. The second laser faced upward to scan vertical profiles of the mine which would later be stitched together into a three-dimensional representation of the upper half of the mine, also described in [5]. This configuration is illustrated in Fig. 2.



Fig. 2. Left: Scanner configuration for Florence. Right: *Groundhog*, half submerged, deep inside the Florence mine.

3.2 Conditions

The Florence mine had been flooded since it was abandoned in the 1920's. As a result, the floor of the mine was covered in more than 20 cm of sulfurous sludge³, obscuring the terrain beneath, littered with indeterminable boulders and ditches. The water from the mine was drained at the portal, but it sloped downward into the hillside, causing the water level gradually rise as the robot progressed inward.

3.3 The Experiment

Groundhog mapped the Florence mine on October 27 and again on October 29, 2002. On the 27th, *Groundhog* made two forays into the mine, reaching a maximum of 30 meters before a water-induced electrical failure terminated the experiment. *Groundhog* negotiated the terrain without difficulty in spite of some of the worst conditions to be found within an abandoned mine. The machine was re-fitted with cameras and made more water-resistant for its return to Florence on the 29th. With the additional waterproofing, *Groundhog* was able to proceed an additional 10 meters for a total of 40 into the mine before becoming dangerously submerged. Fig. 2 shows the robot at its farthest point, sunk in 40-50 cm of sludge and water.

3.4 Results

For these teleoperated experiments the data from the laser scanners was time-stamped and logged for post-processing using scan matching techniques as described in [5]. The two-dimensional map shown in Fig. 3 shows the entrance in the lower-left corner and represents approximately 45 meters from the entrance of the mine to the far extent of the map. Fig. 3 also shows a snapshot of the three-dimensional model of the mine at the portal.

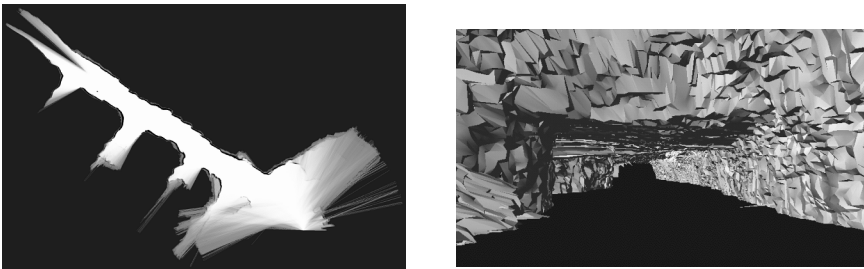


Fig. 3. Left: Two-Dimensional Map of Florence, representing 50 meters from corner to corner. Right: 3D Snapshot of the Florence Portal.

³ Called “Yellow Boy”, this sludge is the acidic, sulfur-laden product of exposing the surrounding strata to water.

4 Endurance Testing

The Florence deployments were successful demonstrations of *Groundhog*'s ability to operate in a treacherous environment, but they did not test the machine's ability to operate for extended periods of time nor over large distances. To that end, *Groundhog* was taken to the NIOSH⁴ Safety Research Mine in Bruceton Mills, PA with the goal of mapping at least 1 km of mine corridors.

4.1 Configuration

Groundhog's hydraulic system is operated in a binary fashion, limiting the machine to exactly one speed. During the Florence experiments, *Groundhog* moved very slowly (7-8 cm/s) and had no trouble overcoming obstacles, so the machine was re-gearred to double the speed to the current rate of 15 cm/s. Notwithstanding the changes to the drive-train, *Groundhog* was configured in almost the same way as for the Florence experiments. The research mine was actively ventilated, so the gas sensors were unnecessary, and rather than managing a wire tether, *Groundhog* was teleoperated over a wireless ethernet link from a mine cart following roughly 10 meters behind.

4.2 Conditions

Bruceton is a well-kept safety research mine and is thus in complete contrast with the abandoned Florence mine. Bruceton is well-ventilated, and the floors are dry, groomed, and clear of debris. The walls and ceiling are coated with concrete to maintain the integrity of the mine, and there are no low-hanging obstructions. Yet, Bruceton and Florence are both coal mines and thus share similar size, shape and general structure. They present many of the same locomotive and navigational challenges, including tight turns and narrow passages. For Bruceton, however, there is a readily available survey of the mine for comparison to *Groundhog*'s map, both shown in Fig. 4.

4.3 The Experiment

Groundhog was driven in this configuration for 3.5 hours, traversing in excess of 1.6 km of the Bruceton mine. In addition to testing the electromechanical endurance, the Bruceton experiments generated a data set that was several times larger and more complex than the mapping software had previously processed. This provided an invaluable means of testing and extending the software that would become a core component in subsequent experiments in exploration autonomy.

⁴ National Institute of Occupational Safety and Health

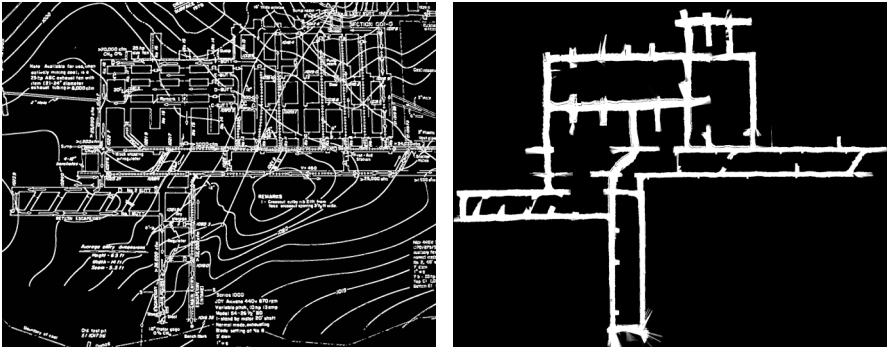


Fig. 4. Left: Current survey of the NIOSH Research Mine, with edges enhanced for clarity. Right: Automatically generated map, scaled and aligned for comparison.

4.4 Results

Fig. 4 shows the two dimensional map generated by the software outlined in [5]. From left to the right, the map represents approximately 300 meters. There is clear geometric correlation between the human-surveyed map and *Groundhog's* map. *Groundhog's* map is, however, much richer in fine details, revealing rounded corners and small deviations in the walls that are beyond the resolution of a human survey team. Formal verification of the accuracy of the maps is scheduled for Summer 2004.

5 Second Field Deployment

Having demonstrated the electromechanical capabilities of the platform, exploration and navigation autonomy were added to the system with the goal of autonomously exploring at least 300 meters of an abandoned mine. The autonomy system is carefully detailed in [6]. *Groundhog* was subsequently deployed into the abandoned Mathies mine in New Eagle, PA at the end of May, 2003.

5.1 Configuration

To facilitate three-dimensional scanning and obstacle avoidance, *Groundhog's* dual front lasers were replaced with a single laser on a tilt unit, shown in Fig. 5, that may be tilted from -90 to $+120$ degrees to generate 3D scans of the environment. An identical unit was also placed on the rear of the robot to allow the machine to reverse course without having to turn around. To provide imagery from inside the mine, *Groundhog* was fitted with a low-light camera and infrared LED ring, as shown in Fig. 5. The signal from this camera was both recorded on-board and transmitted to the surface. *Groundhog's* batteries were upgraded from four 6-volt cells to six 8-volt cells, boosting its operational range to more than three kilometers. Minimal operational feedback, little more than an indication that the machine was still operating, was sent to the base of operations over the wireless ethernet link.

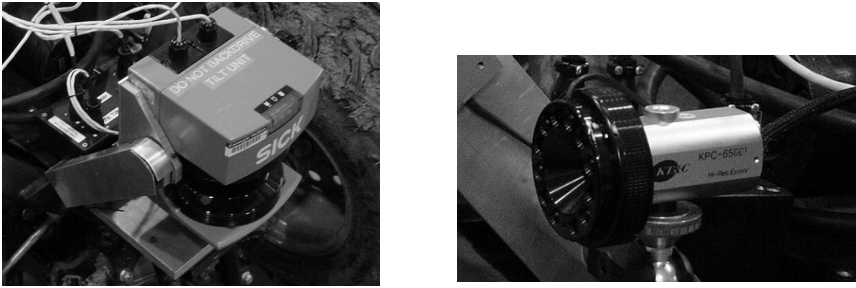


Fig. 5. Left: The forward scanning laser rangefinder and custom tilt unit. Right: Low-light camera and infrared LED ring.

5.2 Conditions

Mathies had been used as a coal-haulage route, effectively a tunnel, that ran for more than 1km through a mountain until it closed in early 2000. The corridor was 5-7 meters wide with a drainage ditch on the right-hand side and was expected to be dry and in fair condition. The intent was to autonomously traverse the length of the corridor from one portal to another, constructing a complete 3D model of the mine. If an insurmountable obstacle was encountered, *Groundhog* was to reverse course and proceed back out the way it came in. PA-DEP⁵ would use the data to determine the feasibility of running a pipeline to move acidic mine runoff through the mine to a treatment facility.

5.3 The Experiment

Groundhog entered the Mathies mine on May 30, 2003 at 10:55 AM, EST. One hour and 308 meters into the mine, the robot encountered a fallen support beam that



Fig. 6. *Groundhog* poised at the entry to the Mathies Mine.

⁵ Pennsylvania Department of Environmental Protection

blocked further progress. The machine started back out of the mine at 12:03 PM, but encountered software difficulties starting at approximately 12:20 PM. After another 30 minutes, the system had not resolved its problems, and manual intervention was attempted over the weak wireless link at 12:56 PM.

Under the strain of teleoperation, the wireless link locked up shortly thereafter, stranding the robot an estimated 200 meters inside the mine at 1:04 PM. Subsequent efforts to re-establish the link failed, and at 3:30 PM, two mine safety inspectors received permission to suit up and proceed into the mine to manually reset *Groundhog*'s wireless link. The link was successfully re-established at 3:50 PM and the robot exited the mine under manual control at 4:02 PM.

5.4 Results

Fig. 7 shows a 50 meter portion of the 300 meter map of the mine *Groundhog* explored. Also shown is a 3D scan of the roof-fall that prevented the robot from continuing forward through the mine.

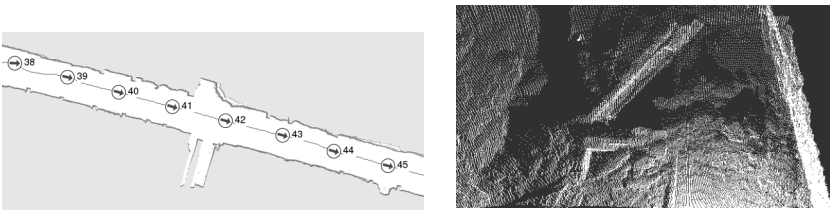


Fig. 7. Left: Two-Dimensional Map of Mathies. Right: Fallen support timber blocking the path of the robot, 308 meters into the mine.

6 Summary

Groundhog is a robust platform for the autonomous mapping of abandoned mines. It has been successfully deployed in two abandoned mines, generating hundreds of meters of detailed 2D maps and 3D models where human survey teams could not safely operate. In combination with state-of-the-art algorithms, it has generated kilometers of mine maps that are detailed well beyond the resolution of manual surveys and are expected to exceed them in accuracy as well.

7 Future Work

Continued testing and development using the *Groundhog* platform is planned for the foreseeable future. Additional missions into the Mathies mine are scheduled

for October 2003, and formal verification of the accuracy of the mapping algorithms is planned for early 2004. Several other mine mapping platforms are also under development, leveraging the unique knowledge and experience gained during *Groundhog's* field deployments.

Groundhog's greatest disadvantage is that it requires portal access to the mine, and while some abandoned mines have open portals, the majority are sealed and can only be accessed by drilling a borehole from the surface into the mine. Therefore, borehole deployable mine mapping robots represent the next wave in economical and effective mine mapping devices. Two such devices are currently under development. *Ferret*, whose exploits are also published in these proceedings, is a borehole-deployable stationary scanning laser range finder. *Helix*, still in the early stages of development, will be a mobile robot which carries a scanning laser range finder, yet can fold into a 15cm diameter cylinder for borehole deployment.

Acknowledgments

Groundhog was the result of the combined efforts of the many students, instructors, and consultants who participated in the fall and spring mobile robot development classes offered through the Robotics Institute at Carnegie Mellon. We would also like to acknowledge the guidance and assistance provided by the Mine Safety and Health Administration, the National Institute of Occupational Health and Safety, and the Pennsylvania Department of Environmental Protection.

References

1. M. Ross and M. Roth, "All Nine Alive: The Story of the Quecreek Mine Rescue" *Pittsburgh Post Gazette* 04 Aug 2002.
2. US Mine Safety and Health Administration, "Report of Investigation: Underground Coal Mine Nonfatal Entrapment: July 24, 2002" US Department of Labor, Released August 12, 2003.
3. USGS CMG InfoBank Definitions
<http://walrus.wr.usgs.gov/infobank/programs/html/main/definitions.html>
Current as of 23 Nov, 2003
4. US Mine Safety and Health Administration, "Code of Federal Regulations: Title 30: Parts 1-199: Mineral Resources" US Department of Labor, Mine Safety and Health Administration, 23 December, 1992.
5. M. Montemerlo, et. al., "A System for Volumetric Robotic Mapping of Underground Mines" Carnegie Mellon Technical Report CMU-CS-02-185
6. D. Ferguson, A. Morris, D. Hähnel, et. al. An Autonomous Robotic System for Mapping Abandoned Mines. *NIPS-2003*

Automatic 3D Underground Mine Mapping

Daniel F. Huber and Nicolas Vandapel

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
dhuber@ri.cmu.edu & vandapel@ri.cmu.edu

Abstract. For several years, our research group has been developing methods for automated modeling of 3D environments. In September, 2002, we were given the opportunity to demonstrate our mapping capability in an underground coal mine. The opportunity arose as a result of the Quecreek mine accident, in which an inaccurate map caused miners to breach an abandoned, water-filled mine, trapping them for several days. Our field test illustrates the feasibility and potential of high resolution three-dimensional (3D) mapping of an underground coal mine using a cart-mounted 3D laser scanner. This paper presents our experimental setup, the automatic 3D modeling method used, and the results of the field test. In addition, we address issues related to laser sensing in a coal mine environment.

1 Introduction

For several years, our research group has been developing methods for automated modeling of 3D environments [3][4][5]. In September, 2002, we were given the opportunity to demonstrate our mapping capability in an underground coal mine, the Mine Safety and Health Administration (MSHA) research mine in Bruceton, Pennsylvania. The opportunity arose as a result of the Quecreek mine accident in July, 2002, in which miners inadvertently breached an abandoned, water-filled mine, trapping themselves amidst thousands of tons of water. After the miners were safely rescued, an investigation was launched to determine the cause of the accident and to identify new procedures necessary to prevent mine breaches in the future. Regulations already in place aim to prevent such an accident: mapping the mine before ending operations, exploratory drilling, and so forth. Unfortunately, old maps may be incorrect, incomplete, or simply lost. In the end, the Quecreek accident was attributed to an inaccurate map [2].

A collaborative effort by several research groups at Carnegie Mellon University (CMU) has been formed to develop robots to autonomously map abandoned mines and active mines before operations are ended. Such robots would be an important contribution to mining safety. Details can be found in [1,14,9]. In this paper, we address the problem of sensing and generating high-resolution 3D models of an active mine. In September, 2002, we conducted a field test to demonstrate the feasibility of high resolution 3D mapping of an underground coal mine using a cart-mounted 3D laser scanner. The remainder of the paper is organized as follows. First, section 2 reviews previous work on mine mapping and localization. Section 3 describes our experimental setup and the data collection process. Section 4 explains

our automatic modeling algorithm and discusses the resulting 3D model. Finally, section 5 presents an analysis of the issues relevant to laser sensing in a coal mine environment.

2 Related Work

In this section, we review the most relevant work on mine mapping and localization. Early work by Shaffer [13] described a method to localize a mobile robot in an underground mine by registering terrain features (corner and line segments) extracted from an *a priori* survey map with cross-sections from an environment map produced by a laser scanner. In [11,12], Scheduling extensively tested a set of navigation sensors mounted on a Load, Haul, and Dump truck (LHD) in the harsh underground mine environment. Using the data from a laser line scanner coupled with the navigation data of the vehicle, he produced a 3D model of a section of the mine. In [7], two line scanners were integrated on an LHD. The iterative closest point (ICP) algorithm was used to register the 2D profiles to an existing map. This implementation was extended to mine mapping in [8]. The contributions presented above focused on vehicle automation for active mines. In the context of mapping abandoned mines, Thrun [14] produced 2D maps and partial 3D models of tunnels, using a SLAM approach with two line scanning lasers mounted on a tele-operated robot [1]. Several systems have been designed to map mines that are inaccessible to a ground robot, for example, by mapping a cavity using a 3D laser sensor inserted through a bore-hole. Such systems include the C-ALS (Cavity Autoscanning laser system) by Measurement Devices, Ltd. and the Cavity monitoring system by Optech, Inc.¹ A similar approach has been followed in [9].

3 Data Collection

For our field test, we used a high resolution 3D laser scanner mounted on a cart as illustrated in figure 1-(a). The sensor, a Zoller and Fröhlich LARA 25200 (Z+F) scanner [6], produces 8000×1400 pixel range and reflectance images with millimeter-level accuracy. The field of view is $360^\circ \times 70^\circ$ with a range of 22.5 m. The laser scan head was inclined to allow higher density scanning of the floor and ceiling near the scanning platform. Unfortunately, in some regions, the low roof was actually too close to the scan head for the sensor to fully scan the ceiling.

We obtained 23 scans at three- to five-meter intervals along a loop trajectory through a sequence of 4 hallways (figure 1-(b)). The cart was kept stationary at each location for the 90 seconds required to obtain each scan. Due to the capabilities of our modeling algorithms, it was not necessary to record the position or attitude of the cart. This greatly simplifies the data collection process. The entire procedure only took about three hours, including setup and disassembly of the equipment. For this experiment, the cart was moved manually, but it would be straightforward to mount the scanner on an autonomous mobile robot.

¹ www.mdl.co.uk, www.optech.on.ca, May 2003

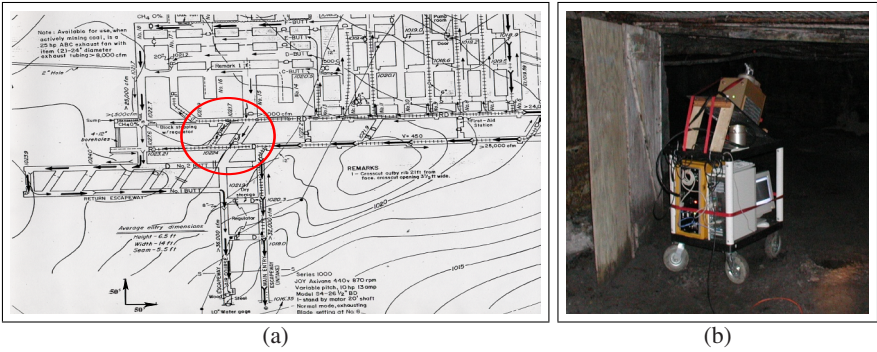


Fig. 1. (a) Surveyed map of the Bruceton mine. The red circle indicates the area mapped in our field experiment. (b) The cart-mounted Z+F laser scanner used in the data collection

4 Automatic Modeling from Reality

Modeling-from-reality is the process of creating digital three-dimensional (3D) models of real-world scenes from 3D views as obtained, for example, from range sensors or stereo camera systems. Recently, we have developed a system that fully automates the modeling-from-reality process [4][5]. The key challenge of automatic modeling-from-reality is the accurate and robust registration of multiple 3D views. Although each input scan is an accurate representation of the 3D structure of the scene as seen from a single viewpoint, the data is expressed in the local coordinate system of the sensor. Our system automatically registers multiple 3D data sets in a common coordinate system without requiring any knowledge of the viewpoints from which the data was obtained. This capability is important in our case, because we did not survey the scan locations during our initial data collection. In a real system, where the sensor would be mounted on a robot, an approximate estimate of the motion between scans may be provided by the robot. Our algorithm has the ability to employ such information when it is available, but, more importantly, it will not break down when the estimates are not available. The problem of registering multiple 3D views obtained from unknown viewpoints is called multi-view surface matching, and it is analogous to assembling a 3D jigsaw puzzle, with each view being a piece of the puzzle.

4.1 Automatic Modeling Method

Briefly, our automatic modeling-from-reality algorithm works as follows²: First, the range images are converted to 3D surfaces, which serve as the input views to the algorithm. A pair-wise surface matching algorithm is then used to identify potential alignments (called matches) between view pairs. The difficulty is that these pair-wise matches may be incorrect, and it is sometimes impossible to distinguish

² Details of the algorithm are given in [4].

correct matches from incorrect ones just looking at pairs of views. We overcome this problem by evaluating the quality of an entire network of matches. This network is a graph data structure called the model graph (figure 2). It contains a node for each input view and an edge for each pair-wise match.

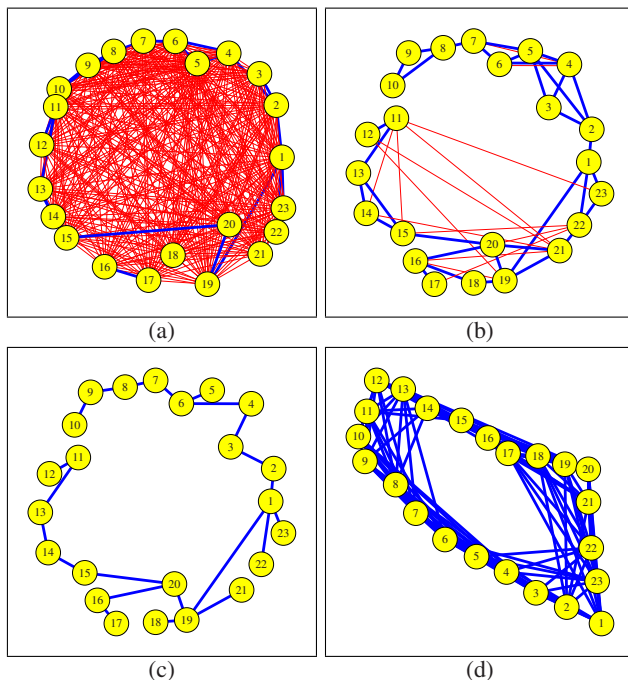


Fig. 2. Coal mine model graphs. Matches are hand-labeled for illustration: thick (blue) edges are correct matches, and thin (red) edges are incorrect matches. (a) The model graph from exhaustive pair-wise registration; (b) After filtering the matches for local consistency (G_{LR}). (c) The solution found by our multi-view surface matching algorithm. (d) The solution with all overlapping views used for verification. Here, the position of the nodes represents the x,y position of each view as determined by our algorithm.

Figure 2-(a) shows the model graph for exhaustive pair-wise registration of the views from the mine data set. Matches are hand-labeled as correct or incorrect for illustration³. The worst pair-wise matches are then removed using a local consistency test (figure 2-(b)). We call this model graph G_{LR} (LR stands for local registration). If we can find a connected sub-graph of G_{LR} that contains only correct matches, it is straightforward to convert the relative poses associated with the matches into absolute poses for each view. We formulate this search as a mixed continuous and discrete

³ A match is defined to be correct if no point in either scan is displaced more than 10% of the scene size from its correct location.

optimization problem. The discrete optimization performs a combinatorial search over the space of connected sub-graphs of the model graph, using a global consistency measure to detect and avoid incorrect, but locally consistent matches. The continuous optimization adjusts the absolute pose parameters to minimize the registration error between all overlapping surfaces, distributing the pair-wise registration errors in a principled way.

Unfortunately, a connected sub-graph containing only correct matches may not exist within G_{LR} . This could happen if one of the input views is corrupted or if some subset of the input views does not overlap sufficiently with any of the remaining views (e.g., if the sensor is moved too far between two scans). We have developed a class of search algorithms that handles these situations gracefully by searching the space of *all* sub-graphs of G_{LR} for the *best* model hypothesis rather than searching only for globally consistent connected sub-graphs.

The output of the optimization is a set of rigid body transforms that aligns the views in a common coordinate system. If the best model hypothesis is not a connected sub-graph of G_{LR} , each component of the sub-graph is a separate part of a multi-part model, and the algorithm outputs transforms that register the views within each part. The redundant surfaces in the registered views can then be merged to form a more compact, unified 3D model.

We have developed several different search algorithms for finding the best model hypothesis within G_{LR} . The method used in this paper is called iterative merging. This algorithm begins with a model hypothesis in which every view is a separate part (i.e., a model graph with a node for each view and no edges). Each iteration of the algorithm merges two parts using the relative pose from a pair-wise match. The chances of merging two parts using an incorrect match are minimized by merging the best-matching parts first and by verifying that all of the views in the merged part are mutually consistent. The model graph produced by the iterative merging algorithm is shown in figure 2-(c) and (d). The corresponding 3D model is shown in figure 3.

4.2 Discussion

Figure 4 shows three close-up views of the 3D model along with photographs taken from the same locations. The details visible in the model highlight the quality possible with high-resolution 3D scanning. In the left picture, a pair of doors with z-braces can be seen. In the center picture, stacks of bricks lie against the walls and on pallets. The partially buried track rails run the length of the corridor and a junction in the rails can be seen in the distance. In the right picture, the rails are more pronounced, and several large pipes are stored on both sides of the corridor.

The difference between the surveyed map of the mine and the actual mine can be significant, as shown in figure 5. The surveyed map is topologically correct, but the scale in parts of the map is significantly off. For example, the trapezoidal column of rock in the center of the surveyed map is actually much narrower near the bottom of the map, which might be a safety concern. The 3D model allows direct measurement of the wall thickness at any point. Furthermore, the 3D model enables measurements

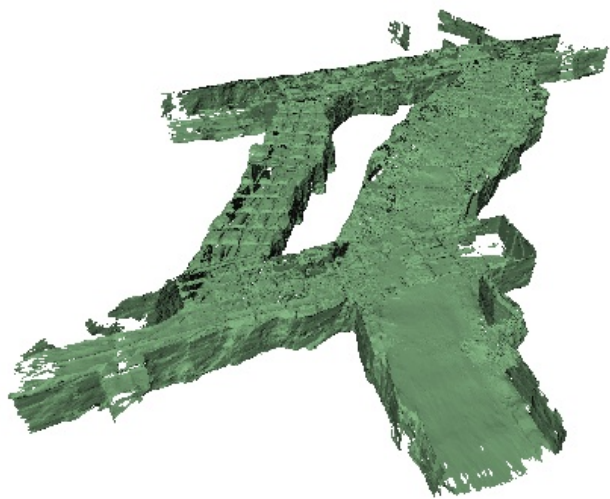


Fig. 3. Perspective view of the automatically constructed mine model. The model consists of 23 separate views covering a region 43x42 m wide and 3 m high.



Fig. 4. Close-up views of the 3D model from various viewpoints (top row) and photographs obtained from the same viewpoints (bottom row).

that are not possible on the 2D map. For example, it is straightforward to compute the ceiling height throughout the mine or to accurately measure the amount of material that has been removed from the mine.

Although we do not have ground truth measurements of the sensor positions, we can still perform some analysis of the accuracy of the model. First, we can tell that the sensor positions are qualitatively correct by looking at cross sections of the model (e.g., figure 5-(b)). If any view were incorrectly registered, some surfaces would not align well with one another and would appear as errant surfaces in the cross sections. The distance between overlapping surfaces gives a quantitative measure of

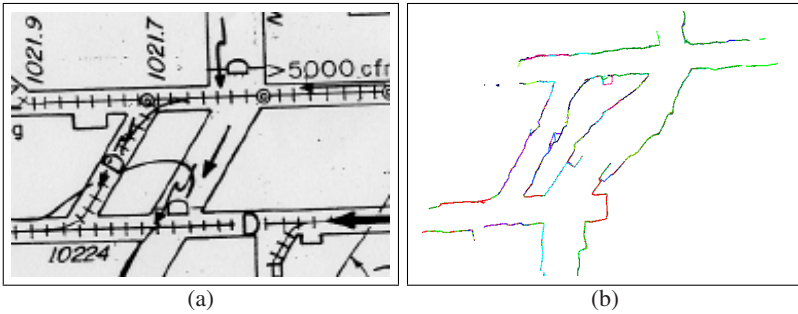


Fig. 5. (a) A close-up view of the map from figure 1-(a) showing the region mapped in our field test. (b) A horizontal cross-section of the 3D model of the corresponding area.

the model accuracy. For this data, the RMS distance between overlapping surfaces was 1.09 cm. Finally, we have performed analysis of other data sets of environments of a similar size for which we measured ground truth positions using a theodolite and fiducials ([4], p. 160). In those experiments, we found the pose error to be less than 0.25% of the model size. The RMS error of the fiducial marker measurements using our algorithm was comparable to the error in the same measurements using bundle-block adjustment on the fiducial points (2.7 cm RMS versus 2.5 cm RMS). This suggests that the primary source of error in that experiment was the measurement of the fiducial position rather than a limitation of our multi-view surface matching algorithm.

5 Laser Sensing in Mine Environments

Coal mine environments present a number of unique challenges for laser sensing systems, including the presence of explosive gas, widely ranging surface albedo, metallic objects, and wet surfaces. In the field test mine, the walls were coated non-uniformly with a white, waterproofing material, and in many places, bare coal was exposed. The roof was reinforced with metallic netting, and the environment contained numerous metallic objects, such as pipes and rails. Furthermore, regions of the walls and ceiling were wet and dripping water. For additional experiments, we collected samples of rocks and bituminous coal for analysis in the controlled environment of our laboratory.

Open beam lasers can be a potential ignition source of methane gas or coal dust, but studies have shown that below 150 mW or 20 mW/mm² methane gas or coal dust cannot be ignited by a laser beam⁴. With an average power of 22 mW [6], the Z+F laser poses no threat.

⁴ NIOSH, May 2003, Laser Safety in Underground Mine, www.cdc.gov/niosh/mining/lasersafety/default.htm

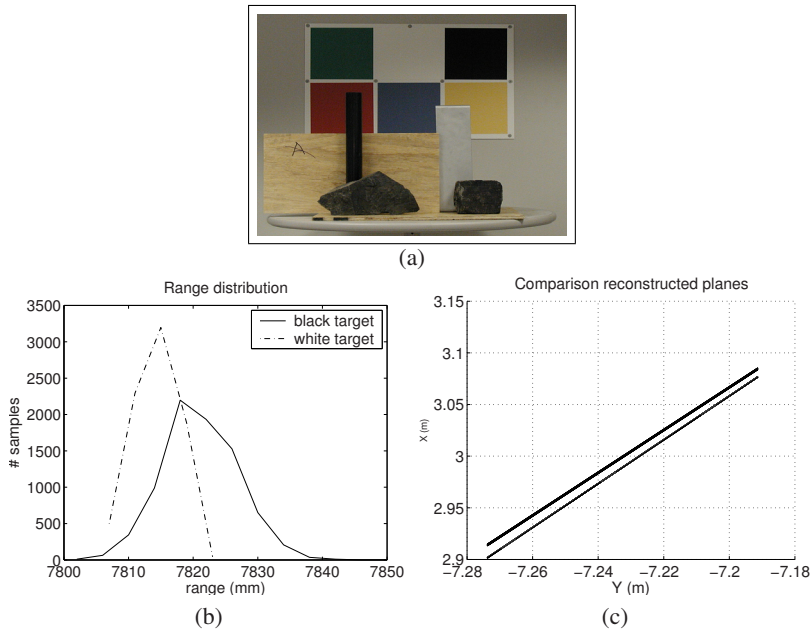


Fig. 6. (a) Photograph of the calibration target with coal samples in the foreground. (b) Distribution of range measurements for the white and black targets. (c) Top view of the two planes fit to the black and white patches showing the reflectance-based range bias.

Our second concern was the level of noise and bias in range measurements when scanning scenes with widely-ranging surface albedo. We analyzed the noise and bias using a calibration target made of 6 different color patches, including black and white (figure 6-(a)). We positioned the laser at 7.5 m from the target and collected 11 identical scans to test the repeatability of the measurement. We measured the range for the pixels within each patch (725 pixels) and computed the mean and standard deviation for each patch over all the scans. Figure 6-(b) shows the distribution of range measurements for the black and white patches. As expected, the level of noise for the black target ($\sigma = 5.80$ mm) is larger than that of the white target ($\sigma = 3.54$ mm); however, even the worst case noise, which occurred with the black patch, was acceptable for a mine-mapping application. We analyzed reflectance-based range bias by estimating the difference in range between the white and black patch. For this experiment, we fit planes to the two patches using the total least squares method. Figure 6-(c) shows a top view of the two estimated planes, which have an offset of 1.3 cm. As with the noise error, this bias is within acceptable limits for mine-mapping.

Finally, we considered the effect of scanning specular targets, such as bituminous coal (which is relatively shiny) or wet surfaces. To test this, we scanned a flat piece of coal twice – once when the sample was dry and again when wet. The sample was positioned at 7.5 m from the sensor and scanned at near-normal incidence. As

Table 1. Reflectance for different targets at 7.5 m

	Coal (dry)	Coal (wet)	Rock	Wood	Aluminum	Black paper	White paper
Min	152	26	291	4413	3330	283	5353
Mean	411	145	491	5663	3698	355	5571
Max	1347	486	754	6709	4022	470	5777

expected the dry sample produced erroneous range measurements associated with specular reflections. Surprisingly, the wetting the coal sample actually reduced the frequency of erroneous measurements. We hypothesize that the reason we did not experience many specular reflections in our field tests is due to the wall-coating and damp environment. Table 1 shows a comparison of reflectance values for several targets scanned at 7.5 m and near-normal incidence, including the wet and dry coal samples.

6 Summary and Future Work

In this paper, we have shown that our automatic modeling-from-reality algorithms can be successfully applied to the problem of high-resolution mapping of underground mines. The model constructed from the 23 scans obtained during our field test was estimated to contain geometric errors on the order of 1 cm. The results of our laboratory experiments indicate that the various sensing challenges presented by the underground mining scenario may introduce error of 1-2 cm into a 3D model. However, it should be noted that these tests are only partially representative because the environment in our laboratory and in the Bruceton coal mine do not fully mimic harsh environment of an active coal mine.

The results of this paper are a proof of concept. The next step would be to further specialize our automatic modeling system for the purpose of mine mapping. First, a ruggedized platform for the system must be developed, either in the form of an electric cart or a tele-operated mobile robot. Second, our automatic modeling algorithms should be modified to operate in an online mode as opposed to the current batch method. The immediate feedback of an online algorithm would enable mine mappers to effectively plan the scan locations. Finally, we are working on new modeling algorithms that scale to very large numbers of views. Our current algorithms have $O(N^2)$ complexity in the number of input views, which limits processing to sub-maps containing about 50 views.

Acknowledgment

We would like to thank the Pittsburgh Research Laboratory of the National Institute for Occupational Safety and Health (NIOSH) for the access to the Bruceton research mine. This research has been supported in part by a fellowship from the Eastman Kodak Company and by the National Science Foundation under grant 0102272.

References

1. C. Baker, Z. Omohundro, S. Thayer, W. Whittaker, M. Montemerlo and S. Thrun, "Case Studies in Robotic Mine Mapping", *International Conference on Field and Service Robotics*, 2003.
2. T. Gibb and D. Hopey, "Quecreek mine accident report blames outdated map", *Pittsburgh Post-Gazette*, November 8, 2003.
3. D. Huber and M. Hebert, "A New Approach to 3D Terrain Mapping", *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, 1999.
4. D. Huber, "Automatic Three-dimensional Modeling from Reality", *Doctoral Dissertation, Carnegie Mellon University*, 2002.
5. D. Huber and M. Hebert, "3D Modeling Using a Statistical Sensor Model and Stochastic Search", *IEEE International Conference on Computer Vision and Pattern Recognition*, 2003.
6. D. Langer, M. Mettenleiter, F. Hartl and C. Frohlich, Imaging Ladar for 3-D Surveying and CAD Modeling of Real World Environments, *International Journal of Robotics Research*, vol 19, no 11.
7. R. Madhavan, M. Dissanayake and H. Durrant-Whyte, "Autonomous underground navigation of an LHD using a combined ICP-EKF approach", *International Conference on Robotics and Automation*, 1998.
8. R. Madhavan, G. Dissanayake and H. Durrant-Whyte, "Map-building and map-based localization in an underground-mine by statistical pattern matching", *International Conference on Pattern Recognition*, 1998.
9. A. Morris, D. Kurth, W. Whittaker and Scott Thayer, "Case Studies of a Borehole Deployable Robot for Limestone Mine Profiling and Mapping", *International Conference on Field and Service Robotics*, 2003.
10. J.M. Roberts, E.S. Duff, P. Corke, P. Sikka, G.J. Winstanley and J. Cunningham, "Autonomous Control of Underground Mining Vehicles using Reactive Navigation", *International Conference on Robotics and Automation*, 2000.
11. S. Scheding, E. Nebot, M. Stevens, H. Durrant-Whyte, J. Roberts, P. Corke, J. Cunningham, and B. Cook, "Experiments in autonomous underground guidance", *IEEE International Conference on Robotics and Automation*, 1997.
12. S. Scheding, G. Dissanayake, E.M. Nebot, and H. Durrant-Whyte, "An Experiment in Autonomous Navigation of an Underground Mining Vehicle", *IEEE Transactions on Robotics and Automation*, vol 15, no 1, 1999.
13. G. Shaffer, A. Stentz, W. Whittaker, and K. Fitzpatrick, "Position estimator for underground mine equipment", *IEEE Transactions on Industry Applications*, Volume: 28 Issue: 5, Sep/Oct 1992.
14. S. Thrun et al., "A System for Volumetric Robotic Mapping of Abandoned Mines", *International Conference on Robotics and Automation*, 2003.

Development of Pneumatically Controlled Expandable Arm for Search in the Environment with Tight Access

Daisuke Mishima, Takeshi Aoki, and Shigeo Hirose

Tokyo Institute of Technology
hirose@mes.titech.ac.jp
<http://www-robot.mes.titech.ac.jp>

Abstract. There is a strong demand for efficient lifesaving techniques and devices in preparation for large-scale earthquakes. We focus on searching survivors and develop the rescue robot "Pneumatic-Drive Expandable Arm." That is an elastic arm type robot driven by pneumatic pressure and has a camera on the head. That can travel stably in the rubble-strewn environment where electric power or wireless communication is not available.

1 Introduction

There is a strong demand for efficient rescue techniques and devices in preparation for large-scale earthquakes. This study aims to develop the robot that focuses on efficient survivor search. This paper reports the Pneumatic-Drive Expandable Arm ("Slime Scope" see Fig. 1), which has a search device, such as CCD camera, at the end of the pneumatically controlled expandable arm.

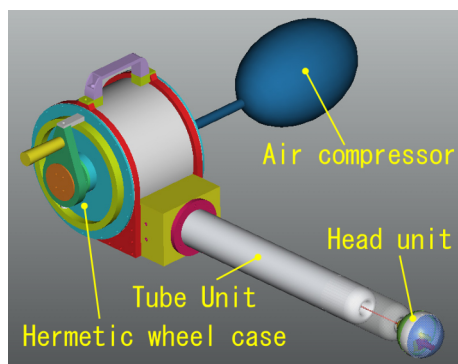


Fig. 1. Slime Scope

2 Conventional Survivor Search

The first step of rescue operation in the rubble is to identify the location of survivors. Conventionally such search mostly depends on the voice of survivors, with occasional

use of rescue dogs. However the voice of survivors are often overwhelmed by the noise of earthmoving machines and helicopters, which significantly reduces the efficiency of rescue operation.

Therefore some new devices have been proposed, such as the one that has a camera at the end of a rod[1] or the one that uses flexible fiber scope[2]. However, they have trouble in the rubble-strewn environment; the former lacks flexibility and thus the search area is limited while the latter is too flexible to go over deep gaps (Fig. 2 , 3). In addition both of them need to be pushed into the rubble that causes large friction and may damage the device. Considering the above, the device to search survivors in the rubble needs following properties:

- Flexibility that allows the device to go inside along the rubble
- Rigidity to go over gaps and holes
- Minimum friction resistance with the rubble

In this study, we developed a new search device called Distal Expandable Tube (DETube), which meets these criteria.

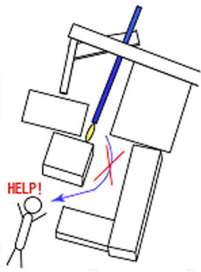


Fig. 2. Search in the rubble (rod-type device)

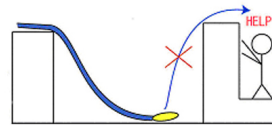


Fig. 3. Bridging a gap (fiber scope)

3 Distal Expandable Tube

3.1 Principle of DETube

Shown below is how the DETube works.

1. Attach a tube made of airtight, flexible and inextensible material to the hermetic case. Then the end of the tube is closed to make a sack. (Fig. 4 :1)
2. Tuck the tube inside. (Fig. 4 :2)
3. Put air in the tube. (Fig. 4 :3)
4. The tube tucked inside extends. (Fig. 4 :4)

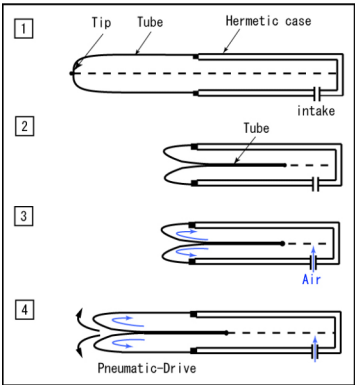


Fig. 4. Principle of DETube

3.2 Characteristics of the DETube

One of the major characteristics of the DETube is that it can expand without causing friction with the outside. That is because, as we mentioned before, it expands by dispensing the tube from inside, so the part once extracted outside stands still against the outside environment.

Another major characteristics is that it uses a pneumatic tube for expandable unit. Because of this, it can bend easily and, with a direction instruction device installed at the end, go along the rubble (Fig. 5). On the other hand, the tube can become more rigid by increasing the inner pressure to go over a ditch (Fig. 6). That is, DETube can serve as a new mechanism to go through the rubble that meets the criteria required for devices that operate in the rubble as described in Chapter 2 (Fig. 7).

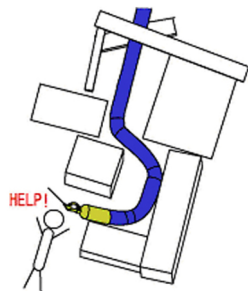


Fig. 5. Search in the rubble (Slime Scope)

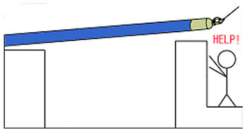


Fig. 6. Bridging a gap (Slime Scope)

4 Development of Slime Scope

We developed the "Slime Scope," an expandable search device, using the DETube we described in the previous section.

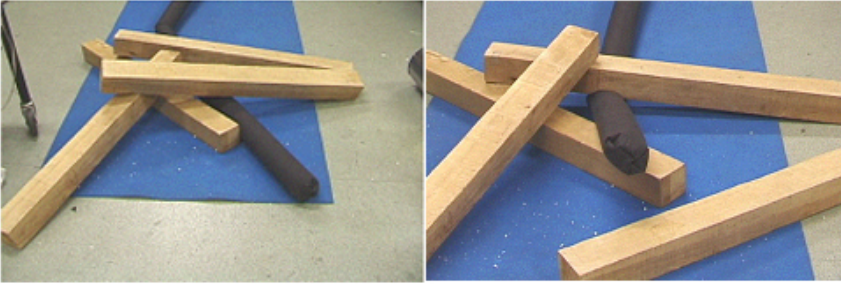


Fig. 7. Search in the rubble using DETube

The Slime Scope is composed of four major units, namely 1) air compressor that supplies air, 2) tube unit that expands using the DETube mechanism, 3) the head unit that has the search device and 4) the hermetic wheel unit which controls the amount of expansion. Shown below are the details of each unit.

4.1 Air Compressor

The test machine uses an electrically operated air compressor to supply air. However, an electric air compressor is not suitable for actual rescue operation since it is too heavy to carry around and electricity is often not available at disaster sites.

Therefore we are planning to replace the current electric air compressor with a man-powered pump in the future. To be more specific, we will use the man-power extracting device, which was jointly developed by the Kanagawa Industrial Technology Research Institute and our laboratory (Fig. 8). This device is a foot pump. When using this device for the Slime Scope, an accumulator and a pressure-reducing valve will be installed between them.

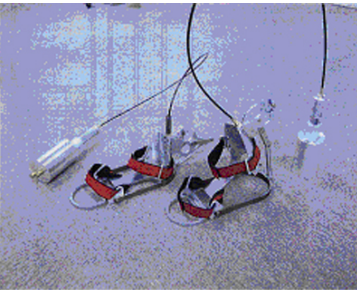


Fig. 8. Man power extracting device

4.2 Tube Unit

The tube unit shall be made of airtight and flexible material that can also bear the inner pressure. We used Kevlar fiber lined with urethane rubber that is not expansive. The model we developed this time has a hose that is 80 mm in diameter and 0.5 mm in thickness and 5 m in length. The tube specifications are shown Fig. 9 , 10 .

Table 1. Tube unit specifications

Diameter of Tube	80 [mm]
Length of Tube	3000 [mm]

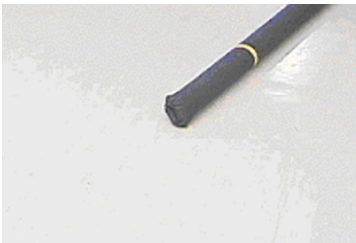


Fig. 9. Tube unit (before expansion)



Fig. 10. Tube unit (after expansion)

4.3 Head Unit

As shown in Fig. 11 , the head unit and tube unit are fixed by the power supplying wire inside of the tube. Therefore, when the tube expands at the velocity u , as shown in Fig. 11 , the inner wire will come out at twice that speed, $2u$. As a result, extra wire will pile up at the end. To make the head unit follow the tube, it is necessary to wind up the excessive wire length. Therefore we added the wire-rewinding device to the head unit (Fig. 12 , 13).

The specifications of the head unit are shown Table 2 .

Table 2. Head unit specifications

Diameter	90 [mm]
Mass	400 [g]

4.4 Hermetic Wheel Case

As shown in Fig. 14 , 15 , the hermetic case is composed of the following:

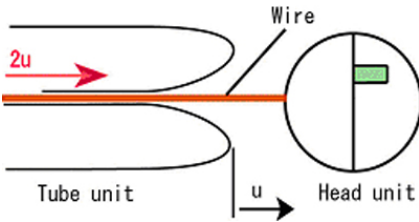


Fig. 11. Connection of head unit

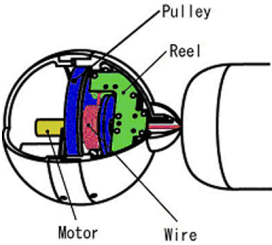


Fig. 12. The mechanism of Head unit

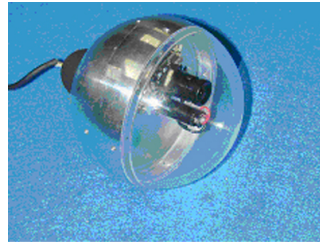


Fig. 13. Head unit

- a winch
- slip ring
- an air tank (Hermetic case)
- lock mechanics

The winch winds up the tube and wire simultaneously. The slip ring allows supplying electricity to the outside through the wire wound up around the winch. The air tank contains the air supplied from the air compressor. The amount of tube expansion is controlled by the handle and lock mechanics.

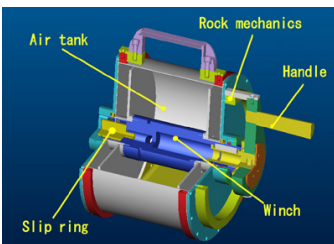


Fig. 14. Hermetic wheel case

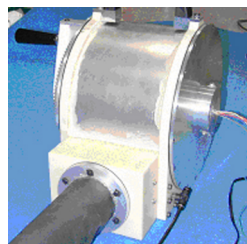


Fig. 15. Hermetic wheel case

This Hermetic Wheel case can wind up up to 10 m of tube. Since the wire is inside of the tube and will be wound up along with the tube, the wire will twist if no measure is taken. The slip ring prevents this twist.

The next section explains the handle lock mechanism. When air pressure is applied, the tube will expand in forward direction. Accordingly, it will continue

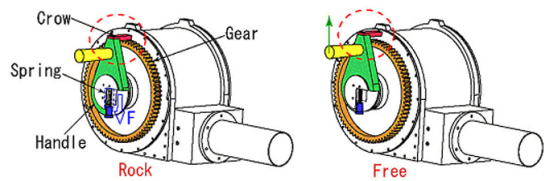


Fig. 16. The principle of Lock mechanics

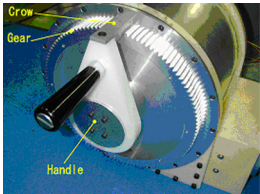


Fig. 17. Lock mechanics

expanding unless the expansion is stopped. In rescue operations, however, sometimes it is necessary to stop tube expansion and search a certain area for a while. Therefore, the device controls the expansion amount by locking the handle to stop tube expansion. Fig. 16 , 17 shows the actual handle lock mechanism. The Slime Scope controls the tube expansion by locking and releasing the handle.

5 Test Machine

Fig. 18 shows our test machine. We used this test machine to confirm that the DETube could travel in the rubble.

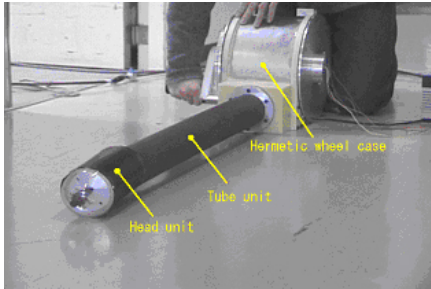


Fig. 18. Test machine

Table 3. Test machine specifications

Total mass	10.4 [Kg]
Length of Tube (Max)	1980 [mm]
Length of Tube (Min)	480 [mm]

6 Characteristics of the DETube

6.1 The Holding Power of the DETube

The DETube's holding power means the power to maintain tube expansion against the propelling force at the head of the DETube. Here the propelling force means the force that let out the tube forward at the end of the DETube. Assuming that propelling force produced by the DETube is F_T , the holding power is F_k , the force that holds the tube from outside is F_f and the loss is F_L , the relationship of these can be expressed as Equation (1). Here the loss means the friction resistance between the external tube and retracted tube and so on.

$$F_T = F_k + F_f + F_L \quad (1)$$

As the DETube's propulsion principle is the same as the principle of a pulley, the following equation is established.

$$F_k = F_f \quad (2)$$

Substituting Equation (2) in Equation (1) yields the following equation.

$$F_T = 2F_k + F_L \quad (3)$$

Here F_T is the product of the pressure inside of the tube and the stress area, so if the holding force F_k is determined, impact of the loss in propelling force can be calculated from Equation (3). In other words, determining the holding power is considered to be important to know the characteristics of the DETube.

Therefore, we calculated the loss in propelling force by measuring the holding power F_k using the experiment device shown in Figs. 19 . Fig 20 shows the result.

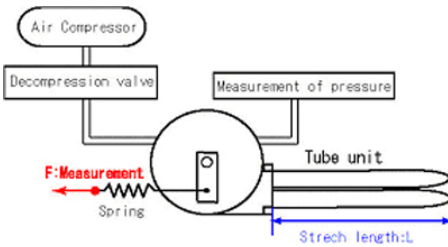


Fig. 19. Experiment apparatus

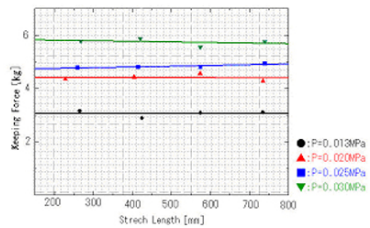


Fig. 20. Result

The result demonstrated that the holding power was not the same as the propelling power. This discrepancy results from the friction loss. Further it was confirmed that the higher the pressure, the larger the impact of the loss became. It is possibly because higher pressure increased the friction resistance inside the tube and thus increased the loss.

In addition, Fig. 20 shows that the holding power remains almost the same regardless of the expansion amount. In other words, the impact of the loss is independent of the expansion amount. However, in theory, as the expansion amount becomes larger, the contact area between the external tube and retracted tube, namely, the area where loss from friction occurs, will increase. As a result, the loss must be larger. The reason of this difference is assumed that since the expansion amount was small (max. 800 mm), the loss from friction resistance was also small.

6.2 Bending of the DETube

In the previous experiment, we studied the impact from the loss by measuring the holding power when the DETube went straightforward. However, in actual search operations in the rubble, the DETube will often need to bend to go through, rather than go straightforward. Accordingly, we studied the changes of the loss when DETube bends, using the experiment apparatus shown in Fig. 21 . Fig. 22 shows the result.

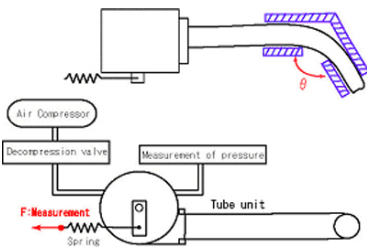


Fig. 21. Experiment apparatus

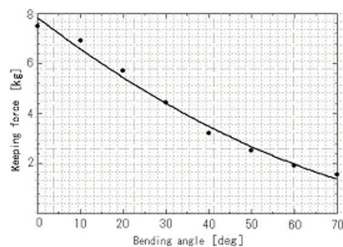


Fig. 22. Result

Fig. 22 shows that the holding power decreases as the bending angle increases. It suggests that increase of bending angle results in increased in loss.

Then we studied the loss. When the DETube is bended, the contact area between the external tube and retracted tube will increase and the area affected by sliding friction will increase. Therefore, it is considered that the loss will increase.

To study the impact of the friction loss, we repeated the previous experiment once again after applying grease inside of the tube for lubrication. Fig. 23 shows the result.

Fig. 23 shows that the holding power is larger, that is, the loss is smaller when the lubricant is applied to the inside of the tube compared to when the lubricant is not applied. It confirmed that the loss from the friction force is a major factor that decreases the holding power.

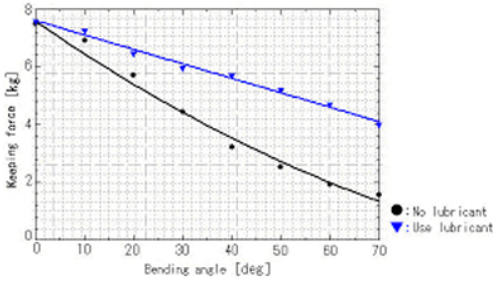


Fig. 23. Result

7 Conclusion

Shown below are the conclusions of this study.

(1) We investigated the conditions necessary for survivor search in the rubble and proposed the DETube, a new rubble traveling unit that meets the above conditions.

(2) We designed Slime Scope, a new search device that uses DETube and created a test machine.

(3) We focused on the DETube's holding power and studied the loss through experiments for both when DETube was straight and bended.

Our future tasks include experiments that assumes actual disaster sites and improvement of the machine.

References

1. Nippon Tsushin Service Co., Ltd.: Product catalogue, color video camera system for survivor search, Bokame
2. Olympus, disaster rescue system, pipe camera
3. The Rescue Robot Equipment Study Group: Actual Condition Survey Committee of Rescue Operation on the Kobe Earthquake, Robotics & Mechatronics Division, The Japan Society of Mechanical Engineers (1997)
4. Shigeo Hirose, Tetsuo Hagiwara, Kenichi Abe: Development of Man Power Extracting Machine and Rubble Removal Machine, Report of 1999 Study on Enhancement of Disaster Rescue Operations, 65/71 (2000)

Development of Mobile Robots for Search and Rescue Operation Systems

Akihiro Ikeuchi¹, Toshi Takamori¹, Shigeru Kobayashi²,
Masayuki Takashima¹, Shiro Takashima¹, and Masatoshi Yamada²

¹ Dept. of Computer & Systems Eng.
Kobe Univ.

ikeuchi@r.cs.kobe-u.ac.jp, takamori@r.cs.kobe-u.ac.jp,
takasima@r.cs.kobe-u.ac.jp, s_takashima@r.cs.kobe-u.ac.jp
<http://www.r.cs.kobe-u.ac.jp/>

² Dept. of Mechanical Eng.
Kobe City College of Tech.
kobayash@kobe-kosen.ac.jp, r202109@kobe-kosen.ac.jp
<http://www.kobe-kosen.ac.jp>

Abstract. This paper proposes a sufferers searching system using the group of robots to find sufferers at debris as quickly as possible in urban disaster. Five kind of new robots (Series UMRS-V) have developed as the searching robots and their hardware and software systems included the feature and mechanism, sensor system, data processing and control system have made clear. The human interface and simulator system have also developed to make the communication between robots and operator easy and to study the searching algorithm etc.

1 Introduction

In Japan, we have many natural disasters every year, and some of them are huge ones like an earthquake, a flood, and an eruption. Among of them, The Great Hanshin-Awaji Earthquake happened in 1995 was serious, and over 5000 peoples were killed even hard human rescue operations were made. Also in the world, numerous numbers of victims were killed in natural and man-made disaster, like in the terrors at New York in United States and the huge earthquake in Turkey. Many of those victims were closed and pinched in half corrupted buildings, and it took too much time to find them[1]. We could have saved much life if we had made a much quicker rescue operation with sufficient safety support and protection for rescuers.

These experiences tell us that the sufferer search activity is top priority as the first step of the rescue operation. To make an efficient and wide area search operation in short time, we are developing a search system with the remote controlled group of robots operated by a few people. So far, we proposed the crawler type robots, UMRS-IV; Utility Mobile Robot for Search-IV, and their control system to find sufferers in debris, and the study and development of them were made. Based on the knowledge obtained by the development of previous type robots and the RoboCup-Rescue 2002 competition games, we describe the development and evaluation of UMRS-V series with new mechanism and search system in this paper.

1.1 Outline of Rescue System in Wide Disaster Area

In huge disaster, the hit area is wide and we have to make a rescue operation at many scattered spots. In addition, usual traffic and communication resources are completely down and the number of people who can make a rescue activity is limited. To take a quick and effective action in these difficult circumstances, the search system has to be independent from traffic and communication infrastructures that supposed to be no use in disaster. And few field operator with many robots search sufferers in half destroyed buildings at each disaster site. Fig. 1 indicates the whole image of the search system that we set as the target in this study.

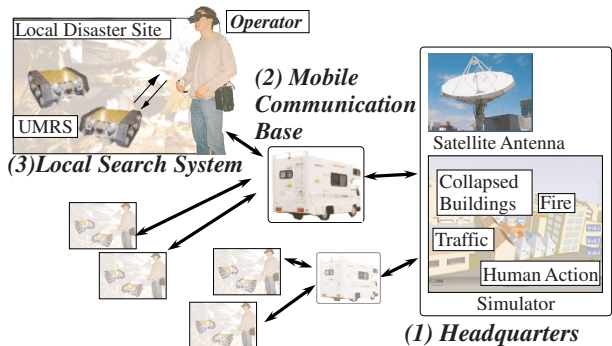


Fig. 1. A Whole Image of Rescue System

1.2 Local Search System with UMRS

This is the part indicated “(3) Local Search System” in Fig. 1. To identify the specific position of buried sufferers quickly, the sufferer searching system with many small search robots, UMRS is currently under development.

The wireless communications is made between the computer for operator’s manipulation and UMRSs, and among UMRSs, so a command, a sensed data and a camera view are transmitted. This is described in detail in the Section 2 as shown in Fig. 3. The operator, in relatively safe place outside of closed search area, manipulates UMRS and makes them go into a searching place. Not only the man-machine control for each UMRS but also the controls for the group of robots are conducted. The operator knows the condition of searching area by using mainly the camera, makes an important judge like confirmation of a human body, and carry out the search operation under the cooperation with autonomous movement of UMRS.

Some special driving mechanism is required for a robot to run into the unsafe and bad road surface condition area like rooms in half corrupted buildings. Also the sensor system to know the condition of the search area, and the communication system to transmit those sensed data to the operator are necessary to furnish. Section

2 expresses the construction of our proposed and current developing robots, UMRS-V.

To realize the man-machine control system that contains the transmission system of sensed data in searched area to operator and the decision making system to whole group of robots and to each robot, the human interface that gather and display the information to the operator and give the necessary command to the robots is required. Section 3 reports the human interface that has the map generation function useful for rescue action afterward.

To make few operators possible to control the group of robots and make a useful search, robots have to be move autonomously by the algorithms that cooperates with the commands from operator. The simulator for search is developed to evaluate and develop these algorithms. Also we use this simulator to evaluate the human interface. Section 4 describes some developments using this simulator.

2 Utility Mobile Robot for Search Series V “UMRS-V”

2.1 The Feature and Mechanism

Based on the knowledge obtained through the development of previous rescue search robot series, new 5 different kinds of search robots for rescue UMRS-V are currently developed as follows. The design specifications required to these robots series is shown in Table 1.

Table 1. The requirement of UMRS-V

Size,Weight	Less than UMRS-IV (Previous developed robots)
Robustness	It doesn't break even if it falls from the height of 50 cm
Structure	Reversible 65mA
Driving Speed	More than 50 cm/sec at flat space
Climbing ability	Incline angle of slope; More than 35 deg.
Duration time of drive	3 hours
Control system	Half-autonomy
Lighting	More than 100 Lux at 5m in front of the robot

A robot with auxiliary crawler (UMRS-V-M1) is developed as a successor of previous robot series UMRS-IV. Its outlook is shown in Fig. 2. The body structure is revised version of UMRS-IV, and taken some countermeasures to the involution of paper and

codes, that is the problem became obvious in the experiment of RoboCup-Rescue, the improvement of the robustness of the body when it drops, and so on. The body size is $600 \times 520 \times 180$ [mm] (when auxiliary crawlers are retracted), and weight is 21.4 [kg].

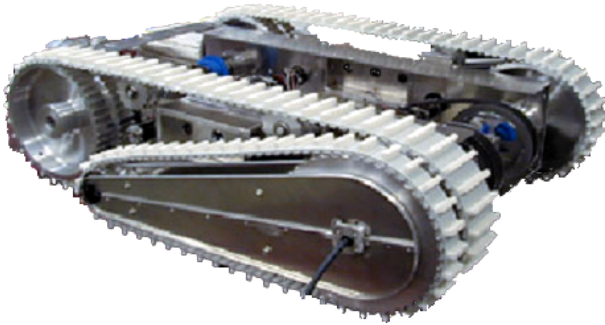


Fig. 2. The Photo of UMRS-V-M1

SGI No.1 robot (UMRS-V-S1) is lightweight and compact. The body size is $360 \times 390 \times 155$ [mm] (when auxiliary crawlers are retracted), and weight is 7.8 [kg]. The main purpose of this robot is to evaluate the operation and control system for further development of a smaller robot than current ones. Auxiliary crawlers are longer than main crawler, and the right and left ones can be swung separately.

A robot with the center of gravity movement mechanism (UMRS-V-M2) is able to control the position of center of gravity to improve the ability getting over obstacles by moving the weight forth and back direction at the bottom of the robot body. The robot size is $590 \times 400 \times 165$ [mm], and total weight is 20.3 [kg]. Because of the heavy weight moving the center of gravity, it is expected that the duration search time is limited.

Rotation robot with triangle shape crawlers (UMRS-V-M3) is currently under assembling. To improve the ability of obstacle overcoming, the drive system with 4 independent rotatable crawler wheels is designed for evaluation purpose. Based on the experience of this robot, improved robot for search will be produced.

SGI No.2 robot (UMRS-V-S2) is the advanced one of SGI No.1 robot (UMRS-V-S1). Higher performance model is aimed. The crawlers for running and auxiliary crawlers are made as almost same length and same shape.

2.2 Sensing, Processing, and Control

Sensor System Sensors installed in UMRS-V are as follows.

Positoining Sensors: The dead reckoning method is hired to measure the position and orientation of robot and a gyroscope and encoders are used. The gyroscope is 3-dimensional motion sensor, MDP-A3U7 by NEC Tokin. It has 3 axis ceramic gyroscopes, 2 axis acceleration sensors and 2 axis geomagnetic sensors. 3 directional orientation angles are calculated by multiply the correction values. Linux version device driver and library are currently developed as an open source.

Obstacle Detection Sensors: Infrared distance measurement sensors are used to detect obstacles. This sensor outputs the analog voltage in accordance with the distance within the detection angle. And this is compact and not expensive, and then we can install many this sensors to the robot. Using this sensor, the position of obstacle can be plotted in the map by comparing with and over wrapping the position of robot.

Human Detection Sensors: Pyroelectric IR sensor and CO₂ sensor are used to detect the mankind. Pyroelectric IR sensor can recognize the human in motion by detecting the variation of temperature, but cannot response to the people in rest. The output can be obtained by installing a shutter in front of the sensor lens for correction and forced temperature change of detection obstacle was made. The detection distance is 1200 cm and the detection angle is about 5 degree. CO₂ density sensor is TGS4161 by Figaro Engineering Inc. The CO₂ density around the sufferers seems to be high by their breathing. Using this sensor, we can judge whether human life is near or far. The accuracy can be maintained by using the sensor fusion system that consists of CO₂ sensor, pyroelectric IR sensor, thermometer and so on, even something is burning inside the room. In this searching system with robots, the final confirmation of sufferer is not by the automatic pattern matching, but by operator's decision mainly by monitoring the camera on the robot as mentioned in next section.

Camara: The information from camera view is the most important to operators and the hugest amount in sensor system. The camera view is transmitted to the operating monitor system as a human interface and the operators finally confirm bodies of sufferer, their position and orientation from operators and the environment where they lie. The camera installed in UMRS-V series is iBot by Orange Micro Inc. connectable to IEEE1394 interface and easy to correspond H.323 movie streaming protocol.

Data Processing and Control system The outline of the data processing and control system of UMRS-V is shown in Fig. 3. This is the basic software and hardware units commonly used in every robots in UMRS-V series, and the proper function of each robot can be added on this basic unit.

UMRSs and the operator have computers, and the commands and the information are received and transmitted between these computers by IEEE802.11b wireless LAN. And the necessary data are received and transmitted between the computer and the motor controller and I/O board on the robot based on the command received from

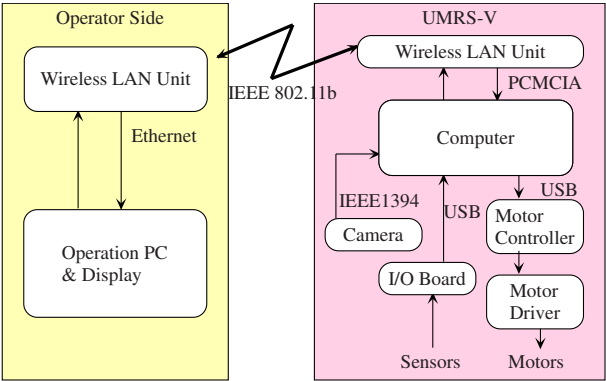


Fig. 3. The Block Diagram of Data Processing on UMRS-V and Operator's Computers

the operator's computer, then the motors are controlled and sensor data are collected. Also the computer on the robot works not only as the relay of the controller's commands but also as the controller of automatic obstacle avoidances and so on if necessary.

The camera images are collected into the computer on the robot by IEEE1394 interface, and transmitted to the operator's computer by H.323 protocol. H.323 provides the real time streaming and the compression of pictures on time and space, and are used in the remote meeting system via the Internet. So it seems to be suitable for the system that the transmission of clear pictures as real time as possible in limited data bandwidth is necessary.

3 Human Interface

3.1 System Configuration

The system requires the computers that the operators can grasp the situation of search by gathering the information from UMRS and gives the instructions to the group of UMRS. For this purpose, even though we can choose from various kinds of computers like a notebook computer and a wearable computer, the computer with Windows OS is adopted because an ordinary operator can easily catch on the way to manipulate, the software works on this OS is developed. The configuration of the software in the operation computer is shown in Fig. 4. Each component is described as follows,

Human Interface receives Operator's input and displays the information, and explained in detail at 3.2.

Command Generator produces the proper commands to manipulate UMRS. The input is the situation of the search by operator's commands and the data, the concrete actions each UMRS is decided based on the search algorithm.

Data have their construction to store information, and generate the search map and

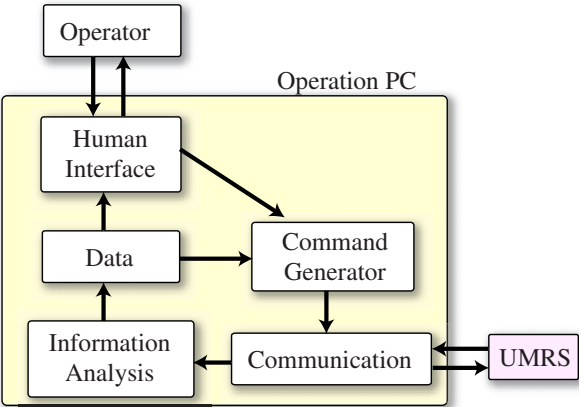


Fig. 4. Software Configuration in Operation Computer

keep the condition of UMRS. A 3-dimensional space is basically, divided to the $0.1[m^3]$ cubic grids, the information of each grid is stored, and each grid is classified as not searched area, searched area, moved trace of UMRS, obstacle.

Communication establishes the communication with UMRS by the protocol and control the amount of data communication.

Information Analysis analyzes the data transmitted from UMRS, transforms to meaningful data and store in Data. For instance, the current position of UMRS is calculated from the values of a gyroscope, encoders and stored ones in Data. And the position of the obstacle is computed by the combination of this current positioning information and the outputs of distance measurement sensors.

3.2 Outline of Operation

Using a mouse and a keyboard does the instruction to the UMRS in search. The cursor keys, forward and backward movements by up and down keys, and the left and the right full turns by left and right keys do the man-machine control. And the mouse does other operation basically.

The operating display of this search system is shown in Fig. 5. 2 windows on the left side of the display monitor the camera pictures from UMRS. This photo is one of UMRS-IV's, and displayed in the application of the web browser. The system for UMRS-V is under development. There is no major change as the human interface. The window on the right side of the display is the map for search, and indicates the search condition. Even current map is 2-dimensional, now updating to 3-dimensional one. There is a window indicating the condition of UMRS at lower right of the search map window, the orientation angles of UMRS and the enlarged condition around UMRS are animated on this window. The search map is explained in detail at the section 3.3.



Fig. 5. The Screenshot of Operation Interface

3.3 3-Dimentional Search Map with Landmarks

Various kinds of information obtained by the sensors of robots are integrated by time series, and as a consequence, the meaningful information is displayed on the 3-dimensional maps graphically in real time. The present type of this map is 2-dimensional, and shown in right side of Fig. 5.

This map indicates not only the whole shape of the searching area and the present positions of each robot, but also the places where possibly sufferers are buried. Then the operator and observers can take the condition of a search as a bird's-eye view, and this condition makes the operator possible to make a quick and accurate instruction to the group of robots. Also the man-machine control for robot itself is possible by using this one with pictures of the camera.

This map is useful not only for a sufferer searching actions but also the rescue activities after find the sufferers. After identifying sufferer's position, the rescue team have to go into the debris or dredge up the places near by to pick up them. The maps required at this stage is as follows,

- The maps indicate the sufferer's position in the absolute coordinates.
- The maps guide the rescuers the route to follow.
- The maps gave us the information in debris like the safety condition.

To confirm the conditions in the debris at a glance, we propose the maps with attachments objects as easy-to-recognize landmarks like desks, doors and so on. Rather than drawing the shape of the wall at small area in detail, it seems to be more effective and useful to show the route by putting the simplified landmarks on the map in searching area.

A few kinds of objects as easy-to-recognize landmarks are prepared as a template, and the operator places one and another chosen from these templates. Based on the

camera’s pictures with considering the current UMRS’s position stored in Data of Fig. 4 and the output values of the distance measurement sensors, the operator decides the positions to place. The object is recognized not by a machine with some kind of the pattern matching methods but by a human with the assistance of camera’s monitor views. This means the real objects must be recognized in the same way by rescuers who enter the area where the robots already searched.

4 Simulator System

A simulator has been developed for studying the searching algorithm for the group control of UMRSs. As follows, it is useful too as the evaluation system for each subsystem in this rescue operation system.

4.1 Configuration and Function of Simulator

As illustrated in Fig. 6, the simulation system consists of various types of subsimulator. Among them, Simulated World and UMRS Program are two major ones. The UMRS Program is an program that is currently installed in the PC of UMRSs, so this simulator enables us easily to develop the robot control program for searching operation without any real robot’s experiments. The Simulated World defines a searching space to be simulated and realizes a virtual world transformable dynamically.

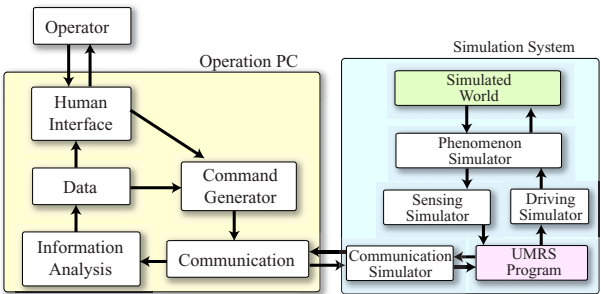


Fig. 6. The Structure of Simulation System

4.2 Application of Simulator

Development of Search Algorithm The search algorithm is calculation to determine how the robots act autonomously under the reflecting recognitions and directions of operator. For example, the information obtained by the CO₂ sensor determines to go to the area in which a sufferer is possible to exist, and makes the route plan to reach there. In simulator, a virtual map is prepared to represent the test field, the developed algorithm is tested and evaluated by a benchmark condition.

Development of Human Interface The human interface is developed to display the condition or state of a searching area and the robots, and to input the operator's commands. This simulator also should have a human interface system. So this simulator enables to evaluate and develop the human interface.

5 Summary and Conclusions

Based on the necessity of the rescue system in a huge disaster, the system and tactics is suggested that the rescue system in a local disaster site is conducted quickly and effectively. To realize the actual rescue system, the system with many small size robots manipulated by one operator is proposed, and the trial system is manufactured. Based on the experiment and basic data of searching robots previously developed, 5 units of new robots as the series of UMRS-V are now under construction. The human interface is made experimentally to operate the searching system with many robots. And as a part of this development, 3-dimensional map with the landmarks is proposed and developed to utilize the rescue operation performed after the searching. The simulator for our search system with the group of robots is developed, the evaluation and development using this simulator is explained.

We plan to improve UMRS and the human interface, and develop the mounting of the simulator, the communication system effective in debris, the sensor system and recognition system for human body detection and environment recognition, the algorithm for searching.

Acknowledgement

This research is supported by the Special Project for Earthquake Disaster Mitigation in Urban Areas, Development of Advanced Robots and Information Systems for Disaster Response of Ministry of Education, Culture, Sports, Science and Technology Japan.

References

1. T. Takamori et al., "Actual Condition Survey on Lifesaving at The great Hanshin Awaji Earthquake for Study and Development of Rescue Robots and their Parts.," *Report of Study Group*, Dept. of Mechatronics, Jpn Soc. Mech. Engng, 1997.
2. T. Takamori et al., "Development of Sufferer Searching System by Mobile Robots," *Report of Study Group on Development of Rescue Robots in Huge Disaster RS150 (in Japanese)*, Jpn Soc. Mech. Engng pp.149-196, 1999.
3. Shigeru Kobayashi and Toshi Takamori, "A Human Body Search System by a Man-Machine Controlled Group of Robots in a Rescue Operation," *Advanced Robotics*, Cutting Edge of Robotics in Japan 2002 Volume 16 Number 6, pp.525-528, 2002.

Distributed Search and Rescue with Robot and Sensor Teams

George Kantor¹, Sanjiv Singh¹, Ronald Peterson², Daniela Rus², Aweek Das³,
Vijay Kumar³, Guilherme Pereira³, and John Spletzer³

¹ Robotics Institute
Carnegie Mellon University
ssingh@cmu.edu
<http://www.frc.ri.cmu.edu>

² Computer Science Department
Dartmouth College
rus@cs.dartmouth.edu
<http://www.cs.dartmouth.edu>

³ GRASP Laboratory
University of Pennsylvania
kumar@cis.upenn.edu
<http://www.grasp.upenn.edu>

Abstract. We develop a network of distributed mobile sensor systems as a solution to the emergency response problem. The mobile sensors are inside a building and they form a connected ad-hoc network. We discuss cooperative localization algorithms for these nodes. The sensors collect temperature data and run a distributed algorithm to assemble a temperature gradient. The mobile nodes are controlled to navigate using this temperature gradient. We also discuss how such networks can assist human users to find an exit. We have conducted an experiment at a facility used to train firefighters in order to understand the environment and to test component technology. Results from experiments at this facility as well as simulations are presented here.

1 Motivation

We consider search and rescue applications in which heterogeneous groups of agents (humans, robots, static and mobile sensors) enter an unknown building and disperse while following gradients of temperature and concentration of toxins, and looking for immobile humans. The agents deploy the static sensors and maintain line of sight visibility and communication connectivity whenever possible. Since different agents have different sensors and therefore different pieces of information, communication is necessary for tasking the network, sharing information, and for control.

An ad-hoc network is formed by a group of mobile hosts upon a wireless local network interface. It is a temporary network formed without the aid of any established infrastructure or centralized administration. A sensor network consists of a collection of sensors and distributed over some area that form an ad-hoc network. Our heterogeneous teams of agents (sensors, robots, and humans) constitute distributed adaptive sensor networks and are well-suited for tasks in extreme environments,

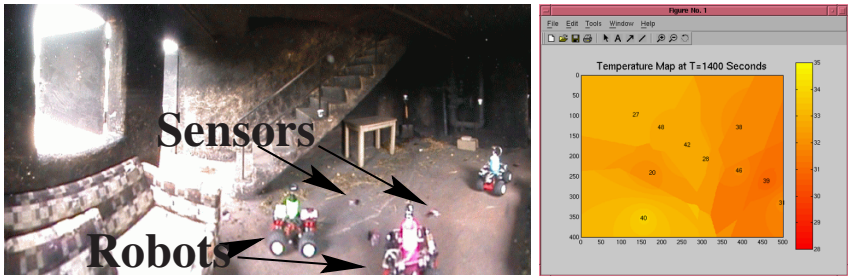


Fig. 1. (Left) An ad-hoc network of robots and Mote sensors deployed in a burning building at the Allegheny Fire Academy, August 23, 2002 (from an experimental exercise involving CMU, Dartmouth, and U. Penn). (Right) The temperature gradient graph collected using an ad-hoc network of Mote sensors.

especially when the environmental model and the task specifications are uncertain and the system has to adapt to it. Applications of this work cover search and rescue for first responders, monitoring and surveillance, and infrastructure protection.

We combine networking, sensing, and control to control the flow of information in search and rescue in unknown environments. Specifically, this research examines (1) localization in an environment with no infrastructure such as a burning building (for both sensors and robots) (2) information flow across a sensor network that can localize on the fly for delivering the most relevant and current information to its consumer, maintaining current maps, and automating localization; (3) using feedback from the sensor network to control the autonomous robots for placing sensors, collecting data from sensors, and locating targets; and (4) delivering the information gathered from the sensor network (integrated as a global picture) to human users. The paper will detail our technical results in these 4 areas and describe an integrated experiment for navigation in burning buildings.

2 Localization

Localization in dynamic environments such as posed by search and rescue operations is difficult because no infrastructure can be presumed and because simple assumptions such as line of sight to known features cannot be guaranteed. We have been investigating the use of low cost radio beacons that can be placed in the environment by rescue personnel or carried by robots (see Figure 2). In this paradigm, as the robot moves, it periodically sends out a query, and any tags within range respond by sending a reply. The robot can then estimate the distance to each responding tag by determining the time elapsed between sending the query and receiving the response. The advantage of such a method is that it does not require line of sight between tags and the mobile robot, making it useful in many environmental conditions where optical methods fail. Note that, since each tag transmits a unique ID number, distance readings are automatically associated with the appropriate tags, so the data



Fig. 2. A radio tag, approximately 12 × 9 cm in size, with which a robot can communicate to obtain range data. Such tags can be scattered into a burning building as firefighters move about or even be deployed by robots themselves. Since these tags are placed without careful survey, their position must be calculated along with the position of the mobile agents themselves.

association problem, a difficult issue in environments that can be visually obscured, is solved trivially.

Since the position of the tags is unknown to start and can potentially change during operation, it is necessary to localize both the receiver and the beacons simultaneously. This problem is often known as Simultaneous Localization and Mapping (SLAM). Although generally it is assumed that a receiver is able to measure both range and bearing to "features", we can assume only that range to tags is known and that this measurement may be very noisy. We have adapted the well-known estimation techniques of Kalman filtering, Markov methods, and Monte Carlo localization to solve the problem of robot localization from range-only measurements [1] [5]. All three of these methods estimate robot position as a distribution of probabilities over the space of possible robot positions. In the same work we presented an algorithm capable of solving SLAM in cases where approximate a priori estimates of robot and landmark locations exist. The primary difficulty stems from the annular distribution of potential relative locations that results from a range only measurement. Since the distribution is highly non-Gaussian, SLAM solutions based on Kalman filtering falter. In theory, Markov methods (probability grids) and Monte Carlo methods (particle filtering) have the flexibility to handle annular distributions. Unfortunately, the scaling properties of these methods severely limit the number of landmarks that can be mapped. In truth, Markov and Monte Carlo methods have much more flexibility than we need; they can represent arbitrary distributions while we need only to deal with very well structured annular distributions. What is needed is a compact way to represent annular distributions together with a computationally efficient way of combining annular distributions with each other and with Gaussian distributions. In most cases, we expect the results of these combinations to be well approximated by mixtures of Gaussians so that standard techniques such as Kalman filtering or multiple hypothesis tracking could be applied to solve the remaining estimation problem.

We have also extended these results to deal with the case when the tag locations are unknown using a geometrically inspired batch processing method. The basic idea is to store the robot locations and measured ranges the first few times the landmark is encountered and then obtain an estimate of landmark position by intersecting circles on the plane. Once an estimate of a new landmark is produced, the landmark is added to the Kalman filter where its estimate is then improved along with the estimates

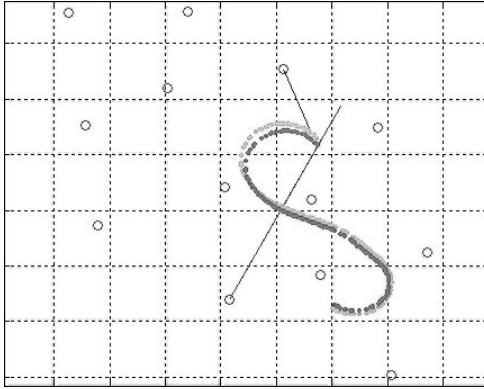


Fig. 3. 13 RF tags in known locations are used to localize a robot moving in an open area in conjunction with a wheel encoder and gyro. While individual range readings have a standard deviation of as much as 1.3 m, it is possible to localize the robot to within 0.3 m of the robots location. Tag locations are denoted by “o”. The true path is denoted by the lighter line and the estimated path is denoted by a darker one.

of the other (previously seen) landmarks. Because it takes advantage of the special structure of the problem, the resulting approach is less computationally cumbersome and avoids the local maxima problems associated with standard batch optimization techniques.

To collect data for this experiment, we used an instrumented autonomous robot that has highly accurate (2 cm) positioning for groundtruth using RTK GPS receivers as well as a fiber optic gyro and wheel encoders. Position is updated at 100 Hz. We equipped this robot with a RF ranging system (Pinpoint from RF Technologies) that has four antennae pointing in four directions and a computer to control the tag queries and process responses. For each tag response, the system produces a time-stamped distance estimate to the responding tag, along with the unique ID number for that tag. The distance estimate is simply an integer estimate of the distance between the robot and the tag.

The localization experiment was conducted on a flat, area about 30 meters by 40 meters in size. We distributed 13 RF tags throughout the area and then programmed the robot to drive in a repeating path among the tags. With this setup, we collected three kinds of data: the ground truth path of the robot from GPS and inertial sensors, the dead reckoning estimated path of the robot from inertial sensors only, and the range measurements to the RF tags. Results from our experiments are shown in Figures 3 and 4. Greater details can be found in [2].

3 Information Flow

Sensors detect information about the area they cover. They can store this information locally or forward it to a base station for further analysis and use. Sensors can also use

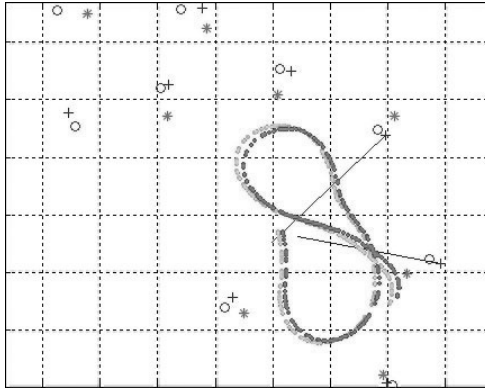


Fig. 4. In the case that the tag locations are unknown to start, they can be determined approximately using a batch scheme and then these approximate locations are used in a Kalman filter to continuously update the position of the tag along with the position of the robot. The “o”s represent actual tag locations, “*”s represent the initial estimate of the tag positions and “+”s represent the final estimation of tag position. The true path is denoted by the lighter line and the estimated path is denoted by a darker one.

communication to integrate their sensed values with the rest of the sensor landscape. Users of the network (robots or people) can use this information as they traverse the network.

We have developed distributed protocols for navigation tasks in which a distributed sensor field guides a user across the field [3]. We use the localization techniques presented above to compute environmental maps and sensor maps, such as temperature gradients. These maps are then used for human and robot navigation to a target, while avoiding danger (hot areas).

Figure 1(Left) shows a picture of a room in which a fire was started. We have collected a temperature gradient map during the fire burning experiment as shown in Figure 1. The Mote sensors¹ were deployed by hand at the locations marked in the figure. The sensors computed multi-hop communication paths to a base station placed at the door. Data was sent to the base station over a period of 30 minutes.

3.1 Directional Guidance

We used the structure of the data we collected during the fire burning exercise to develop a navigation guidance algorithm designed to guide a user to the door, in

¹ Each Mote sensor (<http://today.CS.Berkeley.EDU/tos/>) consists of an Atmel ATmega128 microcontroller (with 4 MHz 8 bit CPU, 128KB flash program space, 4K RAM, 4K EEPROM), a 916 MHz RF transceiver (50Kbits/sec, 100ft range), a UART and a 4Mbit serial flash. A Mote runs for approximately one month on two AA batteries. It includes light, sound, and temperature sensors, but other types of sensors may be added. Each Mote runs the TinyOS operating system.

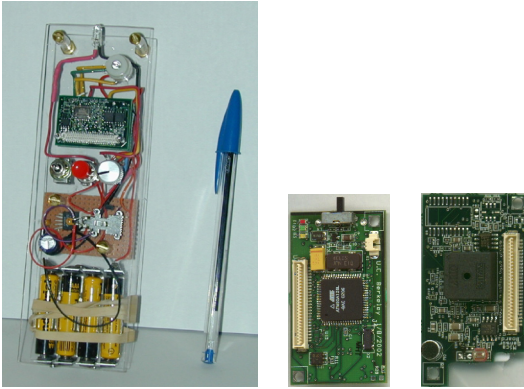


Fig. 5. The left figure shows the Flashlight prototype. The center figure shows a Mote board. The right figure shows the Mote sensor board.

a hop-by-hop fashion. We have deployed 12 Mote sensors along corridors in our building and guide a human user out of the building. Using an interactive device that can transmit directional feedback called Flashlight [4] a human user was directed across the field.

The Flashlight prototype we designed and built is shown in Figure 5 (Left). This device can be carried by a human user or placed on a mobile robot (or flying robot) to interact with a sensor field. The *beam* of the Flashlight is sensor-to-sensor, multi-hop routed RF messages which send or return information.

The Flashlight consists of an analog compass, alert LED, pager vibrator, a 3 position mode switch, a power switch, a range potentiometer, some power conditioning circuitry, and a microcontroller based CPU/RF transceiver. The processing and RF communication components of the Flashlight and the sensor network are Berkeley Motes, shown in Figure 5 (Center, Right). A switch selects the sensor type (light, sound, temperature, etc.). When the user points the Flashlight in a direction, if sensor reports of the selected type are received from any sensors in that direction, a silent vibrating alarm activates. The vibration amplitude can be used to encode how far (in number of hops) was the sensor that triggered. The potentiometer is used to set the detection range (calibrated in number of network hops from sensor to sensor.) The electronic compass supplies heading data indicating the pointed direction of the device.

We have deployed 12 Mote sensors along corridors in our building and used the Flashlight and the communication infrastructure presented here to guide a human user out of the building. Figure 6 shows the map. The Flashlight interacted with sensors to compute the next direction of movement towards the exit. For each interaction, the user did a rotation scan until the Flashlight was pointed in the direction computed from the sensor data. The user then walked in that direction to the next sensor. Each time we recorded the correct direction and the direction detected by the Flashlight. The directional error was 8% (or 30 degrees) on average. However, because the corridors and office doorways are wide, and the sensors sufficiently dense, the exit was identified successfully. The user was never directed

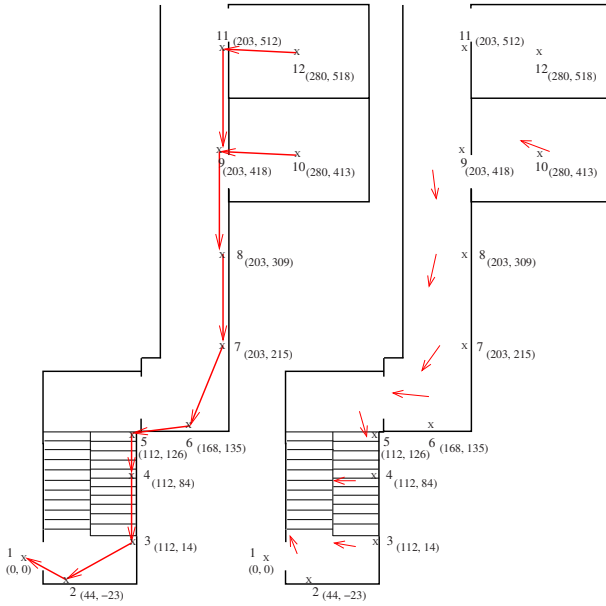


Fig. 6. (Left) The floor map for the directional guidance experiment. Arrows indicate the correct direction to be detected by the Flashlight. (Right) The floor map for the directional guidance experiment with the Flashlight feedback directions marked on it.

towards a blocked or wrong configuration. An interesting question is how dense should the sensors be, given the feedback accuracy.

4 Control of a Network of Robots

Robots augment the surveillance capabilities of a sensor network by using mobility. Each robot must use partial state information derived from its sensors and from the communication network to control in cooperation with other robots the distribution of robots and the motion of the team. We treat this as a problem of formation control where the motion of the team is modeled as an element of a Lie group, while the shape of the formation is a point in shape space. We seek abstractions and control laws that allow partial state information to be used effectively and in a scalable manner.

Our platforms are car-like robots equipped with omnidirectional cameras as their primary sensors. The communication among the robots relies on IEEE 802.11b networking. By using information from its camera system each robot is only able to estimate its distance and bearing from their teammates. However, if two robots exchange their bearing to each other, they are also able to estimate their relative orientations [6]. We use this idea to combine the information of a group of two or

more robots in order to improve the knowledge of the group about their relative position.

We have developed control protocols for using such a team of robots in connection with a sensor network to explore a known building. We assume that a network of Mote sensors previously deployed in the environment guide the robots towards the source of heat. The robots can modify their trajectories and still find the building exit. The robots can also switch between the potential fields (or temperature gradients) computed and stored in the sensor network (see Figure 7). The first switch occurs automatically when the first robot encounters a Mote sensor at a given location. The robots move toward the fire and stop at a safer distance (given by the temperature gradient). They stay there until they are asked to evacuate the building, at which point they use the original potential field to find the exit.

5 User Feedback

When robots or people interact with the sensor network, it becomes an extension of their capabilities, basically extending their sensory systems and ability to act over a much large range. We have developed software that allows an intuitive, immersive display of environments. Using, panoramic imaging sensors that can be carried by small robots into the heart of a damaged structure, the display can be coupled to head mounted, head tracking sensors that enable a remote operator to look around in the environment without the delay associated with mechanical pan and tilt mechanisms (see Figure 8).

The data collected from imaging systems such as visible cameras and IR cameras are displayed on a wearable computer to give the responder the most accurate and current information. Distributed protocols collect data from the geographically dispersed sensor network and integrate this data into a global map such as a temperature gradient that can also be displayed on a wearable computer to the user.

6 Discussion

The three groups met on August 23, 2002, at the Allegheny County firefighting training facility to conduct preliminary experiments involving a search and rescue exercise in a burning building (see Figure 1). A Mote sensor network was deployed manually in the building to collect temperature data and deliver the data to an outside point. A network of robots navigated the space. A network of cameras took panoramic images and IR images that were subsequently used to localize the robots. A network of radio tags was also used for localization. Although these modules were not integrated, the data collected during this exercise was used off-site to test the algorithms described in this paper. The firefighters who assisted us expressed great eagerness for having the kinds of support our vision provides.

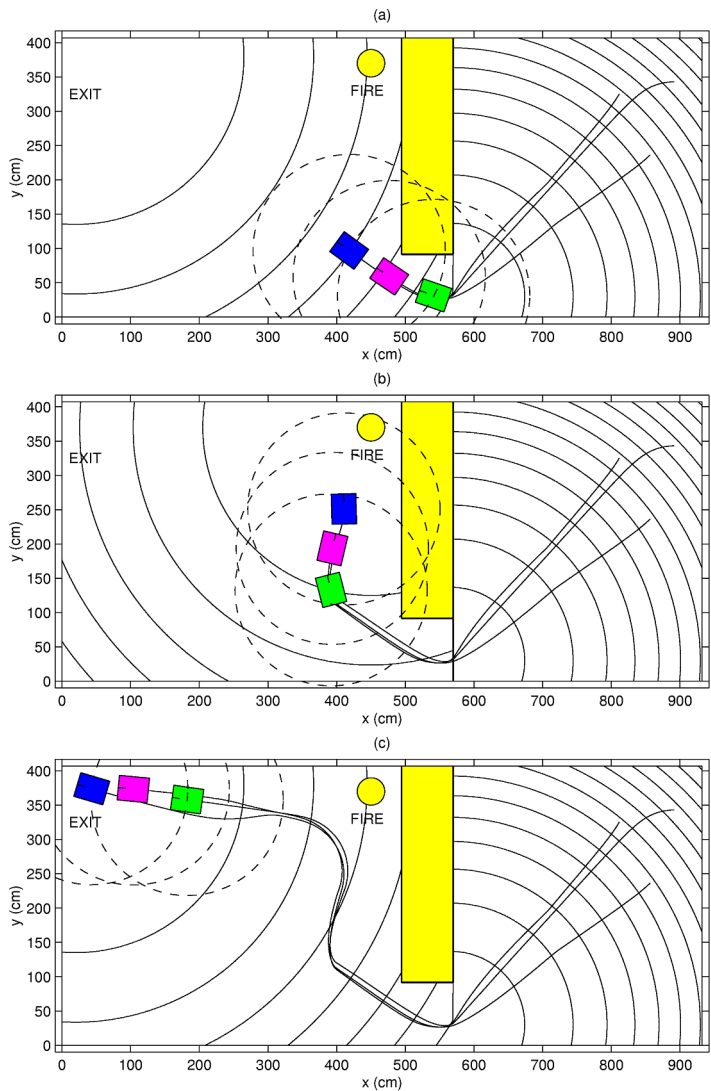


Fig. 7. Three robots switching motion plans in real time in order to get information from the hottest spot of the building. In (b) a gradient of temperature is obtained from a network of Mote sensors distributed on the ground.



Fig. 8. 360 degree panorama of a room in the burning building taken by a catadioptric system. The image can be viewed using a combination of a head tracker/head mounted display to enable a digital pan and tilt with very low latency.

References

1. G. Kantor and S. Singh. Preliminary results in range only localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1819–1825, 2002.
2. Derek Kurth, George Kantor, and Sanjiv Singh. Experimental results in range-only localization with radio. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.
3. Q. Li, M. de Rosa, and D. Rus. Distributed algorithms for guiding navigation across a sensor net. In *Proceedings of the International Conference on Mobile Computing and Networking (Mobicom)*, 2003.
4. R. Peterson and D. Rus. Interacting with a sensor network. In *Proceedings of the Australian Conference on Robotics and Automation*, 2002.
5. S. Singh, G. Kantor, and D. Strelow. Recent results in extensions to simultaneous localization and mapping. In *Proceedings of the International Symposium of Experimental Robotics (ISER)*, 2002.
6. J. Spletzer, A. K. Das, R. Fierro, C. J. Taylor, V. Kumar, and J. P. Ostrowski. Cooperative localization and control for multi-robot manipulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

Spraying Robot for Grape Production

Yuichi Ogawa¹, Naoshi Kondo², Mitsuji Monta³, and Sakae Shibusawa⁴

¹ Kawase Initiative Research Unit
RIKEN (The Institute of Physical and Chemical Research)
y-ogawa@riken.go.jp
<http://www.riken.go.jp/lab-www/THz/>

² Department of Technology Development
Ishii Industry Co., Ltd.
kondo-n@ishii-ind.co.jp

³ Faculty of Agriculture
Okayama University
monta@cc.okayama-u.ac.jp

⁴ Faculty of Agriculture
Tokyo University of Agriculture and Technology
sshibu@cc.tuat.ac.jp

Abstract. A robot which could spray chemicals under grapevine trellis was developed and experimented. From the experimental results, it was observed that the robot system made precise spraying operation and its precise operation record possible. Based on the precise operations and records, an optimum management of chemicals could be expected, that is, necessary amount of chemicals would be sprayed only at necessary places for protection of environment and ecosystem. In addition, it was considered that this robot would be able to contribute the minimum input-maximum output production system by establishment of traceability system in grape production.

1 Introduction

The world has been worrying about the quality of food supply because of recent problems with fruits, vegetables, meats, and other processed foods: food poisoning by bacteria, illegal unregistered agricultural chemicals, camouflage products, BSE, and etc. To solve these problems, a food traceability system which can show the food history during production, distribution and consumption, is desirable. Bio-production robots have substituted for human labor and contributed to raise marketing value of product and to produce uniform products. That also implies that many kinds of operations for food production can be precisely recorded by robots. It is, however, difficult for human to record all operations by manual, because there is too much information to write.

Many bio-production robots have been researched and been developed to achieve following purposes: to release from heavy, dangerous, or monotonous operations, to increase market value of products, to produce uniform products, to make hygienic / aseptic production conditions and etc. Here, it is necessary to add

record of information of bio-production operation to the above roles based on today's social situation.

Among the bio-production operations, one of the most essential information for safety food is sprayed chemical residues on products surfaces. Chemical spraying operation is conducted by human wearing protective clothing and a mask not to expose the operator to spray injury. It is desirable that a robot performs precision spraying not only to save chemical liquid but also to prevent spray injury to the environment and human body. Needless to say, the minimum spraying is important for consumers to purchase agricultural products. In this paper, a robot to precisely spray chemicals in vineyard by use of an ultrasonic sensor control is discussed for chemical residue information.

2 Experimental Device

2.1 Manipulator

Grapevines grow on a trellis training system in Japan. The height from the ground to the trellis is about 170-190 cm so that human can move under the trellis smoothly.

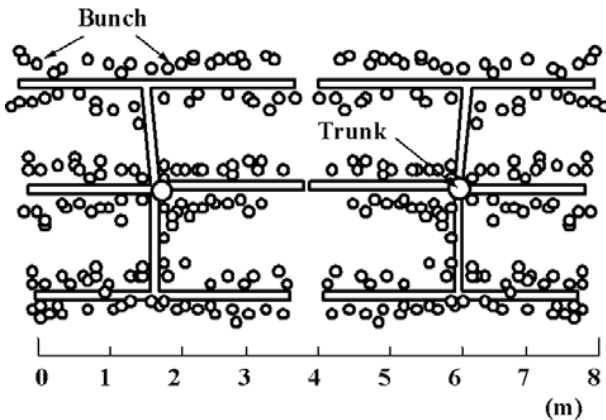


Fig.1. A grapevine field.

Figure 1 shows a sample of grapevine field. Chemical spraying operations are usually done by manual several times in summer season every year.

A five DOF polar coordinate manipulator shown in Figure 2 was applied for the training system at Okayama University, Japan, assuming that the robot traveled along the main scaffold. Since there were few obstacles under the trellis, the mechanism of the manipulator included a prismatic joint so that the manipulator could work with high speed by using a simple control method [1]-[4]. The length of the arm was 1.6 m, and the stroke was 1 m, while the manipulator weighed 200 kg. This manipulator was originally manufactured for grape harvesting operation [5].

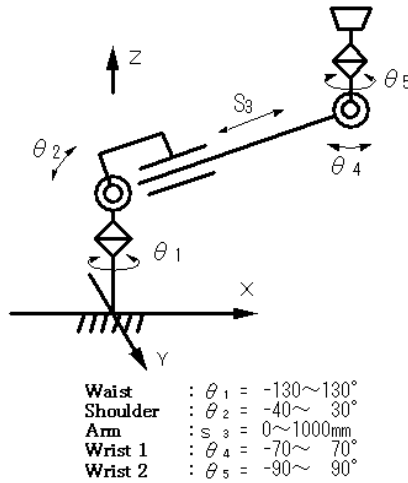


Fig.2. A five DOF polar coordinate manipulator.

2.2 End-Effector

Figure 3 shows spraying end-effector attached to the manipulator end. One of efficient spraying methods is to spray objective plants with certain quantity of chemical liquid per unit area uniformly. A nozzle was moved at a constant speed by the manipulator keeping distance between the nozzle and the target. Its spray angle, injection quantity and particle size of spray were 80 degrees, 21 l/hr and 45-210 μm respectively. A plunger pump was used to supply chemical liquid to the nozzle through a hose as shown in Figure 4, and a solenoid valve set between the pump and the nozzle switched its liquid flow.

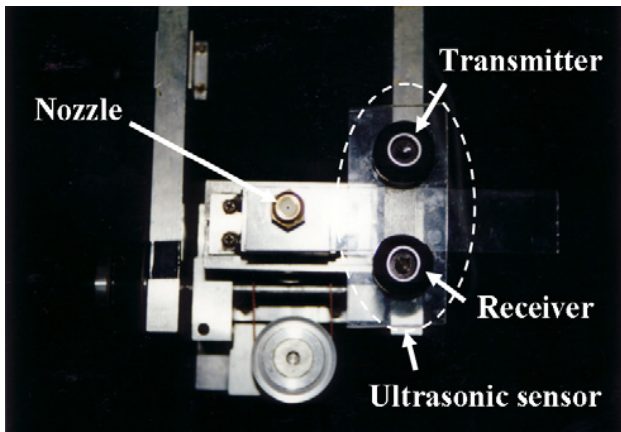


Fig.3. Spray nozzle and ultrasonic sensor.

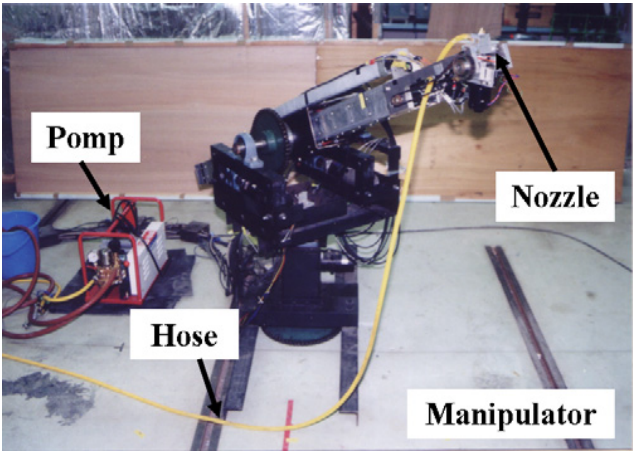


Fig.4. Spraying robot system.

2.3 Ultrasonic Sensor

Specifications of used ultrasonic sensor in this experiment are shown in Table 1. An ultrasonic sensor which 40kHz sonic wave was sent from a transmitter in every 57 ms was used. Its flight time until reflected wave reaches a receiver was measured and distance between the sensor and object was calculated.

Table 1. Specification of ultrasonic sensor

Type name	RS-240
Frequency	40kHz
Detection system	Ultrasonic pulse reflection method
Source voltage	DC12~24V
Consumption current	3.5mA
Maximum distance	6m
Temperature range	10~60℃

2.4 Traveling Device

Figure 5 shows the robot with a traveling device. The traveling device had crawlers, because the field was not tilled and the robot was relatively heavy. The width of the crawler was 360 mm, and the ground contact length was 1,010 mm. The width of the traveling device was 1,400 mm, the length was 2,300 mm, and the height from the ground to the plate on which the robot was mounted was 420 mm. The traveling speed of the device could be changed from 0 to 2 m/s. In the experiment, it was assumed that the device was manually steered into row between the trees, and that the robot operated. Since studies on autonomous mobile systems have been reported [6], this traveling device was manually controlled.



Fig.5. Crawler type traveling device.

3 Experimental Method

3.1 Spray Uniformity

The manipulator is controlled so that the spray nozzle can move below the trellis keeping the distance between nozzle and trellis constant based on distance information from the ultrasonic sensor in order to uniformly spray the target. To evaluate the spray results comparing with human, black colored water was used in stead of chemical liquid and was sprayed to a flat white paper which was set above the manipulator end in parallel to the ground.

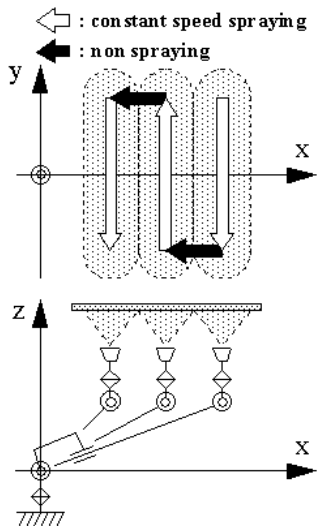


Fig.6. Spraying course.

Figure 6 shows a spraying course by robot. It was assumed that this robot sprayed at a unit area after a traveling device stopped moving in order to minimize the influence of vibration caused by a traveling device or rugged ground. The area of spraying by this robot was about 1.5 m² during the traveling device stop. The spray nozzle was controlled to move in a linear motion with constant velocity keeping constant distance between the nozzle and the object to spray uniformly. The spraying was stopped not to overlap-spray by switching an electric valve when the nozzle moved from a line to the next line. The spraying operation by human was also conducted in the same way with robot to compare its uniformity of spraying.

Texture analysis methods were used for evaluation of spray uniformity by robot and human. Images were acquired by a color TV camera and a capture board which was installed in a PC. NTSC analog signal was inputted to the capture board whose pixel number was 256 × 256 with 256 gray levels. Green component images were used for the texture analysis. In this experiment, three textural features, ASM (angular second moment), IDM (inverse difference moment) and CON (contrast) were used [7]. ASM and IDM indicate the measure of homogeneity in a large area and in a local area of image respectively, while CON indicates the measure of difference of grey level in the whole of image [8].

The texture-context information is adequately specified by the matrix of relative frequencies p_{ij} with which two neighboring resolution cells separated by distance d occur on image, one with gray tone i and the other with gray tone j as shown in Figure 7.

In a case of $d=1$ and $a=0$, the three textural features are calculated as follows:

$$\text{ASM: } \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \{p(l,0)(i, j)\}^2 \quad (1)$$

$$\text{CON: } \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \frac{p(l,0)(i, j)}{l + (i - j)^2} \quad (2)$$

$$\text{IDM: } \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i - j)^2 p(l,0)(i, j) \quad (3)$$

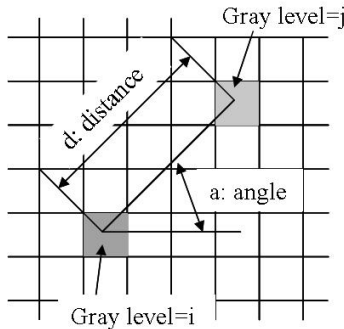


Fig.7. Co-occurrence matrix.

3.2 Obstacle Avoidance

Grapevine trellis is usually not flat but fluctuated according to its leaves, stems and bunches growth. It is necessary that the manipulator should be controlled along the fluctuated trellis to uniformly spray. In this experiment, the manipulator was continuously path-controlled at a constant speed keeping a same distance between end-effector and the trellis (30 cm).

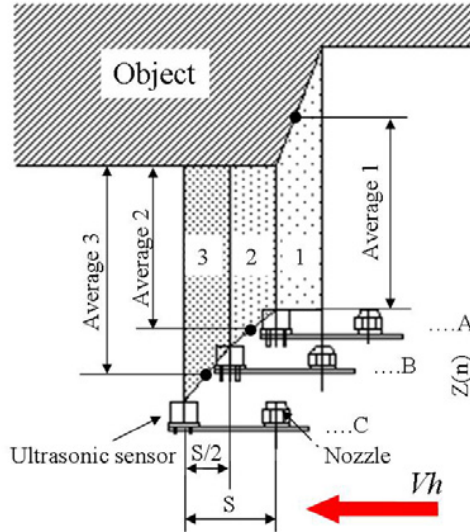


Fig.8. Distance measuring method.

Figure 8 shows distance measurement and control methods. Manipulator moves from right to left in this figure in velocity V_h and the ultrasonic sensor detects distance several times during moving for length $S/2$ (half pitch of length between ultrasonic sensor and spray nozzle). The average of the measured distances is used for vertical moving distance $Z(n)$.

Although many leaves completely covers trellis somewhere, there are many spots where no leaf grows on actual trellis in the field. When the ultrasonic sensor met the no leaf spot, the sensor outputted the maximum value of $Z(n)$ and the manipulator was supposed to be abruptly ordered to move toward out of its operational space. To avoid this problem, the maximum distance data was not added to the data for average calculation. When more than half of the distance data were the maximum, the manipulator was controlled on horizontal movement.

Since sampling interval of the ultrasonic sensor was constant (60ms), sampling times during moving for distance $S/2$ were changed according to the manipulator moving velocity V_h ; 20 times for V_h 50 mm/s, 10 times for 100 mm/s, and 5 times for 200 mm/s. In this experiment, the distance between nozzle and object was set to 300 mm.

4 Result and Discussion

4.1 Spray Uniformity

Figure 9 shows a set of gray level histograms (Y axial direction in Figure 6) of spraying pattern on white color flat paper. The Robot and human sprayed on the condition that the distance between the nozzle and the object was 300 mm and that velocity of the nozzle was 200 mm/s. The horizontal axis indicates pixel number of image and the vertical axis indicates 256 gray levels.

From this result, it was observed that gray level value of robot spray was apparently more uniform than that of manual spray and that human could not spray constantly even on flat plane. It was predicted that robot could follow the fluctuated grapevine trellis so precisely that the difference of spray uniformity would be larger on an actual trellis than this result.

Table 2 shows a result of evaluation by the textural features. It was observed that ASM and IDM of robot spray images were higher than those of human spray images, while CON of robot was lower than that of human, which implies that robot could sprayed uniformly. From the results, it was considered that robot could spray more uniformly than human not only in whole larger area but also in smaller area.

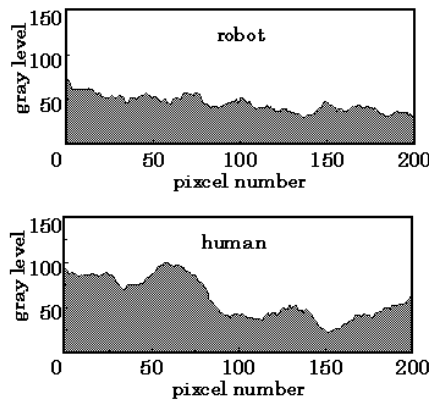


Fig.9. Comparision of spray uniformity.

Table.2. Result of Textural analysis.

X axial features			
	ASM	IDM	Contrast
Robot	0.20	0.75	0.63
Human	0.07	0.62	1.56

Y axial features			
	ASM	IDM	Contrast
Robot	0.28	0.86	0.27
Human	0.15	0.83	0.36

4.2 Obstacle Avoidance

Figure 11 shows a result of obstacle avoidance when an artificial trellis was used in room (Figure 10). When the manipulator moving speed was slow, the ultrasonic sensor was supposed to detect the distances at many points. The manipulator end was, therefore, able to precisely follow the shape of plant on trellis and to uniformly spray. Furthermore, the manipulator could avoid a fruit existed on the way. However, the manipulator end could not follow an abrupt large irregular shape sometimes in case of 200 mm/s moving speed as shown in Figure 11.

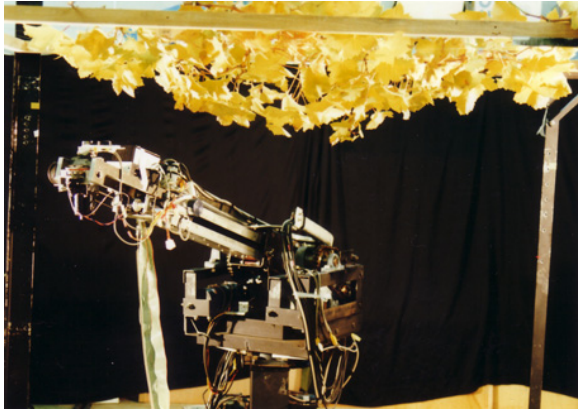


Fig.10. A grapevine trellis.

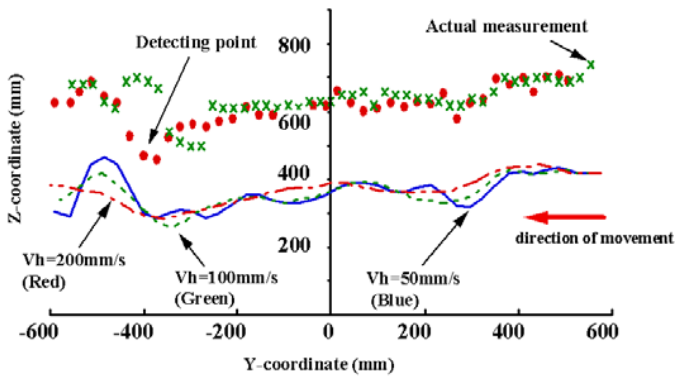


Fig.11. A result of experiment.

This precise spraying makes prediction of chemical residues on agricultural products possible by a simple calculation, if the spray record (type of chemical, spray quantity per unit area) is kept and the spray information is linked to GIS information. It is possible that total quantity of chemicals for protect from insect injuries and disease is determined depending on local region, on crop, on season, and on other conditions. Chemical spraying operation has been usually done in

every year, even if no vermin or no disease was found, because chemical spray operation is conducted not for extermination but for prevention. When a field monitoring system to find insect injury or diseases on early stage is developed by machine vision recognition adding to this precise spraying technology, it is expected that necessary chemicals can be sprayed only at necessary places for protection of environment and ecosystem by establishment of traceability system.

5 Conclusion

From these experimental results, it was considered that the precise spraying operation became possible by using of robot and that prediction of chemical residues on agricultural products was also possible because the chemical spraying operation was easily recorded by the robot system. If a traceability system including chemical spray operation is established, an inspection system of chemical residue on agricultural products will be earlier realized because chemical quantity in the field and chemical residues are able to be calculated. In addition, a monitoring system to early detect insect injuries and diseases of products is desirable for the minimum chemical spraying.

References

1. Kondo, N, "Study on Grape Harvestion Robot", *Proc. IFAC/ISHS 1st Workshop on Material and Control Applications in Agriculture and Horticulture*, pp. 243-246, 1991.
2. Kondo,N. et al, "Basic Studies on Robot to Work in Vineyard (Part 1)", *Journal of the Japanese Society of Agricultural Machinery*, 55(6), pp.85-94.(in Japanese), 1993.
3. Kondo,N. et al, "Basic Studies on Robot to Work in Vineyard (Part 2)", *Journal of the Japanese Society of Agricultural Machinery*, 56(1), pp.45-53.(in Japanese), 1994.
4. Monta,M. et al, "Basic Studies on Robot to Work in Vineyard (Part 3)", *Journal of the Japanese Society of Agricultural Machinery*, 56(2), pp.93-100.(in Japanese), 1994.
5. Monta,M., Kondo,N. and Shibano,Y., 1995a. Agricultural robot in grape production system. In *Proc. 1995 IEEE International Conference on Robotics and Automation*, vol.3: 25042509.
6. Kondo, N. et al, *Robotics for Bioproduction systems*, American Society of Agricultural Engineers, Michigan, USA, 1998.
7. Haralick,R.M. et al, "Textural Features for Image Classification", *IEEE Transactions on Systems, Man, and Cybemetics*, Vol. SMC-3, No. 6, pp. 610-621, 1973.
8. Monta,M., Kondo,N., Shibano,Y. and Mohri,K., 1995b. End-effectors for agricultural robot to work in vineyard, *Acta Horticulturae* 399: pp. 247-254.

Path Planning for Complete Coverage with Agricultural Machines

Michel Taïx¹, Philippe Souères¹, Helene Frayssinet¹, and Lionel Cordesses²

LAAS-CNRS
7 Av. du Colonel Roche,
31077 Toulouse Cedex 4, France
{name}@laas.fr
RENAULT Agriculture, R&D
7 Rue Dewoitine,
78141 Vélizy, France
cordesses@renagri.com

Abstract. The problem of planning reference trajectories for agricultural machines is considered. A path planning algorithm to perform various kinds of farm-works is described. The case of convex fields is first considered. A direction of work being given, the algorithm determines the turning areas and selects a trajectory which guarantees the complete field coverage while minimizing overlapping. The method is extended to the case of fields with more complex shape including possibly obstacles. Simulations are proposed to illustrate the reasoning.

1 Introduction

This paper presents a research work issued from a collaboration between RENAULT Agriculture and the LAAS-CNRS which concerns the automatic guidance of high-end farm tractors on the base of GPS data. Steering strategies can be divided into two classes: relative guidance and absolute guidance. Relative guidance consists steering the vehicle by regulating its posture with respect to the track resulting from the previous passage (crop or ploughing line). In that case, trajectories are often rectilinear and parallel. Absolute guidance consists in tracking a reference path, or a trajectory, issued from a path planning strategy [4], [6]. Our work deals with the absolute guidance problem. It focuses on the description of a trajectory planning algorithm which provides a field coverage strategy adapted to various kinds of farm-works [15], [10]. The main difficulty of the problem comes from the need to realize the complete covering of the field, that is including the regions inside which the manoeuvre are executed. Planning the trajectories inside the manoeuvre area states a difficult problem which is crucial for agricultural applications. Indeed, while these zones are usually covered at the end when ploughing, they need to be worked at the beginning when harvesting. Previous work devoted to the coverage problem only provide algorithms for the case of simple rectangular areas and do not address the planning problem inside manoeuvre areas. In [16], [1] [5], [13], cellular decomposition approaches have been proposed based on breaking down the workspace. The Spiral-STC algorithm proposed in [9] is based on a discretization of the working area and the definition of a spanning tree to solve coverage. Considering fields with more complex shapes states another difficult problem. Indeed, in that case the working direction may differ from a region to another and a cell decomposition

has to be done. Such an approach is proposed in [11] where a sequence of sub-regions is selected with different planar sweep lines to compute the coverage path. Theoretical results based on computational geometry can be found in [2], [3].

The algorithm presented in this paper allows to determine automatically the manoeuvre areas and select a covering trajectory which minimizes overlapping. The planning approach is first presented in the case of convex fields. Two strategies are proposed to this end. On this base the presence of obstacle is then considered and the method is extended to the case of fields with more convex shape.

2 The Automatic Guidance Project

The path planning algorithm presented in this paper comes as a part of an industrial project of RENAULT Agriculture which aims at developing autonomous navigation abilities for farm tractors.

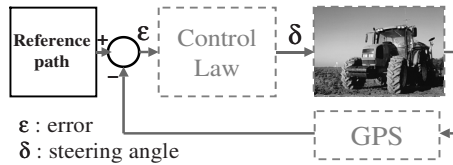


Fig. 1. Farm tractor control system overview

The GPS-based farm tractor control system is based upon the following four units (figure 1):

- The sensor: Real-time, kinematic GPS. Its high three dimensional (3D) accuracy ($\sigma < 2cm$) and its low latency ($t_{latency} < 0.2s$, see [7]) allow its use in a closed loop system. It outputs information about position and velocity of one point of the vehicle to control.
- The farm tractor to control: The only technical requirement is the availability of a model with an electro-hydraulic power steering instead of an all-hydraulic one. The steering angle can be supplied either by the driver, thanks to the driving wheel, or by the embedded computer.
- The Controller implemented on an embedded computer: The system is able to follow paths at various velocities [14,6] with an accuracy better than 10cm.
- The trajectory planner which determines the reference path to follow to perform a specific farm-work.

This paper focuses on the fourth unit only, namely the path planning problem.

3 Covering Path Planning

Farm-work experiments have proven that the choice of the working direction within the field has to be guided by two major factors. First, to reduce sliding and traction

efforts, the tractor must move at best in the direction of the slope and execute trajectories with very low curvature. Second, to reduce the number of manoeuvres, the direction of motion must be, as far as possible, parallel to the longer side of the field. In particular, in wedge-shaped regions, the motion must be parallel to one of the edges. To satisfy these constraints at best, it appears necessary to decompose the field into regions, and define in each of them a “Steering edge” *S-edge* which will guide the successive tracks. Furthermore, when planning trajectories, it is necessary to determine regions called “Turning areas” *T-areas*, located at extremities of the field, inside which the tractor will execute U-turns or manoeuvres. The width of *T-areas* depends on the tractor’s characteristics and the nature of the tool.

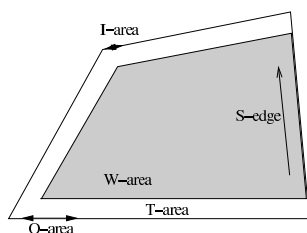


Fig. 2. Definitions

The remaining part of the field constitutes the “working-area”, *W-area*. Inside this central region, the farm-work trajectories are most part of time rectilinear parallel tracks directed along the *S-edge*.

The algorithm proposed in this paper applies to polygonal fields including at most one vertex of concavity. An extension is proposed to consider the case of fields including one moderate curved boundary, that is one smooth low-curved boundary along which the tractor can move. This restriction allows to consider most part of fields encountered in real applications. For such a field, once the input area, *I-area*, and the output area, *O-area*, have been defined on the field’s boundary, the path-planning problem can be stated as follows:

Determine a trajectory starting from a point in the I-area and ending at a point of the O-area which guarantees the coverage of the whole field (W-area + T-areas) while minimizing the overlapping between adjacent tracks and the number of manoeuvres.

Note that, depending on the nature of farm-work, the covering of the *T-areas* is done at the beginning or at the end of the task. For instance, when ploughing the *T-areas* are to be covered at the end, while they are worked at the beginning during harvest. The algorithm is based on the partitioning of the field into convex polygons. The partitioning process is described in section 3.3. Inside each convex polygon, a *S-edge* is determined and a set of characteristic points is defined at the boundary of the *W-areas* and the *T-areas*. These points will constitute the nodes of a graph upon which the trajectory is defined. Two strategies are proposed to this end. The trajectory planning strategy is first described for the case of a convex polygon free of obstacles in section 3.1. The presence of obstacles is considered in section 3.2. Depending on

the size of the obstacles two avoidance strategies are proposed. Finally, section 3.4 describes the extension of the method to the case of fields including one moderate curved border.

3.1 Case of Convex Polygonal Fields

This section presents the trajectory planning method for the case of a convex polygon free of obstacles. The input data are the *S-edge*, the *I-area*, the *O-area*, the kind of farm-work to be executed and the characteristics of the tractor and the tool (type, width, curvature radius).

The path planning strategy is based on three successive steps. The first one is a topological representation of the field which consists of determining a set of characteristic points from which a graph is defined (section 3.1). On this base, two strategies are proposed to construct the reference trajectory.

Determination of characteristic points Once the S-edge is specified, the T-areas are computed by taking into account the space required to perform the turning manoeuvres. In practice, this space is a whole number of the tracks width. This implies to shift or add a pair of characteristic points to guarantee the field coverage without overflow. Outside the T-areas, the field is covered by parallel tracks directed along the S-edge. The tracks are arranged in such a way to insure the complete field covering while minimizing overlapping. Following the same technique, the T-areas are also covered by parallel tracks but directed along the side-edges. The end points of all working-tracks are considered as characteristic points (see figure 3 left).

Construction of the trajectory In order to construct of the trajectory, the characteristic points are considered as the nodes of a graph. Two strategies are proposed to define the arcs and explore this graph. The first one is based on the search for the best Hamiltonian path according to the minimization of a cost criterion, while the second involves a simpler geometric reasoning.

Hamiltonian graph exploration: Let $X = \{x_1, x_2, \dots, x_n\}$ be the set of characteristic points defined by the end points of tracks. These points are considered as the nodes of a graph. A set of graph edges $U = \{u_1, u_2, \dots, u_m\}$ is then defined, representing rectilinear paths between these nodes from which the different kind of farm-work can be synthesized. To achieve a given farm-work, a specific value is assigned to the graph edges. The coverage strategy is deduced from a search within this graph $G(X, U)$. Seven types of edges are to be considered depending on the kind of displacement they represent. A specific value p_i is associated to each type (see figure 3 right):

- p_1 : to execute a working track inside the field,
- p_2 : to pass from a working track to the next one,
- p_3 : to jump from a working track to a track located one after the next one (to avoid manoeuvres),
- p_4 : to jump from a working track to any other track except the next two,
- p_5 : to jump from a working track to a point located inside the T-area,

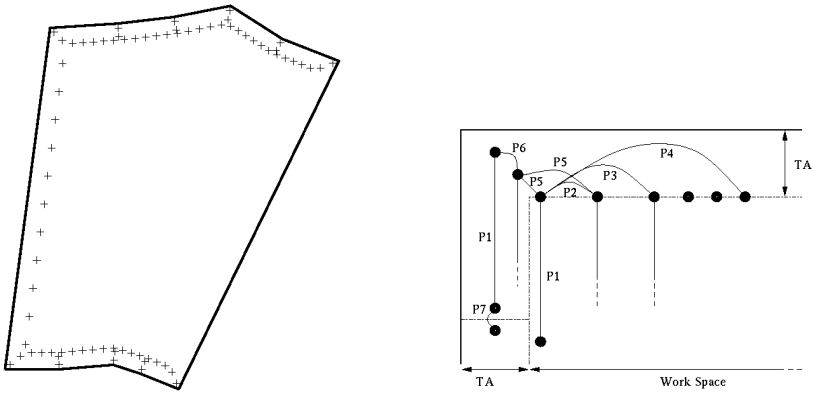


Fig. 3. Characteristic points (left) and arc notation (right)

p_6 : to pass from a working track to the next one inside the T-area,
 p_7 : to pass from a T-area to another one.

From the above construction, the determination of a trajectory is based on the search for an Hamiltonian path in the graph $G(X, U)$. If such a path exists, it insures the whole coverage of the field and minimizes the overlapping, as it passes once through each characteristic point. The problem of finding an Hamiltonian cycle within an undirected graph is NP-complete, i.e. the solution cannot always be found in $O(n^k)$ for k constant. The proposed algorithm is based on the following reasoning. From each node, the edge with maximal weight p_i is selected to reach the next node. This comes to solve a local maximization problem at each step. To evaluate the quality of the solution, a cost criterion depending on path length, working duration, number of U-turns and jumps in T-areas is evaluated. To reduce the computation time an heuristic is introduced which consists of minimizing the set of possible initial points. In practice, the application of this heuristic leads to the optimal solution or to a near optimal solution. Figure 4 shows the final trajectory in the case of a non-convex field. Note that the decomposition in three regions has been done manually.

Geometrical method: Though the previous method allows to determine a near optimal solution when exists, the algorithm complexity is high (NP complete) and therefore highly time-consuming. This fact has motivated the development of a simpler algorithm based on a geometric reasoning. The solutions provided by this second approach turn out to be close to usual agricultural habits.

The reasoning is based on the same set of characteristic points. The idea is to restrict the search for a covering solution to trajectory starting from the external tracks of the *W-area*. As there exist two possible directions of motion along these external tracks, only four solutions are to be considered for each convex polygonal cell. The same reasoning is used to cover the *T-areas*. Note that, depending on the nature of the farm-work, the covering of the *T-areas* has to be done at the beginning or at the end. For each *I-area* and *O-area*, a complete solution is computed to guarantee

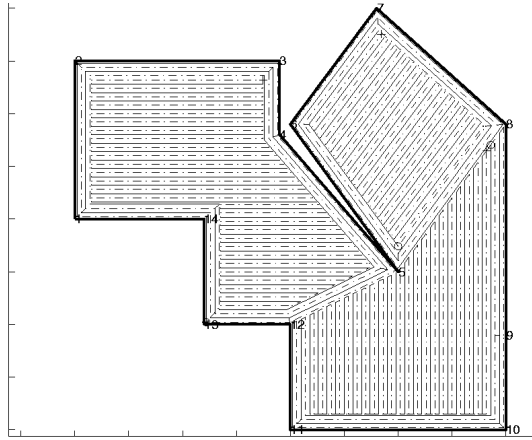


Fig. 4. Solution based on Hamiltonian graph exploration

the whole coverage ($W\text{-areas} + T\text{-areas}$). The algorithm selects the solution along which the cost criterion (which is function of the path length and the number of manoeuvres) is minimized. Figure 5 shows the same field for two $S\text{-edge}$ directions, note that the number of $T\text{-areas}$ is different.

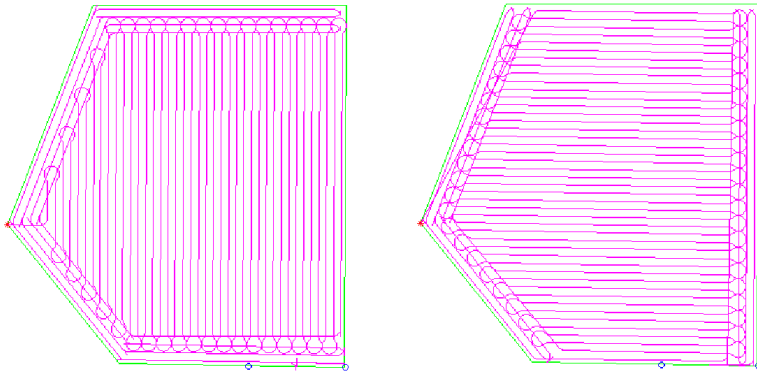


Fig. 5. Example of $T\text{-area}$: a unique one (left), or two separate (right)

3.2 Case of Convex Polygonal Fields with Obstacle

The algorithm is restricted to convex polygonal obstacles. Depending on the size of obstacles with respect to the tool width, two navigation strategies are proposed. An obstacle is considered as *small* if it intersects at most three adjacent tracks of the nominal trajectory, otherwise it is referred to as *large*.

Large size obstacles: The case of large size obstacles needs to be considered first as it induces a strong modification of the structure of the nominal trajectory. The method involves a sub-partition of the field into convex cells (see figure 6) and the introduction of an additional *T-area* around each obstacle. These new turning zones will be used to insure the whole coverage and allow the transition between the adjacent cells surrounding the obstacle.

Algorithm 1

```

Sort the obstacle with the top point
Do vertical plane-sweep from top to bottom
if Cell begin with top point obstacle then
    Do Follow-Obstacle (see algorithm 2)
else
    Sweep cell
end if

```

The navigation strategy is defined as follows: sweep the vertical plane, when a new obstacle is detected (algorithm 1) work all the cells around. Start from the cells located on the right side, then climb up to the top of the obstacle and finish by the left cells. At the end of this stage it is necessary to work the *T-area* located around the obstacle. This method guarantees that the tractor will not pass through a cell already worked in order to start working a new one (algorithm 2).

Algorithm 2 Follow-Obstacle

```

Sweep the right cells from top to bottom
if Not new obstacle then
    Sweep cell
else
    Top point obstacle implies Follow-Obstacle
end if
Climb up to the top point
Sweep the left cells from top to bottom
if Not new obstacle then
    Sweep cell
else
    Top point obstacle implies Follow-Obstacle
end if
Work T-area around the obstacle
Change cell

```

Small size obstacles: In this case, the avoidance strategy consists of modifying locally the nominal trajectory. First a covering trajectory is computed for the whole field, by considering the large obstacles, while ignoring the small ones. Each time a track intersects a small obstacle, a local “avoidance-trajectory” starting from the

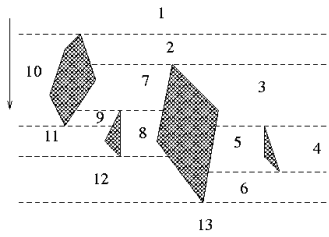


Fig. 6. Sequence of work in the cells

track before the obstacle and reaching back the track beyond it, is computed. The avoidance process is then defined by the following five steps (see figure 7):

- move towards the obstacle along the nominal track until the tractor is close to contact,
- move backwards along the track until the starting point of the avoidance curve is reached¹,
- move along the avoidance curve until the track is reached anew,
- move backwards along the track until the tool is closed to contact,
- move forwards until a new small-size obstacle is detected.

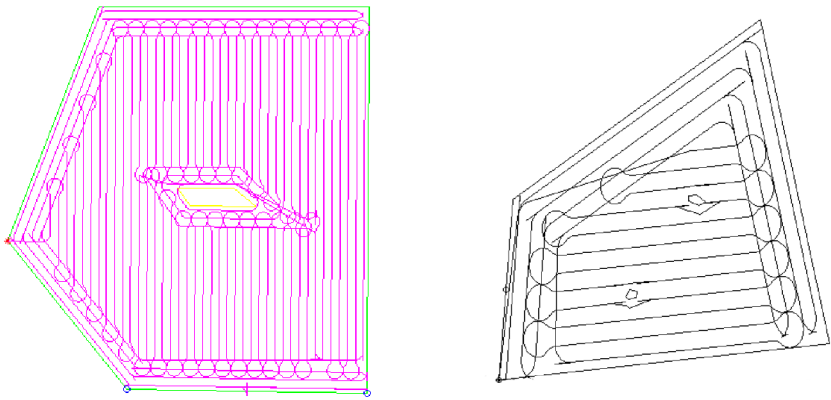


Fig. 7. Examples of obstacle avoidance: large (left) and small (right)

3.3 Case of Nonconvex Polygonal Fields

Let us consider now the upper level of the algorithm, that is the case of polygons including at most one concave vertex (i.e. only one internal angle strictly greater than

¹ In practice a security distance to the obstacle is defined

π). In that case, the method consists in partitioning the field into two adjacent convex cells by defining a boundary segment issued from the concave vertex. The location of the remaining extremity of the boundary segment is chosen so as to minimize its length. Indeed, the partition induces an additional *T-area* which needs to be covered by the trajectory. Finally, in each convex cell the *S-edge* is chosen so as to minimize the number of manoeuvres, or equivalently the number of parallel tracks. The cost criterion associated to the partition is equal to the sum of the costs of the two convex fields plus the length of the additional border segment. This algorithm could be easily extended to consider more complex polygons. Indeed, the partitioning method can be iteratively applied to consider additional vertices of concavity.

3.4 Case of Fields Including One Moderate Curved Boundary

The algorithm has been extended to consider the case of fields including one moderate curved boundary, that is one smooth low-curved boundary along which the tractor can possibly move. The coverage method is based on a convex polygonal approximation of the field and the translation of the curved boundary as follows:

- considering the external vertex construct a convex polygonal hull of the field,
- inside this convex hull, construct a sub-polygon by translating normally the edge corresponding to the curved boundary until it is wholly include in the field,
- by translating the moderate curved boundary, draw the minimal number of curved tracks which are necessary to cover the region of the field located outside the sub-polygon,

On this base, the trajectory is determined by using the algorithm described in section 3.1 to cover the convex cell, while the remaining part is covered by the successive translated curves (see figure 8). Though the successive translated tracks necessarily overlap, this method guarantees to avoid the singularities that may occur when constructing tangent tracks. Indeed, drawing successive adjacent curved tracks may lead to a rapid increase of their curvature making them no more admissible.

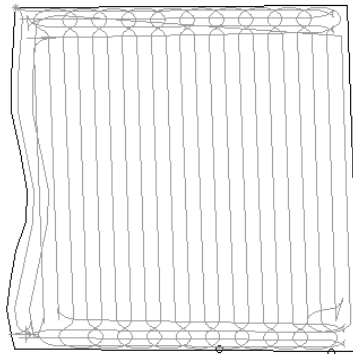


Fig. 8. Example of moderate curved boundary

4 Conclusion

The proposed algorithm based on the treatment of convex cells and the definition of characteristic points can simply be extended to the case of fields with more complex shape. The extension has been studied for the case of fields including one vertex of concavity or one moderate curved boundary. More complex situations could be considered following the same approach. Simulations provide very satisfactory results in the sense that they are realistic and close to the current practices of farmers.

References

1. E. Acar, H. Choset, Y. Zhang and M. Schervish "Path Planning for Robotic Demining: Robust Sensor-Based Coverage of Unstructured Environments and Probabilistic Methods" *Int. Journal of Robotics Research*, V. 22, N. 7-8, 2003
2. E.M. Arkin, S.P Fekete and J.S.B. Mitchell, "Approximation Algorithms for Lawn Mowing and Milling", *Compt. Geom. Theory Appl*, 1997.
3. E. Arkin, M. Bender, E. Demaine, S. Fekete, J. Mitchell, and S. Sethia "Optimal Covering Tours with Turn Costs" *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, 2001.
4. T. Bell, M. O'Connor, V. Jones and A. Rekow, "Realistic autofarming closed-loop tractor control over irregular path using kinematic GPS" in *Europ. Conf. on Prec. Agriculture*, 1997.
5. H. Choset and P. Pignon, "Coverage Path Planning: The Boutrophedon Cellular Decomposition", *Int. Conf. on Field and Service Robotics*, 1997.
6. L. Cordesses, B. Thuilot, P. Martinet and C. Cariou, "Curved path following of a farm tractor using a CP-DGPS", in *Proceeding of the 6th Symp. on Robot Control, SYROCO*, Austria, 2000.
7. L. Cordesses, C. Cariou, C. Veron, and J. Gallice "Image processing for GPS latency measurements", in *QCAV*, 2001, vol 1, pages 287-291, 2001.
8. S. Fabre, P. Soueres, M. Taix and L. Cordesses, "Farm-work path planning for field coverage with minimum overlapping", *IEEE ETFA*, 2001
9. Y. Gabriely and E. Rimon, "Spiral-STC: an On-Line Coverage Algorithm of Grid Environments by a Mobile Robot", *IEEE Int. Conf. on Rob. & Aut.*, 2002.
10. C. Hofner and G. Schmidt, "Path planning and guidance techniques for an autonomous mobile cleaning Robot", in *Robotics and Autonomous Systems*, 14:199-212, 1995.
11. W. Huang, "Optimal Line-sweep-based Decomposition for Coverage Algorithms", *IEEE Int. Conf. on Rob. & Aut.*, 2001.
12. C. Luo, S. Yang, D. Stacey and J. Jofriet, "A Solution to Vicinity Problem of Obstacles in Complete Coverage Path Planning", *IEEE Int. Conf. on Rob. & Aut.*, 2002.
13. R. Neumann de Carvalho, H.A. Vidal, P. Vieira and M.I. Ribeiro, "Complete Coverage Path Planning and Guidance for Cleaning Robots", *Institute for Systems and Robotics*, Lisbon, Portugal, December 1995.
14. D. Bevilacqua and B. Parkinson "Carrier phase differential GPS for control of a tractor towed implement" in *Proceedings of ION-GPS*, Salt Lake City, USA, 2000.
15. T. Pilarski, M. Happold, H. Pangels, M. Ollis, K. Fitzpatrick and A. Stentz, "The Demeter System for Automated Harvesting", *Robotics Institute, Carnegie Mellon University*, Pittsburgh PA 15213, U.S.A., 1998.
16. A. Zelinsky, R.A. Jarvis, J.C. Byrne and S. Yuta, "Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot", in *Int. Conf. on Advanced Robotics*, Japan, 1993.

Author Index

Takeshi Aoki	509	Andrew Howard	145
Tatsuo Arai	255	Daniel F. Huber	189, 497
Hajime Asama	41, 245	Akihiro Ikeuchi	519
Christopher Baker	487	Kenji Inoue	255
Carlos Balaguer	407	Genya Ishigami	225
Frédéric Bourgault	209	Takashi Isoyama	385
Graham Brooker	311	Alberto Jardón	407
Gregg Buskey	267	Hayato Kaetsu	245
Ramiro Cabas	407	Naoto Kakuta	385
Tsuneo Chinzei	385	Sungchul Kang	355
Gregory S. Chirikjian	437	George Kantor	529
Changhyun Cho	355	Ilkka Kauppi	377
Sayeed Choudhury	437	Kuniaki Kawabata	41, 245
Daisuke Chugo	245	Alonzo Kelly	343
Lionel Cordesses	549	Jong-Hyuk Kim	299
Peter Corke	61, 459	Munsang Kim	355
Raúl Correal	407	Boris Kluge	71
Christophe Coué	199	Shigeru Kobayashi	519
Aveek Das	529	Naoshi Kondo	539
Antonio Diaz-Calderon	343	Kazuhiro Kosuge	3, 333
Mark Diel	287	Noriho Koyachi	255
Raghavendra Donamukkala	103	Vijay Kumar	529
Elliot S. Duff	51	Daisuke Kurabayashi	41
Matthew Dunbabin	459	Yoshihiro Kuroki	13
Hugh F. Durrant-Whyte	179, 209	Derek Kurth	189
Michael Fiegert	155	Christian Laugier	199
Helene Frayssinet	549	Michel Lauria	235
Tomonari Furukawa	209	Jonghwa Lee	355
Aantonio Giménez	407	Sangyoon Lee	437
Ali Haydar Göktoğan	311	Yasushi Mae	255
Geoffrey Gordon	417	Shoichi Maeyama	125
Charles-Marie De Graeve	155	Ian Mahon	93
Martin Hägele	447	Yasushi Makihara	427
Dirk Hähnel	287	Roman Mázl	135
David W. Hainsworth	469, 477	Ronald. J. McPhee	469
Aarne Halme	377, 395	Daisuke Mishima	509
Chad O. Hargrave	477	Taketoshi Mishima	245
Panu Harmo	395	Jun Miura	427
Kazuma Hashimoto	113	Noriyuki Mizuno	225
Martial Hebert	103	Mitsuji Monta	539
Shigeo Hirose	509	Mike Montemerlo	487

Aaron Morris	189	Philippe Souères	549
Tatsuhi Mure	255	John Spletzer	529
Nobuyuki Nemoto	385	Pavel Staroverov	407
Eric Nettleton	179	Anthony Stentz	83, 167
Yuichi Ogawa	539	Gaurav S. Sukhatme	145, 297
Kazunori Ohno	125	Salah Sukkarieh	179, 299, 311
Zachary Omohundro	487	Jussi Suomela	395
Toshinaga Ozeki	385	Jackrit Suthakorn	437
Changwoo Park	355	Tsuyoshi Suzuki	41
Igor E. Paromtchik	41	Michel Taïex	549
Guilherme Pereira	529	Toshi Takamori	519
Ronald Peterson	529	Masayuki Takashima	519
Lars Petersson	367	Shiro Takashima	519
Cédric Pradalier	199	Hiroki Takeda	333
Erwin Prassler	71, 447	Koki Takiura	385
Libor Přeučil	135	Masao Takizawa	427
Jonathon C. Ralston	469, 477	Scott Thayer	189, 487
David C. Reid	469	Sebastian Thrun	179, 287, 417, 487
Bernd Reuse	31	Takahiro Tohyama	385
Peter Ridley	61	Takashi Tsubouchi	125
Jonathan M. Robert	51, 267, 459	Taiki Uehara	41
Nicholas Roy	417	Kane Usher	61
Daniela Rus	529	Nicolas Vandapel	103, 497
Dongseok Ryu	355	Zhi-Dong Wang	333
Jari Saarinen	395	Toshinobu Watanabe	225
Itsuro Saito	385	Carl Wellington	83
Takashi K. Saito	385	William Whittaker	189, 487
Srikanth Saripalli	277	Stefan Williams	93
Sakae Shibusawa	539	Graeme Winstanley	459
Bunji Shigematsu	125	Stuart Wishart	299
Nobutaka Shimada	427	Gordon Wyeth	267
Yoshiaki Shirai	427	Masatoshi Yamada	519
Moshe Shoham	323	Yoshio Yano	427
Steven Shooter	235	Sami Ylönen	377
Shraga Shoval	323	Kazuya Yoshida	225
Sajid Siddiqi	145	Shin'ichi Yuta	113, 125
Roland Siegwart	21, 235, 447	Alexander Zelinsky	367
Jouni Sievilä	377	Yu Zhou	437
Sanjiv Singh	529	Robert Zlot	167