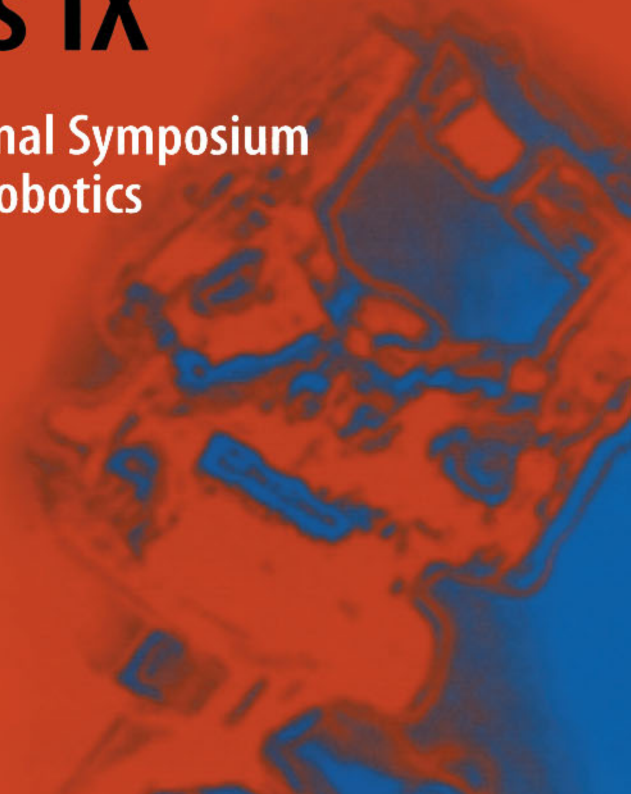
star springer tracts in advanced robotics 21

Marcelo H. Ang Jr.
Oussama Khatib (Eds.)

Experimental Robotics IX

The 9th International Symposium on Experimental Robotics



Springer

Springer Tracts in Advanced Robotics

Volume 21

Editors: Bruno Siciliano · Oussama Khatib · Frans Groen

Springer Tracts in Advanced Robotics

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

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-X]

Marcelo H. Ang Jr. · Oussama Khatib (Eds.)

Experimental Robotics IX

The 9th International Symposium on Experimental Robotics

With 332 Figures and 34 Tables

Springer

Editors

Dr. Marcelo H. Ang Jr.

National University of Singapore
Department of Mechanical Engineering
9 Engineering Dr 1
Singapore 117576
Singapore

Prof. Oussama Khatib

Stanford University
Robotics Laboratory
Department of Computer Science
Stanford, CA 94305-9010
USA

ISSN print edition: 1610-7438
ISSN electronic edition: 1610-742X

ISBN-10 3-540-28816-3 **Springer Berlin Heidelberg New York**
ISBN-13 978-3-540-28816-9 **Springer Berlin Heidelberg New York**

Library of Congress Control Number: 2005931592

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

Typesetting: Digital data supplied by editors.
Data-conversion and production: PTP-Berlin Protago-TEX-Production GmbH, Germany
Cover-Design: design & production GmbH, Heidelberg
Printed on acid-free paper 89/3141/Yu - 5 4 3 2 1 0

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 ill 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 some sixteen years ago, the *International Symposium on Experimental Robotics* (ISER) was published by Springer. Since the past edition, ISER has found a more suitable home under STAR, together with other thematic symposia devoted to excellence in robotics research.

The Ninth edition of *Experimental Robotics* edited by Marcelo H. Ang Jr. and Oussama Khatib offers in its fourteen-chapter volume a collection of a broad range of topics in robotics. The contents of these contributions represent a cross-section of the current state of robotics research from one particular aspect: experimental work, and how it reflects on the theoretical basis of subsequent developments. Experimental validation of algorithms, design concepts, or techniques is the common thread running through this large collection of widely diverse contributions.

From its warm social program to its excellent technical program, ISER culminates with this unique reference on the current developments and new directions in the field of experimental robotics – a confirmation for the series!

Naples, Italy
August 2004

Bruno Siciliano
STAR Editor

Preface

The Ninth Edition of Experimental Robotics is proud to present the latest in the cutting edge robotics research. From numerous submission containing high quality research, the review committee was forced to make difficult decisions to come up with a selection of 57 papers – which were presented in the 9th International Symposium on Experimental Robotics (ISER), 18–21 June 2004, Singapore. After the discussions during the Symposium, authors had a chance to revise their papers. This book contains the final version of the papers presented.

ISER is a series of bi-annual meetings which are organized in a rotating fashion around North America, Europe and Asia/Oceania. Previous venues were Montréal (Canada), Toulouse (France), Kyoto (Japan), Stanford (USA), Barcelona (Spain), Sydney (Australia), Honolulu (USA), and Sant'Angelo d'Ischia (Italy). Singapore is proud to host the 9th meeting of this active and prestigious symposium. The goal of ISER is provide a forum for research in robotics that focuses on novelty of theoretical contributions validated by experimental results. The meetings are conceived to bring together, in a small group setting, researchers from around the world who are in the forefront of experimental robotics research. This book presents the latest advances across the various fields of robotics, with ideas that are not only conceived conceptually but also explored experimentally.

This year, we are pleased to have a keynote address by Dr. Larry Matthies of the Jet Propulsion Laboratory which presented the interesting results of the Mars Rover Project. We have also included the keynote in this edition of Experimental Robotics as the *featured article*.

On behalf of the ISER Steering Committee, we would like to acknowledge the sponsorship of the ISER 2004 student fellowships from International Foundation of Robotics Research (IFRR) and Ben Wegbreit, and the support of the Singapore Exhibition and Convention Bureau and Lee Foundation. Most importantly, we would like to thank you for your participation and sharing of your exciting research developments.

With best regards,

Singapore, August 2004

Marcelo H. Ang Jr.
Oussama Khatib
Co-chairs, ISER 2004

Contents

IV. Localization and Map Building 97

V. Medical Robotics 151

VI. Manipulation 185

I. Keynote

Robots for space exploration have emerged more strongly than ever in recent years, with much limelight given to the success of Mars Exploration Rovers. Navigation, localization and map building in an unknown outdoor environment present a set of very challenging tasks even on the more familiar ground on Earth. The same set of problems, carried out in Mars, pushed the level of complexity to new heights – to which the team at Jet Propulsion Laboratory (JPL) of NASA has risen to answer the challenges. Larry Matthies' keynote address presents the various innovative strategies employed to successfully localize and navigate the landing modules and the rovers. The information rich feedback medium of vision is employed to guide the landing modules during Entry, Descent and Landing (EDL) phases and to assist the navigation of the Mars Exploration Rovers (MERs) to navigate autonomously, map terrain, and estimate motion in high slip areas. In this time where a lot of advanced robotics activities are still confined within the walls of laboratories or at least under the supervision of the robotics engineers, the rover project represents some of the highest achievements in autonomous robotics in real life, both in technical advancements and in robustness of the system. We hope to hear more of these 'adventures on the final frontiers' in the future.

Autonomous Navigation Results from the Mars Exploration Rover (MER) Mission

Mark Maimone, Andrew Johnson, Yang Cheng, Reg Willson, and Larry Matthies

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA, 91109, USA
firstname.lastname@jpl.nasa.gov

Abstract. In January, 2004, the Mars Exploration Rover (MER) mission landed two rovers, Spirit and Opportunity, on the surface of Mars. Several autonomous navigation capabilities were employed in space for the first time in this mission. In the Entry, Descent, and Landing (EDL) phase, both landers used a vision system called the Descent Image Motion Estimation System (DIMES) to estimate horizontal velocity during the last 2000 meters (m) of descent, by tracking features on the ground with a downlooking camera, in order to control retro-rocket firing to reduce horizontal velocity before impact. During surface operations, the rovers navigate autonomously using stereo vision for local terrain mapping and a local, reactive planning algorithm called Grid-based Estimation of Surface Traversability Applied to Local Terrain (GESTALT) for obstacle avoidance. In areas of high slip, stereo vision-based visual odometry has been used to estimate rover motion. As of mid-June, Spirit had traversed 3405 m, of which 1253 m were done autonomously; Opportunity had traversed 1264 m, of which 224 m were autonomous. These results have contributed substantially to the success of the mission and paved the way for increased levels of autonomy in future missions.

1 Introduction

Searching for signs of fossil or extant life is a major goal of Mars rover exploration. Given the central role of water in life as we know it, as well as its likely roles in the geology, climate, and resource inventory of the planet, “follow the water” has emerged as a key theme of Mars exploration. The MER mission sent rovers to the Gusev Crater and Meridiani Planum landing sites to attempt to verify that water had played significant roles in the history of these regions. The mission was designed for the rovers to be able to traverse about one kilometer in the course of a 90 day primary mission, operating for about four hours a day (one Mars day is about 24 hours and 40 minutes). They have far exceeded these goals, both in distance traveled and in lifetime. Given the round-trip communication latency between Earth and Mars of 20 minutes or more, this required autonomous navigation capabilities in the rovers. The Sojourner rover in the 1997 Mars Pathfinder mission used structured light for 3-D perception and obstacle avoidance with an extremely limited onboard computer [1]. With greater computational power and the need for more extensive traverse capability, the MER rovers use

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 3–12, 2006.

stereo vision and a more sophisticated local map and obstacle avoidance algorithm. Visual odometry for position estimation was not in the baseline software design, but was integrated as an “extra credit” item and has grown in importance as surface operations progressed. The DIMES system for horizontal velocity estimation was added late in the mission development phase to address a pressing need to increase the probability of safe landing. This paper gives an overview of the autonomous navigation capabilities of this mission, including the DIMES system used in EDL (section 2), the rover hardware design (section 3), the rover stereo vision and obstacle avoidance system (section 4), and the rover visual odometry system (section 5). Performance of these systems has generally been very good; we show examples drawn from surface operations.

2 Descent Image Motion Estimation System (DIMES)

As shown in Fig. 1, after dropping the heat shield, the MER descent system consisted of a parachute, the backshell from the interplanetary cruise stage, which also held retro-rockets for landing, and the lander itself, which was spooled out on a “bridle” to get it away from the rocket plumes. The retro-rockets (“RADs”, for rocket assisted descent) were designed to bring the vertical velocity to zero 10 to 20 m above the surface, after which the bridle was cut and the lander’s fall was cushioned with airbags. The backshell also incorporated a Transverse Impulse Rocket System (TIRS), which could fire laterally to ensure that the backshell was vertical before RAD firing, so that the RADs did not add to horizontal velocity.

During mission development, the EDL design team learned that steady-state winds near the martian surface could induce a lander horizontal velocity that

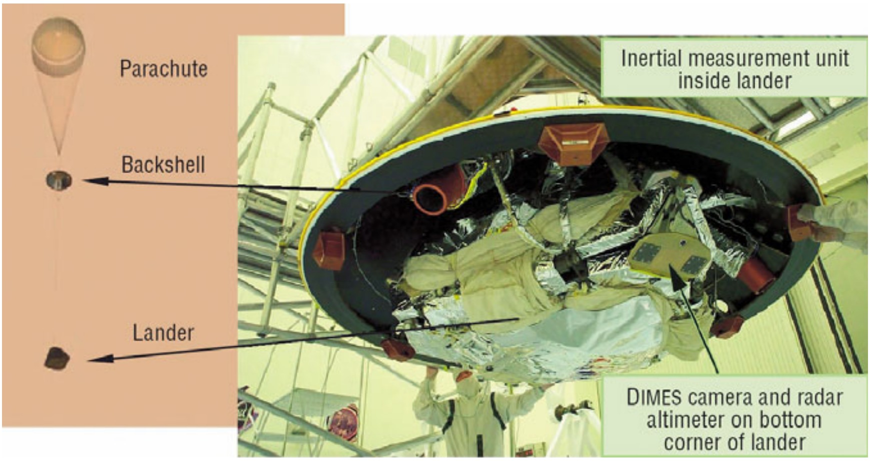

Parachute
Backshell
Lander
Inertial measurement unit inside lander
DIMES camera and radar altimeter on bottom corner of lander

Fig. 1. MER descent system. Left: artist’s illustration of the parachute, backshell, and lander in the last few km of descent. Right: photo of the underside of the lander, before integrating the heat shield.

might result in impact forces exceeding the strength of the airbags. The TIRS system could be adapted to tilt the backshell so that the RADs could offset the wind shear, if a horizontal velocity measurement was available. At the time this was learned, it was too late to incorporate a traditional doppler radar velocity sensor; however, it was possible to add a downlooking camera and software to use the camera to track terrain features to estimate the horizontal velocity. We briefly summarize the algorithm design of that system here; more detail appears in [2].

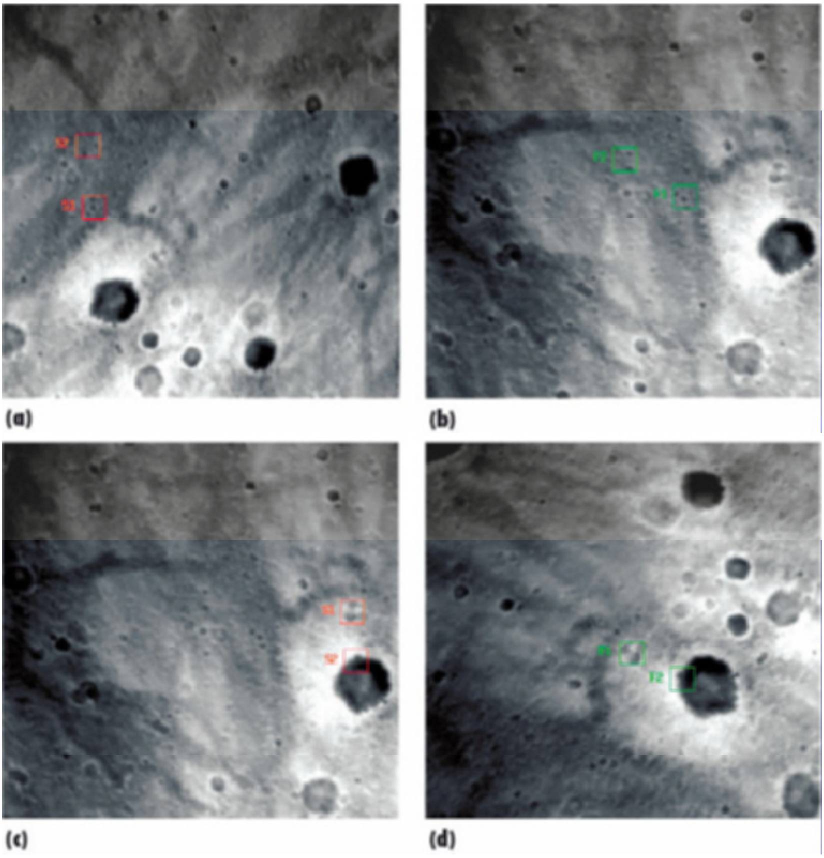


Fig. 2. DIMES results from the Spirit landing at Gusev Crater: (a) first descent image with selected features shown as the red squares; (b) second descent image with the matched features shown as the green squares; (c) second image again, showing the second set of selected features; (d) third image showing the matched features from the second image. All feature matches were correct.

The lander had an inertial measurement unit (IMU) and a radar altimeter that could measure angular velocity and vertical velocity. Thus, in principle, tracking one feature through two images would add enough information to estimate the entire velocity vector. Using just one feature is not very reliable, of course, but the onboard computer was too slow to do much more than this in real-time during descent. The following scheme was adopted to maximize reliability within the available computing resources. An interest operator picked two features in the first image, acquired at about 2000 m above the surface. A multi-resolution correlation algorithm matched those features in a second image, acquired about 1700 m above the surface. A variety of consistency checks were used to validate the matches. This gave one velocity estimate. Two more features were chosen from this image and matched in a third image, acquired at about 1400 m altitude, to give a second velocity estimate. These two velocity estimates gave an estimate for acceleration for the intervening interval, which was checked against accelerations measured with the IMU as a final consistency check. If all consistency checks passed, the velocity estimate was used to determine whether and how to use the TIRS to offset horizontal velocity.

DIMES determined that TIRS firing was needed for Spirit, but not needed for Opportunity. After-the-fact reconstructions of the landing events showed that without DIMES, the Spirit impact velocity would have been right on the edge of the tested airbag performance limits. Fig. 2 shows the descent images from Spirit, with the selected and tracked features for each pair of images.

3 Rover Hardware Overview

Before discussing the rover navigation capabilities, we give a brief overview of the rover hardware. More details are available press release material on the web [3]. Each vehicle weighs about 174 kg, has a wheelbase of 1.1 m, and is 1.5 m tall to the top of the camera mast. Locomotion is achieved with rocker bogie system very similar to that used in the 1997 Mars Pathfinder mission, with six driven wheels that are all kept in contact with the ground by passive pivot joints in the rocker bogey suspension. The outer four wheels are steerable.

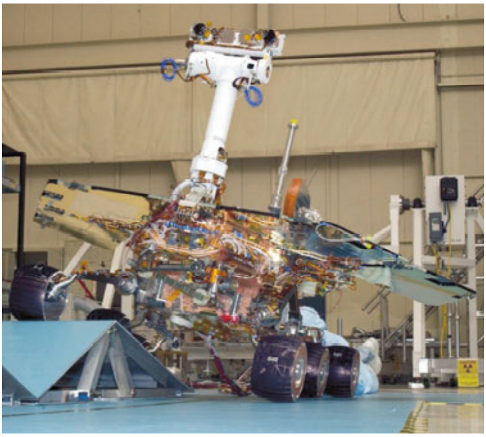


Fig. 3. One of the MER rovers in testing.

The rovers are solar powered, with a rechargeable lithium ion battery for nighttime science and communication operations. The onboard computer is a 20 MHz RAD6000, which has an early PowerPC instruction set, with 128 MB of RAM and 256 MB flash memory. Navigation is done with three sets of stereo camera pairs: one pair of "hazcams" (hazard cameras) looking forward under the solar panel in front, another pair of hazcams looking backward under the solar panel in the back, and a pair of "navcams" (navigation cameras) on the mast. All cameras have

1024x1024 pixel CCD arrays that create 12 bit greyscale images. The hazcams have a 126 degree field of view (FOV) and baseline of 10 cm; the navcams have a 45 degree FOV and baseline of 20 cm [4]. Each rover has a five degree of freedom arm in front which carries a science instrument payload with a microscopic imager, Mossbauer spectrometer, alpha/proton/x-ray backscatter spectrometer (APXS), and a rock abrasion tool (RAT). The camera mast has two additional science instruments: a stereo pair of "pancams" (panoramic cameras) and the "mini-TES" (thermal emission spectrometer). The pancams have filter wheels for multispectral visible and near-infrared imaging for mineral classification. They have the highest angular and range resolution of all cameras on the rover, with a 16 degree field of view and 30 cm baseline. The mini-TES acquires 167 bands between 5 and 29 μm in a single pixel. All cameras on the mast and the mini-TES are pointable by pan/tilt motors.

4 Obstacle Detection and Avoidance

The long round-trip communication latency between Earth and Mars and scheduling constraints on the Deep Space Network make it difficult to control long distance rover traverses from Earth, which necessitates some degree of onboard autonomous navigation to improve operational efficiency, reduce operations cost, and increase mission safety. However, the limited onboard computing power, the need for safety, and other factors constrain the level of sophistication in the algorithms that can be put onboard. Owing to budget and schedule constraints, the baseline autonomous navigation system includes only local obstacle avoidance with stereo vision; that is, there are no onboard global mapping, global path planning, or global localization functions. Stereo vision is used as the range sensor for obstacle avoidance because mature algorithms and reasonably compact, low-power, flight-qualified cameras are available for this, whereas flight-qualified versions of alternate sensors (e.g. ladar) with acceptable performance and form factor were not available.

4.1 Stereo Vision

Details of the stereo algorithm are described in [5]; here we give a brief overview of the algorithm, discuss some details that are specific to the MER implementation, and discuss its performance on Mars.

The stereo algorithm is a typical area-based algorithm using the sum of absolute differences (SAD) criterion for matching. Due to very slow readout from the flight cameras (about 5 seconds per frame for full resolution, 1024x1024 pixel imagery), images are generally binned vertically within the CCD cameras and read out at 256x1024 pixel resolution. This is reduced by averaging to 256x256. Stereo matching is then performed at 256x256 resolution. The images are rectified and bandpass filtered before the SAD operation, a variety of stereo matching consistency checks are applied after the SAD operation, subpixel disparities are computed by quadratic interpolation of SAD scores, and XYZ ranges images are produced as the final result. On the MER flight processor, this can take 24 to 30 seconds per image pair to compute.

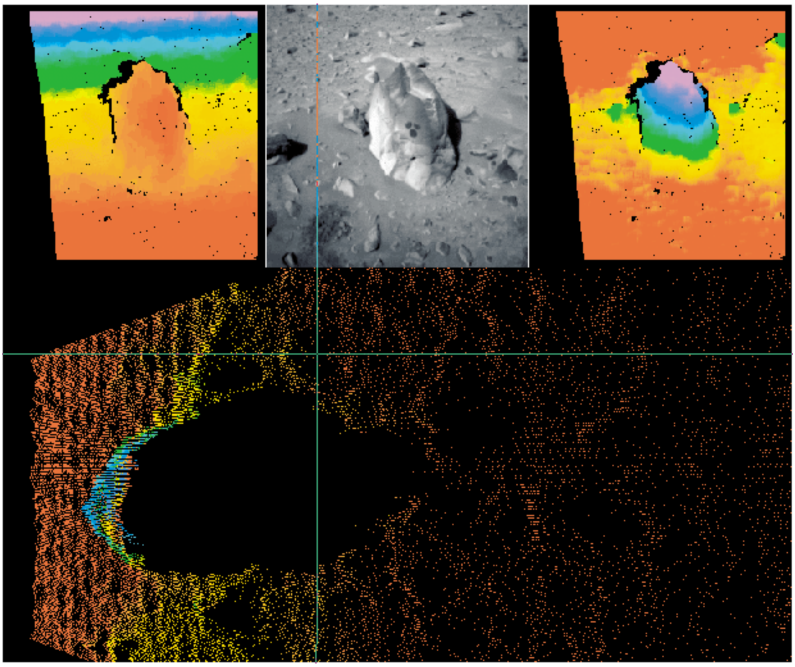


Fig. 4. Stereo results from the Spirit navcams, looking at Humphrey rock in Gusev Crater. The rock is about 0.5 m tall. Upper left: false color range image (red is closest, magenta is furthest). Upper right: false color height image (red is lowest, magenta is highest). Bottom: elevation plot, seen from above, where the cameras are at the left looking right (same color coding as the height image). Green crosshairs are not significant in this image.

Either the hazcams or the navcams can be used for autonomous navigation. The wide field of view (FOV) of the hazcams was designed to see more than the full width of the rover a short distance ahead of the rover, which is important for obstacle avoidance and to verify the safety of turn-in-place operations. However, the useful lookahead distance with the hazcams is at most 3-4 meters, due to their wide FOV and narrow baseline. The navcams can see further with their narrower FOV and wider baseline, but the FOV is only wide enough to verify the traversability of one candidate path several meters ahead of the vehicle. The pancams are not used for autonomous navigation.

Spirit uses the hazcams for stereo. Opportunity was unable to get acceptable range data with the hazcams, because the finer texture in the rock-free soil at Meridiani produced inadequate texture in hazcam images for stereo matching. Adequate results could be obtained with the navcams. Since Meridiani is largely obstacle-free, it has been sufficient to use navcam stereo to check the traversability of the nominal path forward and stop the vehicle if a hazard is detected.

Fig. 4 shows sample stereo results from the Spirit navcams looking at a rock that was studied by the science team.

4.2 GESTALT

The GESTALT obstacle avoidance algorithm is also described in detail in [5]; here we give a brief overview and discuss implementation and performance issues specific to MER. A higher level description of the overall rover driving software architecture appears in [6].

Range images from stereo are converted to "goodness" or "traversability" maps with 20 cm cells in a 10x10 m grid centered on the rover. For each range image, the complete set of range points is analyzed for traversability by fitting planar patches centered on each map cell in turn, where each patch is a circle with the diameter of the rover (nominally 2.6 m). The surface normal, RMS residual, and minimum and maximum elevation difference from the best fit plane determine a "goodness" factor for that map cell that characterizes its traversability. Goodness maps from each range image are registered and accumulated over time with the usual modulo map indexing arithmetic to avoid the need to scroll map data to keep the map bounded. Where new data overlaps old data, the new data overwrites the old data in the map. The merged goodness map is then used to evaluate traversability of a fixed set of candidate steering trajectories, which are circular arcs of varying radius. 23 forward arcs, 23 backward arcs, and two point turns are evaluated in each driving cycle. Evaluation amounts to adding up the goodness scores along each arc, with nearby cells given higher weight. The result is a set of traversability votes for all arcs. These votes are input to an arbiter, which also takes input from waypoints provided by human operators during mission planning. The rover drives a fixed distance along the winning arc before stopping to acquire new images for the next driving cycle. The distance per cycle is set by human operators at anywhere from 35 cm to 1 m or more depending a variety of operational factors, including terrain difficulty and overall distance goals for the day.

Typical computing time per cycle of GESTALT is around 70 seconds. While the rover is driving, its peak speed is 5 cm/sec, but it is typically operated at less than that (3.75 cm/sec) for power reasons. With computing time, the median net driving speed was about 0.6 cm/sec. Because this is so slow and the science team desired to cover large distances to Endurance Crater and the Columbia Hills, a hybrid daily driving scheme was designed in which human operators use navcam and pancam stereo imagery to plan each day's traverse manually as far as they can see it to be safe. The rover drives this segment blindly, then switches to autonomous navigation to drive for whatever time remains in the day. Blind drives can be several 10's of meters. The maximum daily traverse for Spirit up to June 14, 2004, was 124 m, of which 62 m were autonomous.

After some tuning of the algorithm to overcome excessively conservative behavior at slope changes, GESTALT has performed well. Fig. 5 shows some results of obstacle avoidance from Spirit. The terrain on the plains at Meridiani was benign enough that the only “obstacles” were occasional hollows from small, infilled craters. Within Eagle and Endurance Craters at Meridiani, the main navigation issues were slopes and slippage, not obstacles *per se*.

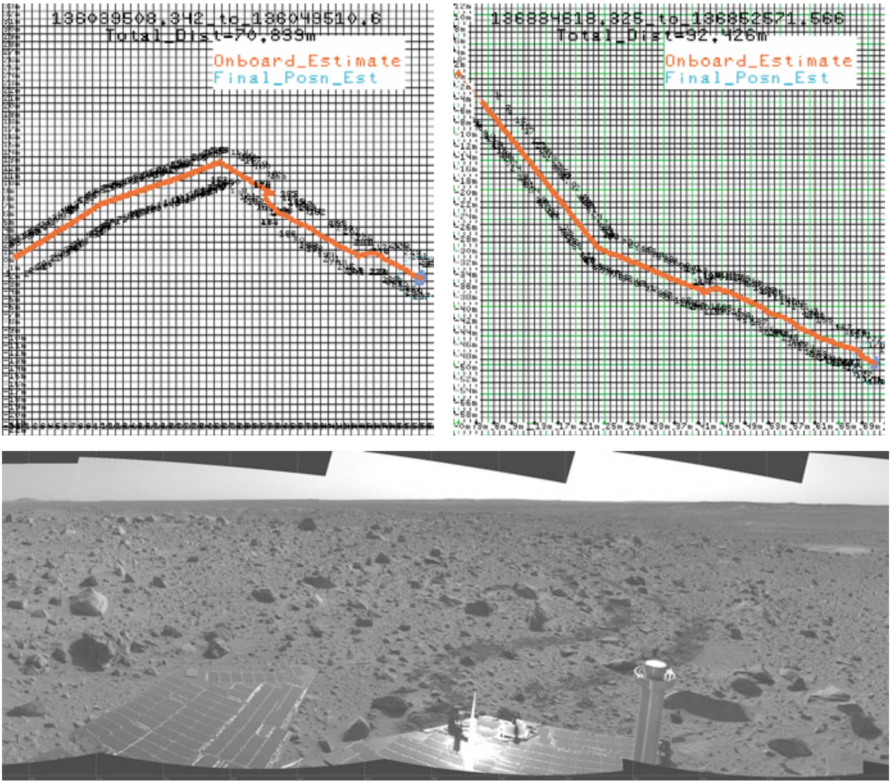


Fig. 5. GESTALT results from Spirit. Top: two examples of daily traverse plots that started with blind drives and ended with autonomous portions that included some rock avoidance maneuvers, covering total distances of 70.9 m (left) and 92.4 m (right). Bottom: a mosaic of navcam imagery looking back on a day’s traverse that ended with some rock avoidance, as can be seen from the rover tracks.

5 Visual Odometry

In routine operation, onboard position estimation is done by dead reckoning with the wheel encoders and IMU, with occasional heading updates by sun sensing with the pancams. Long distance localization is done on Earth using bundle adjustment from manually matched tie points in panoramic imagery [7]. On the plains at Gusev Crater, interpreting the bundle adjustment results as nominal ground truth, dead reckoning errors have been only a few percent of distance over several kilometers of travel. However, both rovers have experienced large slippage on slopes in the Columbia Hills, Eagle Crater, and Endurance Crater; in fact, up to 125% in one case in the Columbia Hills (ie. the rover slipped backwards in an attempted forward drive). Slip is becoming a bigger issue for Spirit as it attempts to drive with one wheel locked, due to likely imminent failure of that wheel's drive motor. Stereo vision-based visual odometry is part of the flight software, but was not routinely used in most of the mission to maximize driving speed. It has been used to assess slippage of Opportunity in Eagle Crater and is now being used by Spirit to measure and counter the effects of slip on slopes and of dragging the locked wheel.

Our visual odometry algorithm is described in detail elsewhere [8]. In a nutshell, it selects point features, uses multi-resolution area correlation to match them in stereo, tracks them in subsequent stereo pairs, and uses the tracking results to estimate the six degree of freedom rover motion between consecutive stereo pairs. This runs in about 160 sec. We have evaluated its performance against accurate ground truth on Earth-based rover testbeds and found that it can achieve 2% or better of distance over 30 m of travel. Fig. 6 shows sample results from Opportunity in Eagle Crater, where visual odometry correctly detected a 50% slip.

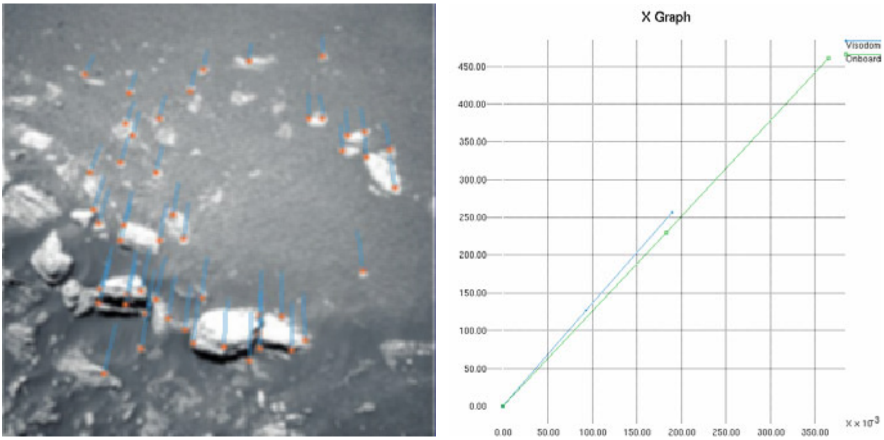


Fig. 6. Visual odometry results from Opportunity in Eagle Crater. Left: features selected (red dots) in an image near a rock outcrop; blue lines show optical flow to the feature position in the next image. Right: plots of onboard dead reckoned position versus visual odometry (in mm) for a case where visual odometry correctly inferred a 50% slip.

6 Discussion

The MER mission was the first use of stereo vision, local map-based obstacle avoidance, and visual odometry for autonomous rover navigation in a planetary exploration mission, as well as the first use of computer vision in an autonomous, real-time function (horizontal velocity estimation) during landing in a planetary mission. The algorithms are competent and emphasize reliability within the constraints a very slow onboard computer. Algorithms for analogous functions on Earth-based research vehicles can have more sophistication because of the much greater computing resources often available, but are rarely, if ever, designed or tested to reach the same level of fault tolerance. These algorithms have performed well and contributed to the success of the mission; in particular, DIMES may have saved Spirit from a disastrous landing.

In this mission, the most valuable science results have been found in rock outcrops in sloping, slippery terrain, either inside craters or on hills; moreover, the rovers have had to drive much further than anticipated prior to landing to reach such outcrops. Thus, for future missions, key issues include increasing the speed of the vision and planning algorithms and integrating visual odometry inside the driving and steering loop to enable safe, efficient traversal on slippery terrain. Incorporating path planning algorithms for longer lookahead and rougher terrain will also be valuable. For landers, the focus of vision algorithm research will switch to enabling precision landing.

References

[1] Wilcox, B., and Nguyen, T. Sojourner on Mars and Lessons Learned for Future Planetary Rovers. 28th Int'l Conf. on Environmental Systems, Danvers, MA, July 1998.

[2] Cheng, Y., Goguen, J., Johnson, A., Leger, C., Matthies, L., San Martin, M., and Willson, R. The Mars Exploration Rovers Descent Image Motion Estimation System. IEEE Intelligent Systems Magazine, May/June 2004.

[3] http://marsrovers.jpl.nasa.gov/newsroom/presskits.html.

[4] Maki, J., et al. Mars Exploration Rover Engineering Cameras. Journal of Geophysical Research, vol. 108 E12, December 2003.

[5] Goldberg, S., Maimone, M., and Matthies, L. Stereo Vision and Rover Navigation Software for Planetary Exploration. IEEE Aerospace Conference, Big Sky, Montana, March 2002.

[6] Maimone, M. and Biesiadicki, J. The Mars Exploration Rover Surface Mobility Flight Software: Driving Ambition. IEEE Aerospace Conference, Big Sky, Montana, March 2005.

[7] Li, R., Di, K., Matthies, L., Arvidson, R., Folkner, W., Archinal, B. Rover Localization and Landing Site Mapping Technology for the 2003 Mars Exploration Rover Mission. Journal of Photogrammetric Engineering and Remote Sensing, January 2004.

[8] Olson, C., Matthies, L., and Schoppers, M. Rover Navigation using Stereo Ego-motion. Robotics and Autonomous Systems, 43(4), June 2003.

II. Human-Centered Robotics

Since the day the concept of robotics was born, robots were intended to assist humans in their daily lives. Though initially implemented in the well-structured areas such as the factories, where humans are kept out of their workspace, robots are slowly finding its way into human world. More and more emphasis has been placed on having robots come into direct contact with human activities in human world. This trend is happening across the entire spectrum of robotics research frontier, although the articles in this chapter are just the few more obvious examples.

Mori *et al* proposes a method of recognizing human action from the motion data input. This is an interesting and innovative way of human-machine interaction where the algorithm extracts motion features that are "remarkable", and then identifies certain human actions, such as: walking, sitting, raising hand, etc. This capability would certainly be useful in making present robots more interactive with humans. Learning is achieved using Support Vector Machines. Good results of identification have been obtained.

Burdet *et al* presents work to combine (functional Magnetic Resonance Imaging) fMRI technology with haptics technology. The requirement is a haptic system that is not disturbed by the harsh environment of the fMRI, and at the same time, it should not interfere with the activity of the fMRI. Add onto the list the inherent requirement for a haptic device to deliver a realistic kinestatic feedback of the environment of interest, and we have a very interesting set of problems that is being addressed by this collaborative international team. An electrostatic master-slave system is used to power the robot outside the MR shielded room, and optical sensors are used to provide haptics feedback.

Minimally invasive treatment of the human body is an area with strong interest and growth. Reducing the need for extensive surgery not only reduces the risk of the treatment, but also allows quick recovery of the patient and shorter hospital stay, which also works out to considerable financial savings. The article by Webster *et al* introduces a new idea of a bendable and steer-able needle – an injection needle that is capable of working its way around other organs or bodily parts to reach the exact spot within a human body to administer the treatment. A non-holonomic model is developed and validated experimentally.

The last article in this chapter, by Kagami *et al*, presents work to improve human-machine interaction by utilizing microphone arrays to capture sound signals, and to detect its direction and position of the origin. The ultimate goal of their work is to recognize human speech and to be able to localize the position of the human speaker.

SVM-Based Human Action Recognition and Its Remarkable Motion Features Discovery Algorithm

Taketoshi Mori[1], Masamichi Shimosaka[2], and Tomomasa Sato[3]

[1] Interfaculty Initiative in Information Studies, The University of Tokyo,
http://www.ics.t.u-tokyo.ac.jp/, tmori@ics.t.u-tokyo.ac.jp
[2] Mechano-Informatics, The University of Tokyo, simosaka@ics.t.u-tokyo.ac.jp
[3] Mechano-Informatics, The University of Tokyo, tomo@ics.t.u-tokyo.ac.jp

Abstract. This paper proposes a discovery algorithm of knowledge of remarkable motion features in SVM-based action recognition. The main characteristics of the proposed method are a) basic scheme of the algorithm is based on Support Vector Learning and its generalization error, b)remarkabale motion features are discovered in response to kernel parameters optimization through generalization error minimization. Experimental results show that this proposed algorithm makes the recognition robust and finds remarkable motion features that are intuitive for human.

1 Introduction

Recognizing human actions has potential to contribute to intuitive communication between human and machine, such as human-computer interaction, search engine for multi-media databases, or intelligent video editing. It may be applied to design some level of humanoid actions efficiently.

It is proper to divide the process of action recognition into the following two phases. The former is to get time series of 3D body motion structurally from some instruments. The latter is to symbolize these kinds of motion to action names. There are many researches which use video image as input. For example, Starner et al. constructed a sign language recognition based on HMM[1], and Wilson et.al made a gesture recognition system[2]. However, usually the main work of such researches is on how to acquire motion robustly, since the time series image processing is still a difficult problem.

On the other hand, our approach concentrates on the latter part. We use structured motion capture data as input, and focuses on ’recognition’ part. As for the recognition method, we have been developing a daily life action recognizer which recognized daily life action such as walking or sitting based on Support Vector Machine[3].

The main important characteristics of the method is to utilize expressions of action described based on human knowledge. This enables aids for the designer of the recognition method to detect remarkable motion features in motion candidates. But, it is often difficult to generate expressions for actions. And, there is some difficulty of selecting or evaluating relevant motion feature from the described expressions for the actions. Therefore, this paper proposes an algorithm which discovers remarkable motion features for a certain action automatically based on many corresponding

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 15–24, 2006.

motion captured data. The algorithm finds out which motion features are important for which action to be recognized.

In section 2, the action recognition method as the basis of the proposed algorithm will be noted. The qualitative and detailed description of our knowledge discovery scheme, which utilizes kernel parameters optimization, will be explained in section 3. The performance evaluation of the proposed algorithm will be described in the final section.

2 HARS: Daily Life Action Recognition System

As for the basis of the proposed algorithm, we constructed a SVM-based action recognition method. It takes over the characteristics of Shimosaka et al.'s system[3] and Mori et al.'s[4,5] mentioned below.

First feature is *Simultaneous Recognition*. This is because human can recognize multiple action at the same time in parallel. For example, human can readily recognize someone waving his or her hand while walking as "waving one's hand and walking". Second feature is *Expressing Vagueness in Recognition*. This is because human cannot always give absolute decision whether some action really occurs or not when watching someone acting. For instance, decision whether lying or not made by human may contain unclarity on observing someone getting up. Third feature is *Utilization of Action Expressions by Human*. This has arisen from feature selection problem, because such remarkable features for recognizing some action as motion and pose of body region have wide variation. For example, forward motion of hips could be one of the features for walking, meanwhile the direction of head is considered as irrelevant. Human uses only relevant features of the recognition target actions. And human can easily express an action by representing the motion or the pose of body parts, a designer of the system selects input motion with aid of these expressions. Consequently, our system is designed to be able to output multiple action name labels at the same time and to output not only decisive result, but unclarity result based on action expressions made by human[4,5]. Because this manual fashion approach invokes some critical problem noted above, the necessity of this research proposing automatic detector becomes clear.

The criterion of recognition accuracy in this kind of research is difficult since daily actions have no distinct definition. Thus, the correctness of recognition results are basically based on human's judgment. We defined the correctness of a recognition result is to be generated by humans from moment to moment with some motion as decision result whether one specific action occurs or not. On observing someone acting, even if a person is performing several actions simultaneously, the reference of recognition result is written by paying attention to this action only. For instance, if the target action is assigned as "walking", human pays attention to "walking". Thus, the reference is equal to the result whether someone is walking, even if he or she actually "walking" and "knodding" at the same time. This means, multiple action can be recognized at the same time (For example: Running while waving hands).

The input of the system is time series of human motion. Our system utilizes articulated human body motion whose three-dimensional configuration is recovered. The output of our system contains some action names in synchronized with a frame of the input motion.

2.1 Processing Configuration

Figure 1 shows the processing configuration of the method. To realize simultaneous recognition, the method contains multiple recognition processes, each of which is assigned to a certain action's recognition. This primitive recognition process for one action is called as Action Element Recognizer in this research. The process of each Action Element Recognizer runs in parallel with other Action Element Recognizers. The method collects the results of all processes and outputs the results of each recognition process per frame.

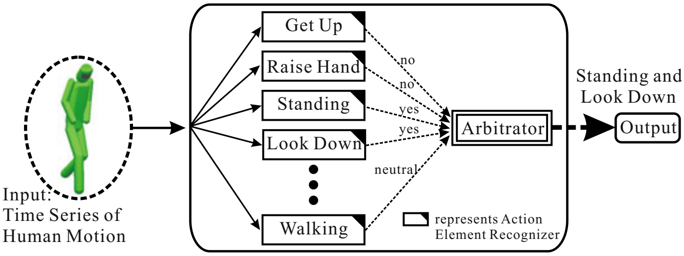


Fig. 1. Configuration of Recognition Method

New action can be recognized simply by adding the corresponding recognizer to the method. This means, the result of one Action Element Recognizer process is independent of the others.

2.2 Kernel Based Action Element Recognizer

An Action Element Recognizer outputs the recognition result whether the assigned action occurs or not per frame in sync with the input motion. The output consists of multiple classes which represent not only decisive but also inexplicit result. Concretely speaking, the number of the class category is 3, yes, no and neutral.

The Action Element Recognizer has two binary classifiers and outputs integrated result of the two binary values. The one binary classifier judges whether "yes" or not-"yes", and the other judges whether "no" or not-"no". The configuration of Action Element Recognizer is shown in Figure 2. Since"neutral" category rarely happens in some actions, we have adopted this type of composition.

In this research, this each Action Element Recognizer's classifier is implemented using Kernel classifier. We denote by x the time series of input motion. $D =$

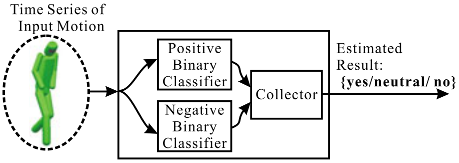


Fig. 2. Configuration of Action Element Recognizer

$\{\boldsymbol{x}_i, y_i\}_{i=1}^{l}$ are the input-output pairs in total l frames, where $\boldsymbol{x}_i$ represents i th frame sample motion and its corresponding reference binary(e.g. "yes" or not-"yes") signal by y_i. We can write by α_i the co-efficiencies whose value is proportional to importance of the templates. Similarity value calculated by Kernel $K(\cdot,\cdot)$ between the input motion and one template motion is represented by $K(\boldsymbol{x}_i, \boldsymbol{x})$. The mapping between the input and the output of the binary classifier in the Recognizer f can be written as $f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b\right)$, where b depicts offset and the function $\operatorname{sgn}(\cdot)$ is a step function where the output is 1 if input is more than 0, and otherwize -1.

The learning process in the classifier of Recognizer gets the co-efficiencies($\boldsymbol{\alpha}, b$) from the training data. Support Vector Machine is utilized as learning method for it. Support Vector Machine has such advantages as good model selection, well regularization and small computational resources.

The kernel value in the binary classifier is the product of all the kernel values corresponding to the similarity in each target motion. When the number of the target motion in the target action is d, the kernel value in the target action $K(\cdot,\cdot)$ can be written as$K(\boldsymbol{x}, \boldsymbol{x}_i) = \prod_{j=1}^{d} K_j(\boldsymbol{\varphi}_j(\boldsymbol{x}^{(j)}), \boldsymbol{\varphi}_j(\boldsymbol{x}_i^{(j)}))$ where $\boldsymbol{x}^{(j)}$ denotes the selected input motion in the j th target motion, $\boldsymbol{\varphi}_j(\cdot)$ represents the converter from the selected input motion to the input feature, and the kernel value which corresponds to the similarities in j th target motion represented by $K_j(\cdot,\cdot)$. Radial Basis Function (RBF) is utilized as the kernel for target motion. Thus the final form of the kernel is represented as

$$K(\boldsymbol{x}, \boldsymbol{x}_i) = \exp\left(- \sum_{j=1}^{d} \frac{\left| \boldsymbol{\varphi}_j(\boldsymbol{x}^{(j)}) - \boldsymbol{\varphi}_j(\boldsymbol{x}_i^{(j)}) \right|^2}{\sigma_j^2} \right). \tag{1}$$

Figure 3 shows an example recognizer which uses several kernel classifiers, which are denoted K_i in the figure.

3 Remarkable Motion Features Discovery based on Kernel Parameters

3.1 Remarkable Motion Features Discovery

The main result of this research is that it treat the sensitivity of the kernel value with respect to change of input as relevance of selection of motion features for

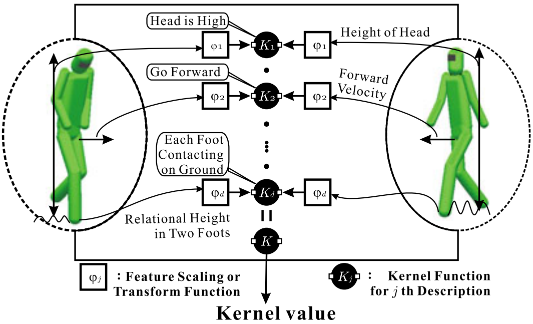

Head is High
Height of Head
φ1
K1
φ1
Go Forward
Forward Velocity
φ2
K2
φ2
Each Foot Contacting on Ground
φd
Kd
φd
Relational Height in Two Foots
K
φj : Feature Scaling or Transform Function
Kj : Kernel Function for j th Description
Kernel value

Fig. 3. Example: Kernel Value in Binary Classifier of 'Walking' Recognizer

recognition. It means, the smaller sensitivity of the kernel value might be desired in the case of less importance for recognition. As for RBF kernel, the remarkable input feature requires smaller variance in relevant input feature than irrelevant.

If it is required to know which input feature is relevant, human judges which input motion is remarkable by observing the kernel parameters. In this research, the kernel parameters optimization which adjusts kernel sensitivity is used in order to discover remarkable motion features and to optimize recognition performance. In the case of our method, the proposed algorithm adjusts the variances in Eq.(1) and trade off positive number in SVM.

3.2 Kernel Parameters Optimization

The generalization error is utilized as the indicator of the kernel parameter optimization. The optimized kernel parameters $\boldsymbol{\theta}^*$ is defined as $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} T(K_{\boldsymbol{\theta}})$, where $\boldsymbol{\theta} \in \boldsymbol{R}^{d_k}$ denotes the kernel parameters, $T(K_{\boldsymbol{\theta}})$ depicts the generalization error of the SVM. The gradient descent algorithm is utilized in this research to optimize kernel parameters easily and robustly. General outline of the kernel parameters optimization algorithm is listed as Table 1.

Table 1. Kernel Parameters Optimization

1. Initializes $\boldsymbol{\theta}$ with some value and iteration number i as 0
2. Learns by SVM with $K_{\boldsymbol{\theta}}$, finding co-efficiencies.
3. Calculates generalization error T and its derivative.
4. Updates the kernel parameter $\boldsymbol{\theta}$ such that $T(K_{\boldsymbol{\theta}})$ is minimized as $\Delta\boldsymbol{\theta} = -\epsilon\partial T/\partial\boldsymbol{\theta}$, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$,$i \leftarrow i + 1$. ($\epsilon > 0$)
5. In the case that $|\Delta\boldsymbol{\theta}|$ is less than some positive constant or i is larger than some iteration times, then terminates, otherwise returns to **2.**

There are several good estimators for the performance of SVM. In this research, the Span technique proposed by Chapelle et al.[8] is adopted. This is because the gradient descent technique requires the derivative function of generalization error by kernel parameters and it can be explicitly written if estimator is based on Span. Span technique requires less computational resource than the case of Leave-One-Out.

After learning by SVM, the Span corresponding to the p th support vector by the variable S_p can be defined as the distance between the $\boldsymbol{\phi}(\boldsymbol{x}_p)$ and linear combination Λ_p by all the support vector except the p th support vector in feature space $\boldsymbol{\phi}(\cdot)$ as

$$\Lambda_p = \left\{ \sum_{i \neq p, \alpha_p > 0} \lambda_i \boldsymbol{\phi}(\boldsymbol{x}_i), \sum_{i \neq p, \alpha_p > 0} \lambda_i = 1 \right\} \tag{2}$$

$$S_p^2 = \min_{\boldsymbol{\phi}(\boldsymbol{x}) \in \Lambda_p} ||\boldsymbol{\phi}(\boldsymbol{x}_p) - \boldsymbol{\phi}(\boldsymbol{x})||^2 = \frac{1}{(\tilde{K}_{sv}^{-1})_{pp}}. \tag{3}$$

where function $\boldsymbol{\phi} : \mathcal{X} \rightarrow \mathcal{F}$ ($\mathcal{F}$ represents some feature space) satisfies $\boldsymbol{\phi}(\boldsymbol{u})^t \boldsymbol{\phi}(\boldsymbol{v}) = K(\boldsymbol{u}, \boldsymbol{v}), (\boldsymbol{u}, \boldsymbol{v} \in \mathcal{X})$ and $\tilde{K}_{sv}$ corresponds to extended Gram Matrix of all the support vectors. The upper bound of the generalization error based on the Span is defined as

$$T_u = \frac{1}{l} \sum_{p=1}^{l} \Psi(\alpha_p S_p^2 - 1) \tag{4}$$

where $\boldsymbol{\alpha}$ denotes the co-efficiency obtained by SVM, Ψ depicts step function to penalize.

Because the optimization process requires the derivative by the kernel parameters, the gradient of Ψ, $\boldsymbol{\alpha}$, S_p^2 must be calculated. The probabilistic approach by approximating the step function with sigmoid function[9] is adopted. The gradient of the co-efficiency $\boldsymbol{\alpha}$ can be calculated because the relation between output and input of the SVM can be written only by support vectors. The computation for the derivative of S_p^2 can be calculated by Woodbury Theorem. This is derived as

$$\frac{\partial S_p^2}{\partial \theta_q} = S_p^4 \left(\tilde{K}_{sv}^{-1} \frac{\partial \tilde{K}_{sv}}{\partial \theta_q} \tilde{K}_{sv}^{-1} \right)_{pp} \tag{5}$$

where θ_q represents the q th kernel parameter.

4 Experiments

4.1 Target Action and Motion Candidates

Standing, Folding arms etc. from ICS Action Database[6] are used to evaluated the performance of the proposed method. The database is a collection of motion data with full reference action name labels. The format of motion is BVH which is defacto standard computer graphics motion format. A BVH file contains the structure

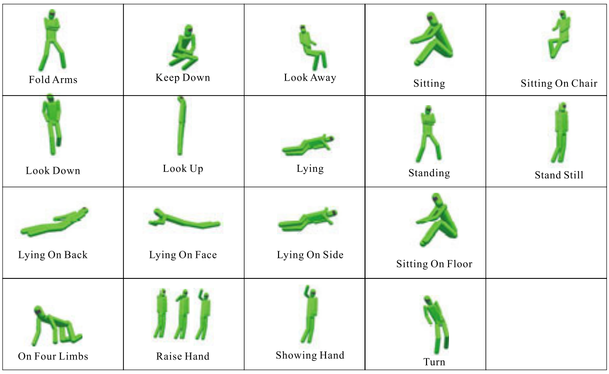


Fig. 4. Target 18 Daily Life Actions

of a human as a linked joint model and the time-series motion data of the joint model per frame(usually 30Hz). Total of 18 actions are used(Fig4).

The motion information used as feature candidates are listed in Table 2 . The bracket after candidates(right side of the table) represents the duplication of candidates, each of whose Span is different.

For the experiments, 125 BVH files and 2250 reference answer action label files are used.

Table 2. Given Motion Information List

1	Relative horizontal position of right hand to left	**2**	Sum of distance between hands and body
3	Mean speed of hands	**4**	Bentness of Hips
5	Mean speed of hips(1)	**6**	Height of hips
7	Upper direction of head from hips	**8**	Distance between hips and foots
9	Upper direction of hips from foots	**10**	Horizontal orientation of head from hips
11	Upper orientation of head from hips	**12**	Upper orientation of head from ground
13	Mean height of head(1)	**14**	Height of hips
15	Upper orientation of hips	**16**	Horizontal orientation of hips
17	Upper direction of head from left hand	**18**	Upper direction of head from right hand
19	Upper direction of hips from left knee	**20**	Upper direction of hips from right knee
21	Height of left hand	**22**	Height of right hand
23	Relative height of left hand from hips	**24**	Relative height of right hand from hips
25	Mean upper velocity of left hand	**26**	Mean upper velocity of right hand
27	Highest relative height of hands from head	**28**	Mean height of head(2)
29	Mean speed of hips(2)	**30**	Mean speed of left foot
31	Mean speed of right foot	**32**	Speed of Rotation of hips in vertical axis

4.2 Given Parameters and Condition

As the initial parameters of this experiment, RBF kernel parameters and trade-off number in SVM C are given as $\sigma_j = 1.5\sqrt{f_d}\ (1 \leq j \leq f_d),\ C = 5\sqrt{f_d}$ in all 18 action names, where f_d denotes the dimensionality of the kernel input space (i.e. 32).

The Maximum iteration times of the gradient descent procedure is set as 30. As the updating rate ϵ of the second procedure in Table 1 is set as $\epsilon_\sigma = 0.05$ for kernel parameters and $\epsilon_C = 0.1\epsilon_\sigma$ for penalty term.

4.3 Results of Applying Proposed Method

Figure 5 shows that error rate the before and after the kernel parameters optimization in the case that all the 32 motion features are utilized. Error rate in the case of the initial given kernel parameters are represented by the square points. Circle points depict the case of the optimized kernel parameters and cross points represent the score reported in our previous research [3]. As the recognition accuracy result, each target action achieves 80[%]. Especially, the accurate rate in 14 out of 18 action names are larger than than 90[%].

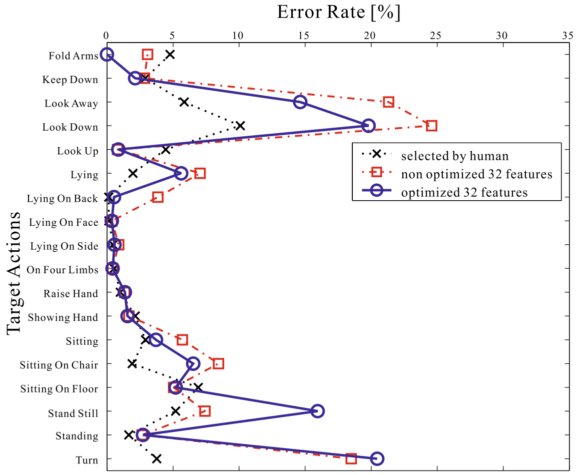


Fig. 5. Error Rate Before and After Kernel Parameters Optimization

In almost all the target actions, the accuracy gained by the optimized kernel parameters is better than the case of the initial kernel parameters. The accurate rate reported in our previous research [3] is better than the proposed method. This result shows that the strategy which utilizes expression of action works good if expressions can be generated easily.

Figure 6 and 7 shows the relative ratio of the kernel parameters after the procedure of the proposed algorithm in the case of "Standing" and "Lying on side", respectively. In each figure, the number on the horizontal axis corresponds to the number in Table 2. The vertical axis shows the normalized inverse of the variances whose maximum value is 1. In the scheme of the proposed detection method, the larger value in the vertical indicates more relevant motion feature, because smaller kernel parameter σ makes kernel more sensitive.

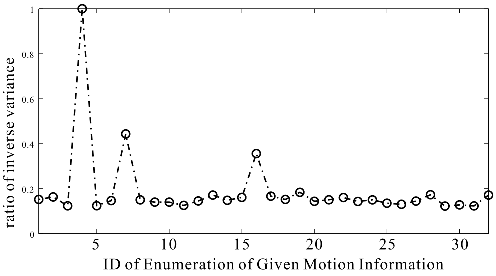


Fig. 6. Normalized Inverse Variance after Optimization in “Standing”

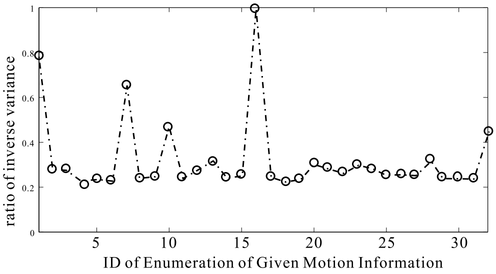


Fig. 7. Normalized Inverse Variances after Optimization in “Lying on Side ”

In the the case of "Standing", inverse of the variance corresponds to bent of hips is largest. Next, the orientation of the upper body is detected, and horizontal posture of hips is detected as the third relevant motion feature. This result fit with human intuition.

As for “Lying on Side”, the proposed algorithm detects horizontal orientation of hips, upper direction of head from hips and relative horizontal position of right hand to left.

These results show that the relation between the gained ratio of the kernel parameters and motion in the training data seems to be natural.

5 Conclusion

This paper proposes an algorithm for discovering remarkable motion features from candidates of motion features in human daily life action recognition based on Support Vector Machine. This discovery algorithm is based on kernel parameters optimization as minimization of generalization error. In this paper, Span based generalization error is utilized. The value can be calculated effectively and has close relationship to Leave-One-Out error. This paper adopts gradient information because search method for kernel parameters space must be effective.

The experimental results for performance evaluation show that the accuracy of recognition reaches high enough, and the performance after the optimization of the kernel parameters is better than the case of the initial settings. It is also proved that the relative importance values which corresponds to inverse of the variances (the kernel value in RBF kernel) fit with human intuition.

In the future work, the validity of our discovery algorithm will be evaluated by analizing relationship between sensitivity of kernel functions and recognition accuracy. Also, systematic method for listing motion candidates will be considered, since this paper just lists the candidates of the motion features utilized in our previous research[3]. And, analysis between density in input space and its sensitivity of kernel feature space will be explored, because the proposed algorithm initializes the parameters which satisfy the range of kernel value is the same in each feature.

References

1. T. Starner et al. Visual Recognition of American Sign Language Using Hidden Markov Models. In International Workshop on Automatic Face and Gesture Recognition. pp.189-194, 1995.
2. A. Wilson et al. Parametric Hidden Markov Models for Gesture Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.21, No.9, pp.884-900, 1999.
3. M. Shimosaka et al. Recognition of Human Daily Life Action and Its Performance Adjustment based on Support Vector Learning. In *CD-ROM of 3rd IEEE Intl. Conf. on Humanoid Robots*, 2003.
4. T. Mori et al. Human-Like Action Recognition System on Whole Body Motion-Captured File. In Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2066-2073, 2001.
5. T. Mori et al. Human-like Action Recognition System Using Features Extracted by Human. In *Proc. of the 2002 IEEE/RSJ Intl. Conf. on IROS*, pp. 1214–1220, 2002.
6. B. Schölkopf et al. *Learning with Kernels*. MIT Press, 2002.
7. J. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Scholkopf et al., editor, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.
8. O. Chapelle et al. Model Selection for Support Vector Machines. In *Advances in Neural Information Processing Systems 12*, pp. 230–236. MIT Press, 2000.
9. J. Platt. Probabilities for SV Machines. In *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press, 1999.
10. T. Mori et al. ICS Action Database. http://www.ics.t.u-tokyo.ac.jp/action/, 2003-2004.

fMRI Compatible Haptic Interfaces to Investigate Human Motor Control

Etienne Burdet[1], Roger Gassert[2], Ganesh Gowrishankar[1], Dominique Chapuis[2] and Hannes Bleuler[2]

[1] Department of Mechanical Engineering and Division of Bioengineering,
National University of Singapore
http://guppy.mpe.nus.edu.sg/∼eburdet e.burdet@ieee.org

[2] Laboratory of Robotic Systems,
Swiss Federal Institute of Technology Lausanne (EPFL)
http://lsro.epfl.ch/

Abstract. We are developing haptic interfaces compatible with functional magnetic resonance imaging (fMRI) to study the brain mechanisms of motor control in humans. This paper describes the different phases of our project, examines the constraints and presents possible solutions. The constraints imposed by the harsh yet sensitive MR environment as well as the smooth and safe control required for interaction with human motion demand a novel robotic technology. Our concept is based on a hydrostatic master-slave system used to power the robot near the scanner from outside the shielded MR room. Force/torque and position sensors measure the deflection of an elastic polymer probe via light intensity measurement over fiber optics, thus allowing all electronic components to be placed outside the MR room. This concept was validated through two interfaces able to provide force and motion feedback simultaneously with imaging.

1 Introduction

Ten years ago, the Laboratory of Robotic Systems of the EPFL developed the first comprehensive robotic system for neurosurgery using computer tomography (CT) for guidance [7]. This robot performed fourteen operations in the Lausanne University Hospital in Switzerland between 1993 and 1995, all of which were successful [5]. In the meanwhile, Magnetic Resonance Imaging (MRI) has established itself as a standard diagnostics and advanced brain research tool. MRI has a fine spatial resolution, is well suited for visualization of soft tissues, and does not use ionizing radiation or injection of radioactive liquid [14]. A next challenge will consist of migrating MRI from the diagnostic radiology to the operating room. MR compatible robots guided by real-time 3D imaging could revolutionize surgery, enabling more reliable and precise minimally invasive interventions with minimal recovery time.

Functional MRI or fMRI is an excellent indicator of cerebral activity and has allowed significant advances in neuroscience [9]. On the other hand, haptic interfaces [1,8,11] can dynamically interact with humans performing movements and deliver forces fast and smooth enough to study neuromuscular response. Investigating adaptation to virtual dynamic environments produced by such interfaces has brought major advances in neuroscience [13,2] (Fig.1). A robotic interface in conjunction

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 25–34, 2006.

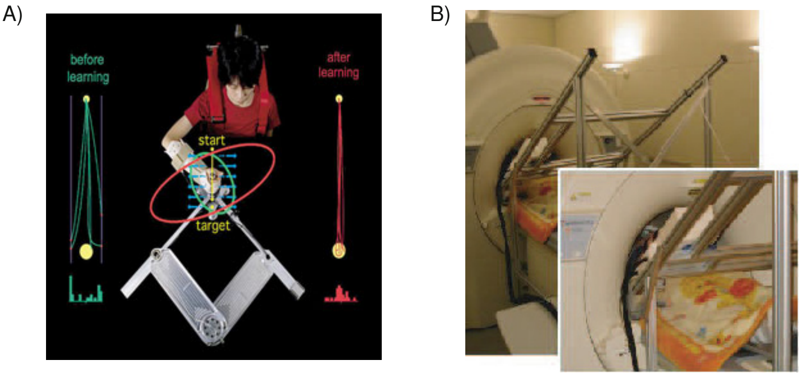


Fig. 1. We infer human motor control by examining the effect on motion and the adaptation to computer-controlled dynamics produced by a haptic interface. (A) Pictorial representation of the finding that the central nervous system stabilizes unstable dynamics by learning optimal impedance [2]. (B) fMRI compatible haptic interface installed at ATR in Japan.

with fMRI has great potential: it would enable neuroscientists to 'view' and investigate the brain mechanisms involved in performing tasks with arbitrary dynamics. This could become a critical tool in neuroscience and rehabilitation.

There is, however, a major problem for robots working within an MR environment: conventional materials, actuators and sensors interfere with the static magnetic field, static and switching magnetic field gradients and radio frequency (RF) signals generated by the scanner. Safety is crucial as a robot working within an MR environment is exposed to a strong static magnetic field of 1.5 to 3 T in current scanners (up to 7 T body scanners available for research). The robot must therefore be insensitive to the imaging sequence and should not disturb the imaging itself. fMRI scanning sequences are even more sensitive to inhomogeneities of the magnetic field than MRI sequences. Another constraint is the limited space inside the scanner bore, a narrow tube of about 60cm in diameter, also occupied by the subject.

Possible techniques for MR compatible robots can be found in [10]. AIST in Japan has developed a five degrees-of-freedom robot using ultrasonic actuators to pilot a catheter for brachytherapy of prostate cancer under MRI [4]. However, this robot is created for low field open scanners, the actuators are placed outside the magnet bore and this application does not require interaction with human motion.

Our goal is to design fMRI compatible haptic interfaces, i.e. robotic systems with force-feedback which can safely interact with human subjects and operators, do not disturb the particularly sensitive functional imaging and are not disturbed by the MR scanner. We will in particular use such interfaces to investigate the brain mechanisms involved in the novel kind of learning we have identified [2] (Fig.1). This requires the creation of novel MR compatible actuators as well as force/torque and position sensors. These devices must have very smooth control to interact with human motion. The control hardware will be based on existing PC technology and can be synchronized with commercially available scanners.

The present paper gives an overview of our developments. It describes the constraints, initial attempts and developed fMRI compatible robotic technology (section 2), its practical implementation and the experiments we performed to examine the MR compatibility and performances of the system (section 3).

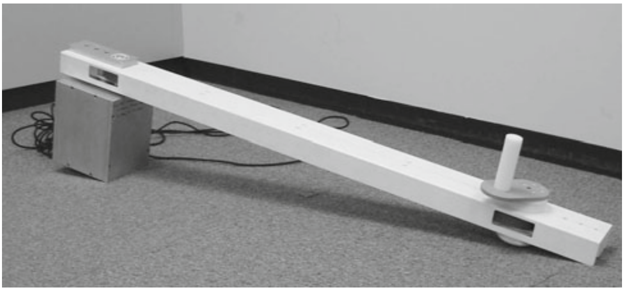


Fig. 2. Electromagnetic actuators, even when placed at a distance of several meters to the magnet coils as implemented in this first prototype of a haptic interface, disturb the imaging significantly.

2 An MRI Compatible Robot Technology

The main problem to create MR compatible robots is the electromagnetic compatibility. In a first attempt, we designed and tested an interface using a belt to transmit force and motion from a DC torque motor to a rotary output with a handle placed at the entry of the scanner bore (Fig.2). In this arrangement, the DC motor is placed within the shielded MR room but as far as possible from the bore of the electromagnet. The belt ensures a stiff transmission between the actuator and the output handle. The torque motor was placed inside an aluminum box with thick walls and soft iron plates fixed to the inside surfaces for additional electromagnetic shielding. With this design, the electromagnetic field produced by the motor should be far enough away from the scanner so as not to influence the homogeneity of the static magnetic field. At the same time, this arrangement would ensure that the torque motor (containing permanent magnets and ferromagnetic materials) be placed at a safe distance from the scanner to eliminate attraction by the electromagnet (missile effect).

First tests with standard MRI sequences did not show significant noise due to the actuation. However, when we migrated the interface to another scanner and used fMRI sequences the image was significantly disturbed as soon as a load was applied to the output. This raised a very important point: with such an active device, MR compatibility will also strongly depend on the MR equipment and on the applied scanning sequence. Further tests with a DC motor alone (with external load), in two different scanners using fMRI sequences, showed significant disturbance of the image when the DC motor was producing power. Based on this experience, we decided to systematically examine possibilities for actuation:

- As demonstrated with the first prototype, electromagnetic AC or DC actuators cannot be used within the MR room, as they produce strong magnetic fields and contain ferromagnetic materials.
- Electrostatic and ultrasonic actuators are magnetically neutral, but currently cannot produce a sufficiently good control for haptic interfaces. In particular, ultrasonic actuators are not suitable as they have high moment at rest and slow movements are difficult to achieve or would result in perceptible saccades.
- Hydraulic or pneumatic actuators do not have these drawbacks, but are difficult to realize and control, and require a heavy infrastructure.
- Long transmissions using cables or belts to place an actuator outside the MRI room are compliant and require a fixed design (pulleys). This is a disadvantage when transmitting high forces over long distances inside a small workspace, or when multiple actuators are required close to the scanner.

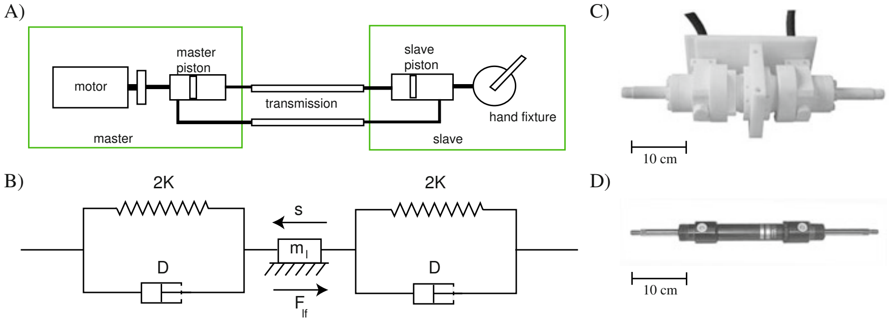


Fig. 3. MR compatible actuation concept with hydrostatic transmission. (A) System components - the transmission lines link the master and slave systems in a closed loop. (B) The hydrostatic transmission is modelled as a spring-damper system. Two such systems are used to represent the two lines connecting the master actuator with the slave. (C) MR compatible piston placed at the slave. (D) Equivalent commercial metallic piston used on the master side.

2.1 An MR Compatible Master-Slave System

The solution we have developed to comply with the MR environment consists of a conventional actuator placed outside the scanner room and a hydrostatic connection to transmit force and motion to a magnetically inert slave placed by or inside the MR scanner [12]. In contrast to a pneumatic transmission, the hydrostatic transmission should be sufficiently stiff to transmit force and motion with short delay, even over the relatively long distance of typically five to ten meters from the master actuator in the control room to the slave interface placed at the scanner bore.

Fig.3A sketches the proposed MR compatible master-slave system using a rotary direct drive motor. This motor drives the master piston via a transmission stage and a belt and pulley mechanism. The master cylinder is of a standard steel construction with a double ended piston-rod and hydraulic connections to both chambers (Fig.3D). It is connected to an MR compatible slave cylinder (Fig.3C) via metal-free hydraulic conducts. Motion range and cross section of the slave cylinder are identical to those of the master cylinder. In the case of a rotary actuator, the linear motion of the slave piston can be converted via a magnetically inert belt drive system.

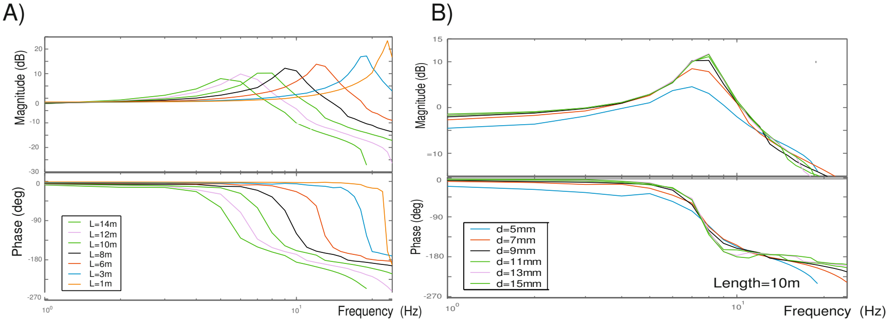


Fig. 4. Influence of critical parameters on the frequency response. (A) The Bode plot (over a frequency range of 1 to 24*Hz*) shows how the low-pass property of the hydrostatic transmission changes with increasing hose length. (B) The hose diameter does not influence the Bode plot significantly.

2.2 Analysis of Actuation with the Hydrostatic Transmission

To evaluate this concept and design suitable control strategies, we developed a nonlinear model of the hydrostatic transmission consisting of two hydraulic pipes connecting a master actuator to a passive slave in a cyclic arrangement [6] (Fig.3B). This model considers motor and fluid friction and inertia as well as viscoelasticity of the oil. Simulations were performed to determine critical parameters and evaluate their influence on the system behavior.

This nonlinear model can reproduce well the behavior of the hydrostatic transmission of [12] over a wide range of periodic movements [6]. To compare the model with the physical plant, most parameters were derived from manufacturer specifications or from our own measurements on the first interface. Only the friction parameters and a damping factor needed to be tuned in the simulation. This model was used to improve the design of the second generation hydrostatic transmission presented in section 3.3.

The simulation showed that the natural frequency (of about 7*Hz*) and cut-off frequency (of about 20*Hz*) increase with decreasing hose length (Fig.4A) and are almost independent of the hose diameter (Fig.4B). In general, the efficiency of the system in power transfer increases with decrease in hose length [6]. These dynamic features suggest that hydrostatic transmissions are suitable to create haptic interfaces interacting with human motion.

While transmissions of 10*m* provide sufficient bandwidth relative to the (typically 1 to 2*Hz*) human movements and can be controlled sufficiently well, the control will improve significantly if the transmission length is shorter. For example, our simulation suggests that by reducing the length from 10*m* to 6*m* the energy transmission approximately increases by a factor of about two. Therefore it is worth using a transmission with minimal length, and this should be considered at the time an MR facility is built. While a 10*m* long transmission can be used with most MRI facilities, adding a penetration panel in the wall closest to the scanner gantry may help reduce the transmission length to 4*m*.

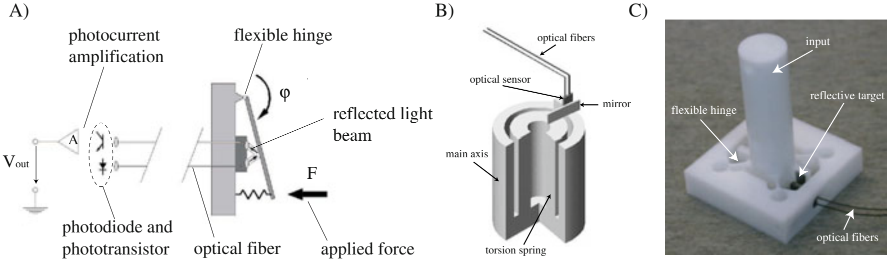


Fig. 5. MR compatible force sensing by measuring light intensity transmitted through optical fibers. (A) Schematic of a force sensor based on flexible hinges. (B) Schematic of a torque sensor based on a torsion spring. (C) Redesign of B) with reduced sensitivity to transverse forces.

2.3 Sensing

Force and Torque To allow accurate force measurement over a relatively long distance without any electronic circuitry, we use optical sensors with light transmitted trough optical fibers to detect deflection of a polymer probe by reflected light intensity measurement [12,3].

Fig.5A shows the principle of a force sensor based on a fiber optics sensor used to detect the displacement of a flexible hinge spring. Force/torque sensors for several degrees of freedom can be built based on this principle. Fig.5B shows the torque sensor realized for a one DOF fMRI compatible haptic interface based on the same principle. The sensing element consists of a polymer torsion beam guided in rotation by a ball bearing. To measure the rotation, a mirror is fixed on the beam. Fiber optics are used to move all the electronic components outside of the scanner room. The intensity of the reflected light beam is measured to derive the applied torque. The intensity variation is due to the beam spot displacement on the receptor optics. This kind of light measurement requires only two fibers per degree-of-freedom, and the fibers can be fixed on the same optical component, allowing a compact design. However, a ball bearing is needed to prevent sensitivity to transverse force [12]. Fig.5C shows a redesigned version of B) which is more compact and less sensitive to transverse forces, without the need of a ball bearing.

Displacement sensors Position sensors can also be realized using the proposed optical system. The reflective measurement eliminates alignment problems. Two optical sensors detect alternating black and white lines on a rotating mask. Alternatively, elevated tracks, deposed by photolithography, can be detected. To obtain useful position and direction information, the two sensors are mechanically shifted by one fourth the pitch of a line pair. This is achieved with a preloaded screw-sled system. As the disk rotates, the two sensors generate sinusoidal signals with a 90 degrees phase-shift, allowing detection of direction. The analog output signals are filtered, amplified and fed to a microprocessor for A/D conversion and, if required, for interpolation.

2.4 Redundant Safety

Interaction with human subjects requires severe and redundant safety measures. There are mainly four ways to prevent possible consequences of a malfunction, which will be implemented simultaneously:

i) mechanical travel limitations mounted on both the master and slave actuators;
ii) software emergency routines;
iii) electronic end-of-travel switches that stop the actuator before running into the mechanical limiters;
iv) a main power supply interruptor which can be actuated by the human subject in the scanner using a pneumatic emergency bellow *as well as* by the experimenter using a standard emergency button.

3 Implementation

3.1 Hardware

The main element of the proposed MR compatible robot concept is the magnetically inert hydraulic slave piston. Fig.3 shows the realized piston (C) with its metallic master counterpart (D). Both have the same characteristics of displacement range and active surface. The size difference is due to the different mechanical properties of the non-ferromagnetic, nonconducting materials. The master piston is driven by a direct drive torque motor. The slave actuator is connected to the motor rack by hydraulic conducts and the (Keystone FU-38) fiber optics. The rotary output is equipped with a torque sensor. The ten meters flexible metal-free hydraulic conducts are filled with a synthetic oil free of metal particles and pre-pressurized at 15 *bar* to assure a stiff transmission.31

The interface is programmed via MATLAB/Simulink on a Windows based host computer. This computer communicates with a target computer running an xPC Target (The MathWorks) realtime kernel. The target computer holds a National Instruments acquisition and an ADDI-DATA encoder card to generate the force profiles and run the control algorithm. A specially designed acquisition box acquires the sensor data, outputs the motor control signal and connects the control computer to the motor rack. This rack contains the direct drive (Inland RBHR series), the master piston and a commercial motor controller (Axor Super Drive). The acquisition box is also equipped with digital and analog input channels, thus allowing connection of additional measuring equipment such as EMG electrodes or a trigger signal from the MR scanner. This prototype was installed within an MR scanner at ATR International in Japan and proved to be compatible with fMRI.

3.2 Experimental Results

Magnetic compatibility is a sine qua non condition for our system. As fMRI protocols are generally more sensitive to inhomogeneities of the magnetic field than diagnostic MRI protocols, we tested our device in a (Marconi Magnex Eclipse 1.5 Tesla Power Drive 250) MR scanner with a GE EPI sequence. Compatibility tests with a phantom, i.e. a sphere filled with water, are generally considered as sufficient. However, in tests with our first prototype (Fig.2) using a motor placed inside the MR room at seven meters from the center of the magnet coil, the phantom test was fine, yet a test with human subjects revealed that the motor perturbed the images significantly.

Therefore, we completed the successful phantom tests by tests conducted with human subjects performing a real task. This test consisted of comparing the brain activity at rest with the activity when human subjects moved the wrist during repeated phases of 30 seconds with a 30 seconds break between two consecutive movement phases. There was no significant difference between movements performed with or without the interface [12], which indicates that the presence and actuation of the interface do not interfere with the functional imaging.

3.3 Hybrid MR Compatible System

In a next step we developed an interface to study the brain mechanisms of tactile sensing, which combines the passive hydrostatic transmission with active components we had tested and found to be MR compatible. This interface has two degrees of freedom: the linear displacement of a rotary output is actuated over a hydrostatic transmission. Displacement of the slave piston is measured with a potentiometer. The rotation of the output is actuated by an ultrasonic motor (USR60-E3N, Shinsei Corp., Japan).

The hydrostatic transmission was shortened to 8.5 m and reduced in size. After a series of material tests within an MR scanner, the slave piston was redesigned and fabricated from brass. This ensures better surface quality and thus reduces static friction. The master module was also redesigned - the mechanics and actuator were downsized and adapted to the novel transmission.

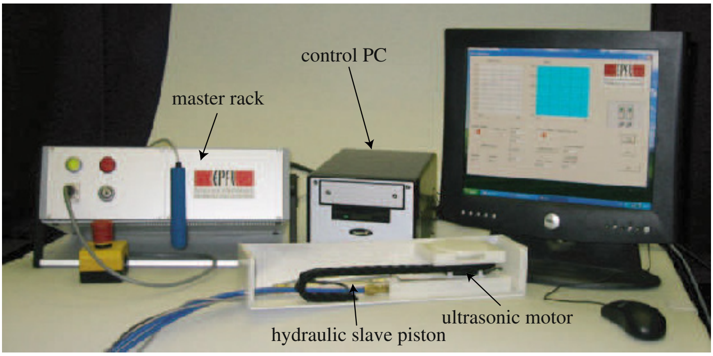


Fig. 6. Hybrid MR compatible system. The slave module, shown in the front, contains an active rotary degree-of-freedom (ultrasonic motor) linked to a passive translatory degree of freedom (brass hydraulic piston).

4 Conclusion and Research Perspective

In this paper we have presented our research on fMRI compatible haptic interfaces for neuroscience studies. To operate in conjunction with MRI/fMRI, it may be tempting to "simply" adapt conventional robot concepts and shield the perturbing devices. However, in tests with several MR scanners we found that electromagnetic actuators disturb the image significantly, even when placed at a distance of several meters from the magnet. Using such actuators inside the MR room may work for a given MR scanner and imaging sequence, however, with another scanner or sequence, the compatibility will need to be reexamined carefully, and important modifications of the device may be required.

Therefore, we developed a master-slave system with a hydrostatic transmission to provide force and motion from the control room to the interface near the scanner, using a non-conducting, non-ferromagnetic slave. A nonlinear model was developed and validated, which suggested performances suitable for haptic applications, and can be used to optimize the design of future interfaces. High-resolution force, torque and displacement sensors were developed using light transmission via fiber optics with no electronic components inside the MR room.

Using this technology we designed and realized two fMRI compatible haptic interfaces with force-feedback. While good performance was obtained with the first interface, which was developed using exclusively polymers, we were able to further improve performance using non-ferromagnetic materials such as aluminum and brass. These materials can support higher strain than polymers and components can thus be reduced in size. Additionally, higher surface quality can be achieved, which reduces static friction in the hydraulic cylinders.

To our knowledge, these interfaces are the first MRI/fMRI compatible devices able to deliver force and motion feedback during imaging. They will be used to investigate the brain mechanisms of motor learning and tactile perception in humans.

Preliminary experiments showed that they perform well within an MR scanner and allow (functional) image acquisition during motion without disturbance.

The developed interfaces allow precise and smooth movement control over a broad range of speed and force/torque. The modular approach enables the realization of robots for various applications requiring interaction with human motion such as haptic interfaces for neurophysiological studies and rehabilitation, as well as surgical robots for MRI-guided (minimally invasive) interventions.

We are currently developing multidimensional MR compatible force/torque and position sensors, and comparing the master-slave system with hydrostatic transmission to other potential actuation principles such as cable transmissions and ultrasonic actuators.

Acknowledgement We thank Roland Moser for his contribution to this project.

References

1. http://www.forcedimension.com, http://www.sensable.com.
2. E Burdet, R Osu, DW Franklin, TE Milner, and M Kawato. The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature*, (414):446–449, 2001.
3. D Chapuis, R Gassert, L Sache, E Burdet, and H Bleuler. Design of a simple MRI/fMRI compatible force/torque sensor. *IEEE International Conference on Robotics and Intelligent Systems (IROS)*, 2004.
4. K Chinzei, N Hata, F Jolesz, and R Kikinis. Surgical assist robot for the active navigation in the intraoperative MRI: Hardware design issues. In *Proc. IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS)*, pages 727–732, 2000.
5. Frankhauser et al. Robot for CT-guided stereotactic neurosurgery. In *Meeting of the World Society for Stereotactic and Functional Neurosurgery*, pages 93–98, 1994.
6. G Ganesh, R Gassert, H Bleuler, and E Burdet. Dynamics and control of an MRI compatible master-slave system with hydrostatic transmission. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1288–1294, 2004.
7. D Glauser et al. Conception of a robot dedicated to neurosurgical operations. In *Proc. International Conference on Advanced Robotics (ICAR)*, pages 899–904, 1991.
8. H Gomi and M Kawato. Equilibrium-point control hypothesis examined by measured arm-stiffness during multi-joint movement. *Science*, 272:117–120, 1996.
9. DJ Heeger and D Ress. What does the fMRI tell us about neuronal activity? *Nature Reviews*, 3:142–151, 2002.
10. N Hogan and HI Krebs. System and method for medical imaging utilizing a robotic device, and robotic device for use in medical imaging. *US patent #5794621*, 1995.
11. HI Krebs et al. Increasing productivity and quality of care: Robot-aided neurorehabilitation. *Journal of Rehabilitation Research and Development*, 37(6), 2000.
12. R Moser, R Gassert, E Burdet, L Sache, HR Woodtli, J Erni, W Maeder, and H Bleuler. An MR compatible robot technology. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 670–675, September 2003.
13. R Shadmehr and HH Holcomb. Neural correlates of motor memory consolidation. *Science*, 277(5327):821–825, 1997.
14. DD Stark and WG Bradley. *Magnetic Resonance Imaging*. St. Louis, 1988.

Nonholonomic Modeling of Needle Steering

Robert J. Webster III, Noah J. Cowan, Gregory Chirikjian, and Allison M. Okamura

Department of Mechanical Engineering
The Johns Hopkins University
{robert.webster,ncowan,aokamura,gregc}@jhu.edu
http://www.me.jhu.edu/

Abstract. As a flexible needle with a bevel tip is pushed through soft tissue, the asymmetry of the tip causes the needle to bend. We propose that, by using nonholonomic kinematics, control and path planning, an appropriately designed needle can be steered through tissue to reach a specified 3D target. Such steering capability will enhance targeting accuracy and markedly improve outcomes for percutaneous therapies, facilitate research on therapy effectiveness, and enable new minimally invasive techniques. In this paper, we consider a first step toward active needle steering: design and experimental validation of a nonholonomic model for steering flexible needles with bevel tips. The model generalizes the standard three degree-of-freedom (DOF) nonholonomic bicycle model to 6 DOF using Lie group theory. Model parameters were fit using experimental data, which were acquired via a robotic device designed for the specific purpose of inserting a flexible needle. The resulting parametric fit quantitatively validates the bevel tip needle steering model, enabling future research in flexible needle path planning, control and simulation.

1 Introduction

Needle insertion is perhaps the most widespread surgical technique in existence. It is a critical aspect of many medical diagnoses, treatments and scientific studies, including percutaneous procedures requiring therapy delivery to or sample removal from a specific location. However, errors in needle targeting can mitigate the effectiveness of diagnosis or therapy. Biopsies, for example, cannot completely rule out malignancy due to inaccuracy in positioning the needle tip. Also, radioactive seeds in prostate brachytherapy are often placed at locations substantially different than those preplanned for optimal dosage. In this work, we focus on modeling the needle bending that occurs due to tip asymmetry. Control and planning based on such a model can compensate for targeting disturbances due to needle bending, error in insertion angle, and tissue deformation.

A kinematic model of needle steering allows one to draw on the many path planning techniques developed by the robotics community. Judiciously chosen paths will allow needles to reach small targets in locations that are inaccessible via straight trajectories. This will expand range of procedures that can be accomplished minimally invasively, requiring fewer open surgeries. Needle steering also has the potential to expand new arenas of clinical research, allowing accurately and precisely targeted injection of new drugs, genetic agents, ablative and cryoablative treatments, etc. Accurate targeting is essential in such clinical research, because it must be verified that

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 35–44, 2006.

results and outcomes are due to differences in the therapy itself, and not a result of the targeting of the therapy. Needle bending models will also aid in the development of realistic needle insertion simulators for therapy planning and training.

While the needle insertion modeling community has generally regarded the effect of bevel tip asymmetry as an unwanted source of error, it is interesting to note that in clinical practice, some surgeons do make use of the bevel tip [11]. They use it both for steering the needle and aiming injected fluids (the fluid will flow outward at an angle from the bevel). Surgeons accomplish this from experience, making it difficult to teach and limiting accuracy to that of human hand/eye coordination.

Early work in needle modeling and simulation involved recording the forces applied to a needle during insertion and playing back "haptic recordings" or simple force vs. position models in a force-feedback virtual environment [2,7,8]. More recently, researchers have used reality-based needle/tissue interaction modeling to generate models used for planning and simulation [1,4,14]. The effects of needle bending have been explored by several groups. O'Leary, *et al.* [13] demonstrated experimentally that needle bending forces are significantly affected by the presence of a bevel tip. Others have generated needle bending using different strategies such as incorporating a pre-bent stylus inside a straight canula [6], or a telescoping double canula where the internal canula is pre-bent [3]. Kataoka, *et al.* [9] attempted to create a model for needle deflection, but did not account for the bevel tip and admit that the bevel is likely the main source of deflection.

There is only one previous study that has analyzed needle paths from a kinematic viewpoint. DiMaio and Salcudean [5] formulate a needle Jacobian that describes tip motion due to needle base motion and a tissue finite element model. However, their work does not explore the effect of tip asymmetry. Their needle is stiff relative to the tissue, and steering is accomplished by pulling on and angling the needle shaft outside the body to cause the tissue to move. Our approach contrasts theirs in that we consider a system where the needle is flexible relative to the tissue, and does not displace a large amount of tissue in order to steer itself. Ultimately, we expect that a combination of these methods will be most accurate in controlling and describing needle insertions into actual soft tissue.

2 Modeling of Needle Steering

Consider a bevel tip needle driven with two velocity inputs, insertion speed and rotation speed, actuated from the base of the needle. As the needle is inserted into tissue, the tissue imposes a reaction force on the bevel that deflects the needle tip, causing it to follow an arc. Neglecting torsional compliance of the needle, the rotational input at the base causes the needle to turn about its shaft, reorienting the bevel. We model both effects – insertion and rotation – as inputs to a quasi-static nonholonomic system.

For a fixed needle shaft rotation, the model is similar to that of a bicycle with a fixed front wheel angle, ϕ, and fixed distance, ℓ_1, between the front and back wheel, as depicted in Figure 1. A third model parameter, ℓ_2, determines the location along

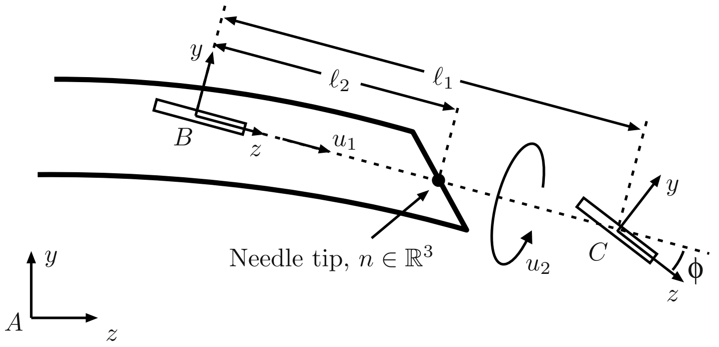


Fig. 1. Configuration of a bevel tipped needle during steering showing the front and back "wheels" at frames B and C of a superimposed bicycle-like nonholonomic model. In this particular configuration, the x-axes for all three frames are pointing into the page.

the bicycle attached to the needle tip, n. The purpose of this study is to determine if the model quantitatively captures the needle steering kinematics. To do so, we fit the model parameters (ϕ, ℓ_1, ℓ_2) experimentally as described in Section 3. We suspect that the model parameters depend on many factors such as tissue stiffness, needle stiffness and bevel angle, but we leave this hypothesis for future investigation.

2.1 Notation and Definitions

Ultimately, we seek to use the two control inputs, insertion and rotation, to drive a needle to a desired position and orientation in six degrees of freedom (DOF). Since generalized coordinates (such as $(x, y, z, \text{roll}, \text{pitch}, \text{yaw})$), have singularities, we resort to a coordinate-free representation of the kinematics. Fortunately the kinematic needle equations are quite simple in the coordinate-free representation, but the convenience and generality comes at the added expense of the formalism and notation presented in this section. We follow the conventions in [12].

Consider the three reference frames depicted in Figure 1: a stationary world frame, A, and two "body" frames, B and C, attached to the needle tip. Using the homogeneous matrix representation, let

$$g_{ab} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3) \quad \text{where} \quad R_{ab} \in \mathrm{SO}(3),\ p_{ab} \in \mathbb{R}^3$$

denote the rigid transformation between A and B. Likewise, let $g_{bc} = (R_{bc}, p_{bc}) \in \mathrm{SE}(3)$ denote the transformation between B and C.

The isomorphism $\mathbb{R}^3 \simeq \mathfrak{so}(3)$ is defined by

$$\widehat{\ }: \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \mapsto \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathfrak{so}(3), \quad {}^\vee: \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \mapsto \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix},$$

where $\mathfrak{so}(3)$ is the Lie algebra of SO(3). It will be convenient to "overload" the definitions of $\hat{}$ and $^\vee$ for $\mathfrak{se}(3)$, the Lie algebra of SE(3). In other words, if $(v, \omega) \in \mathbb{R}^6$, then

$$\hat{}\,:\begin{bmatrix} v \\ \omega \end{bmatrix} \mapsto \begin{bmatrix} \hat{\omega} & v \\ 0^T & 0 \end{bmatrix} \in \mathfrak{se}(3), \quad {}^\vee : \begin{bmatrix} \hat{\omega} & v \\ 0^T & 0 \end{bmatrix} \mapsto \begin{bmatrix} v \\ \omega \end{bmatrix}.$$

Given two frames, X and Y, related by the rigid transformation $g_{xy} \in \mathrm{SE}(3)$, the body-frame velocity between them is given by

$$V_{xy}^b = \begin{bmatrix} v_{xy}^b \\ \omega_{xy}^b \end{bmatrix} = (g_{xy}^{-1} \dot{g}_{xy})^\vee, \quad \text{where} \quad \begin{aligned} v_{xy}^b &= R_{xy}^T \dot{p}_{xy}, \\ \omega_{xy}^b &= (R_{xy}^T \dot{R}_{xy})^\vee. \end{aligned}$$

Given three frames A, B and C moving relative to each other, their body velocities are related by $V_{ac}^b = \mathrm{Ad}_{g_{bc}^{-1}} V_{ab}^b + V_{bc}^b$, where

$$\mathrm{Ad}_g = \begin{bmatrix} R & \widehat{p} R \\ 0 & R \end{bmatrix}$$

is the Adjoint operator for a rigid transformation $g = (R, p) \in \mathrm{SE}(3)$.

The unit vectors $e_1, e_2, e_3 \in \mathbb{R}^3$ are the standard basis.

2.2 Nonholonomic Constraints and Control Inputs

Frames B and C are rigidly connected with parallel x-axes, such that the origin of C is a distance ℓ_1 along the z-axis of B. The y-z plane of C is rotated by angle ϕ about the x-axis, as shown. Thus $R_{bc} = e^{\widehat{e_1}\phi}$ and $p_{bc} = \ell_1 e_3$, where ϕ and ℓ_1 are constants to be determined experimentally as described in Section 3.

There are four Pfaffian constraints, because the velocity of the origin of frame B cannot have a projection along the B frame x-axis or y-axis, and the velocity of the origin of frame C cannot have a projection along the C frame x-axis or y-axis. In other words:

$$e_1^T v_{ab}^b = e_2^T v_{ab}^b = e_1^T v_{ac}^b = e_2^T v_{ac}^b = 0.$$

Since frames B and C are fixed with respect to each other, $V_{bc}^b = 0$. Thus $V_{ac}^b = \mathrm{Ad}_{g_{bc}^{-1}} V_{ab}^b + V_{bc}^b = \mathrm{Ad}_{g_{bc}^{-1}} V_{ab}^b$, and we have

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & \ell_1 & 0 \\ 0 & \cos\phi & \sin\phi & -\ell_1 \cos\phi & 0 & 0 \end{bmatrix}}_{A} V_{ab}^b = 0\,.$$

Assuming $\ell_1 \neq 0$ and $\phi \in (0, \pi/2)$, a basis V_1, V_2 for the right nullspace of A defines the two allowable directions:

$$V_1 = \begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} e_3 \\ \frac{\tan\phi}{\ell_1} e_1 \end{bmatrix}, \; V_2 = \begin{bmatrix} v_2 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} 0_{3\times 1} \\ e_3 \end{bmatrix}.$$

The vector V_1 corresponds to pure needle insertion, while V_2 corresponds to pure needle shaft rotation.

Since we assume the needle shaft is held in place by the surrounding tissue, the effect of the shaft is to replicate needle base control inputs at the tip. Let $u = (u_1, u_2)$ denote the control inputs, where u_1 is the insertion speed, and u_2 is the shaft rotation speed. This leads to the following kinematic model:

$$V_{ab}^b = u_1 V_1 + u_2 V_2, \quad \text{or, equivalently} \quad \dot{g}_{ab}(t) = g_{ab}(t)(u_1 \widehat{V_1} + u_2 \widehat{V_2}). \tag{1}$$

and

$$n(t) = R_{ab}(t)\ell_2 e_3 + p_{ab}(t). \tag{2}$$

Note that the constraint matrix A is independent of g_{ab}, and thus the control vector fields are left-invariant. The system (1) is nonholonomic, since the distribution $\Delta = \text{span}\,\{V_1, V_2\}$ is not involutive. This can be seen by taking the first Lie bracket

$$V_3 = [V_1, V_2] = (\widehat{V_1}\widehat{V_2} - \widehat{V_2}\widehat{V_1})^\vee = \begin{bmatrix} 0_{3\times 1} \\ -\frac{\tan\phi}{\ell_1} e_2 \end{bmatrix},$$

which is linearly independent of V_1 and V_2 (and thus Δ is not involutive). Successive Lie brackets reveal that the system is of nonholomy degree 4, with a relative growth vector of $(2, 1, 2, 1)$ (see [12], Chapter 7). This suggests that this system is controllable, but we leave needle control and steering to future work.

2.3 Discrete Model

A discrete implementation of the kinematic model (1-2) enables simulation and visualization. Advancing the homogeneous transformation, g_{ab}, along V_{ab}^b for T seconds for each time step, $k = 0, 1, 2, \ldots$, yields the discrete-time model

$$\begin{aligned} g_{ab}(k+1) &= g_{ab}(k) e^{\left(u_1(k)\widehat{V_1} + u_2(k)\widehat{V_2}\right)T} \\ n(k) &= R_{ab}(k)\ell_2 e_3 + p_{ab}(k). \end{aligned} \tag{3}$$

The control inputs $u_1(k)$ and $u_2(k)$ now denote the insertion distance and change in rotation angle, respectively, at step k.

3 Experimental Validation

3.1 Materials

A needle driving mechanism (Figure 2) was designed to control both insertion (u_1) and rotation (u_2) speeds. The insertion subassembly drives the needle by grasping it on the barrel using two opposing rubber wheels actuated by a worm gear attached to a motor. Rotation of the needle about its axis is achieved by rotating the insertion subassembly as a unit. Since the wheels grasp the needle tightly by the barrel,

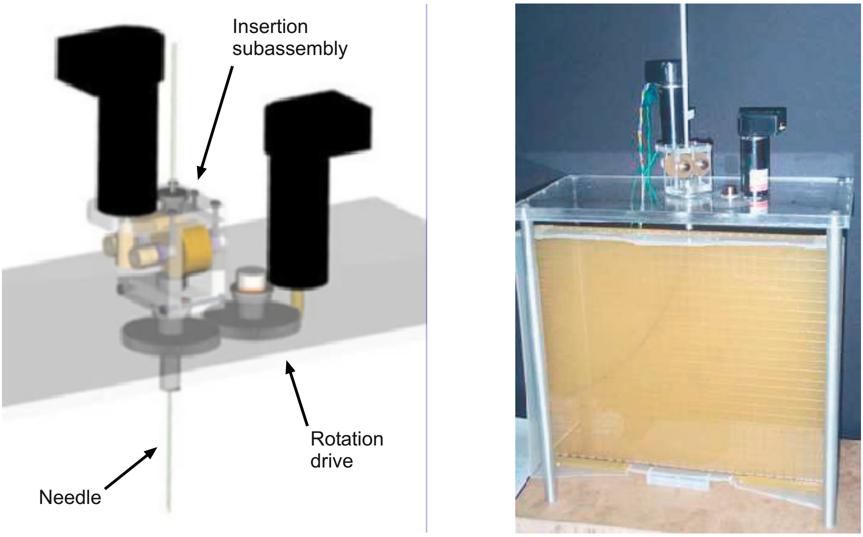


Fig. 2. A needle driving mechanism for steering of flexible needles: CAD model (left) and experimental assembly (right).

rotating the subassembly causes the needle to rotate as well. A slotted needle guide (shown only in the photograph) further fixes the orientation of base of the needle, and thus the bevel direction, relative to the drive wheels. This prevents unwanted needle rotation as the drive wheels turn. Buckling is prevented by passing the needle through a 1.5 mm hole drilled through the aluminum rod that supports the insertion subassembly. This rod extends to the surface of the phantom tissue into which the needle is inserted.

The needle used in the experiments was a 0.7 mm diameter solid Nitinol cylinder (simulating a 22 gauge needle) with a smooth surface finish and a hand-machined bevel tip of 45^o. The phantom tissue material used in this experiment was Simulated Muscle Ballistic Test Media from Corbin, Inc. This material was specifically chosen to be stiff (4.9 N/mm by a blunt indentation test) to fit our modeling assumption that the tissue is stiff enough relative to the needle that macroscopic displacement of the tissue by the needle will not occur. That is to say, the needle shaft will follow the trajectory of the tip position. The rubber-like Ballistic Test Material was cast into a sheet approximately 15 mm thick, and the needle was introduced vertically.

To collect coordinates describing the needle path in each insertion, a physical grid was overlaid on the phantom tissue. The 1 cm square grid was laser etched into a clear polycarbonate sheet so that digital photographs could be taken of the needle path through the grid (Figure 2). Coordinates of the needle path were recorded from the digital photographs for each insertion using the physical grid. One data point was recorded for each discrete z value (in 1 cm increments). The y values were recorded to within ±1 mm. The nonholonomic model was fit to the data as described in the following section.

3.2 Experimental Procedure

The needle described previously was inserted multiple times into a single phantom tissue sample for all experiments. Care was taken to insert the needle at a different location each time so that the holes cut by previous experiments would not affect the needle path. The rotation of the bevel was maintained by putting a human in the loop (for the rotation degree of freedom only) who watched the needle and minimized out of plane motion by rotating the bevel slightly as necessary. Sources of error in these experiments include initial insertion angle from vertical, human servoed (approximately constant) spin angle, slippage of drive wheels relative to the shaft (not visually perceptible), small deformations of the phantom tissue, and identification of points on the needle path in digital images.

Two sets of input parameters were used in the experimental insertions. In one, u_2 was set to zero, and the needle was inserted at a constant u_1 to a depth of 235 mm. This created a "single bend" insertion profile. In the other, u_2 was set to zero for the first 1/3 of the total insertion depth (83.3 mm). u_1 was then set to zero and the needle was rotated 180 degrees. Finally, with u_2 again fixed at zero, the needle was inserted the remaining 2/3 of the insertion depth at constant u_1, until the needle reached a total insertion depth of 250 mm. This created an S-shaped or "double bend" insertion profile. A total of 13 insertions were performed, composed of eight single bend insertions and five double bend insertions.

When the insertion speed u_1 is constant and needle rotation does not change ($u_2 = 0$), the needle tip follows a circular arc that is a function of the parameters $w = (\phi, \ell_1, \ell_2)$:

$$y = f_w(z) = \sqrt{r_w^2 - (z - a_w)^2} + b_w$$

where the arc center (a, b) and radius r are straightforward functions of the parameters, w. The model can then be fit to the experimental data by minimizing the mean squared error (MSE) cost function:

$$\mathrm{MSE}(w) = \frac{1}{n}\sum_{i=1}^{n}\big(y_i - f_w(z_i)\big)^2. \tag{4}$$

Matlab's `fminsearch` command minimizes a cost function via unconstrained nonlinear optimization according to the Nelder-Mead Simplex Method [10]. To fit all 13 trials simultaneously, the above cost function (4) was modified to include two unique nuisance parameters for each individual trial. These parameters were x_o (the x entry point of the needle) and θ (the initial angle of needle in the y-z plane). The θ parameter was included because it was observed that while all insertions had similar basic shape and curvature properties, they differed by a slight angle in rotation in a manner indicative of a small amount of error in initial entry angle. This error was probably caused by the needle tip deforming the surface of the rubber slightly before puncturing it, and deflecting itself a small amount in the process.

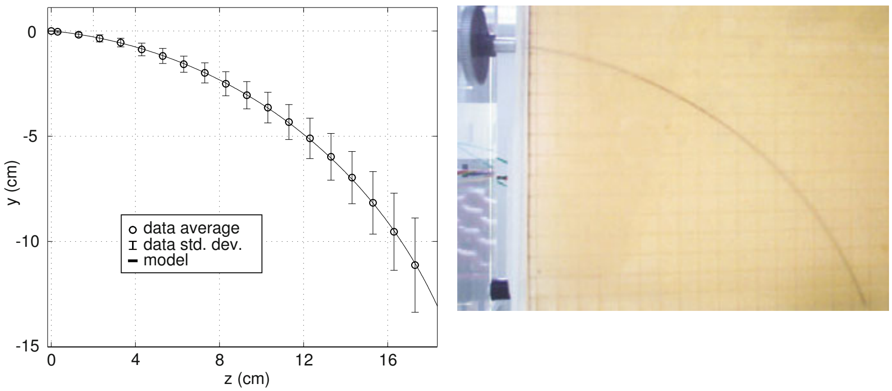


Fig. 3. (Left) The nonholonomic model fit to the average single bend experimental needle path. (Right) During the experiment, the needle was inserted 23.5 cm, without spin.

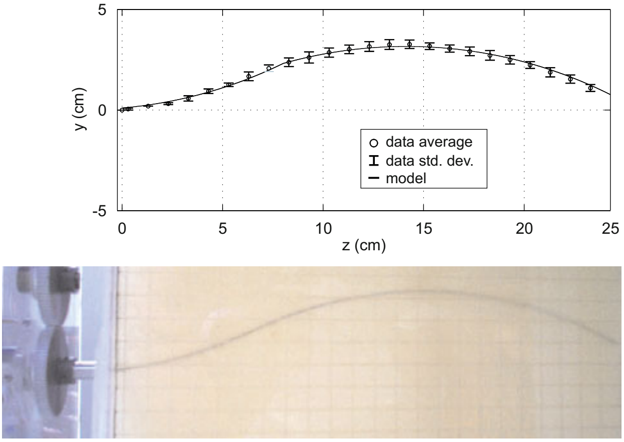


Fig. 4. (Top) The nonholonomic model fit to the average double bend experimental needle path. (Bottom) During the experiment, the needle was inserted 8.3 cm, spun 180^o, then inserted another 16.7 cm.

3.3 Results

The experimentally fit parameters were $\phi = 9.96^o$, $\ell_1 = 3.96$ cm and $\ell_2 = 2.03$ cm. Figures 3 and 4 show plots of the single bend and double bend fitted models along with mean data values and standard deviation bars for each data point. Note that the error bars are for raw data and do not include fitted nuisance parameters. This explains why they grow as the needle progresses into the tissue, since the initial angle offset, θ, has a greater effect at greater depth. To create the plots, nuisance parameters for the mean insertion data were generated using the same method originally used to fit the model to the data, but with (ϕ, ℓ_1, ℓ_2) held constant.

As shown in Figures 3 and 4, the model qualitatively fits the data well. This can also be seen quantitatively in that the MSE over all 13 insertions was 9.8 mm^2. This corresponds to a y error of ± 3.1 mm per data point, which is quite low. Nevertheless, the error is somewhat above the measurement error of ± 1 mm, so clearly there is some inherent variability in the data not captured by our kinematic model.

While these results reveal excellent agreement between the model and experimental data, several specific improvements to the experimental and model fitting methods are possible, and are the subject of future investigation. First, we are not able to accurately control the spin of the needle tip (u_2) by adjusting the rotation of the needle base because of the finite torsional stiffness of the needle shaft. We hypothesize that this issue can be overcome by controlling torque. That is, if the needle tip and base are at the same angle, the torque in the shaft will be zero. To enable this, we plan to add a torque sensor to the drive mechanism and implement a controller to obtain the desired shaft rotation. Second, the method for measuring the position of the needle tip as it travels through the tissue can be improved. Currently, we measure points along the needle shaft after the needle has reached its final position. However, this may not precisely reflect the actual path of the needle tip due to shaft stiffness and phantom tissue deformation. Third, it will be useful to more fully understand and assess the quality of the local minima of the cost function, since several parameter combinations appear to fit the data with nearly equal success. This can be an advantage in terms of fast convergence, or a drawback if the optimization is not initialized near a “good” solution.

4 Conclusion and Future Work

A first step in steering a needle to a desired location inside the human body is a kinematic analysis of the needle path. This paper introduces a 6 DOF nonholonomic model based on steering due to bevel tip asymmetry. Using a robotic mechanism designed for flexible needle insertion, we demonstrated that our model accurately predicts the path of a compliant needle through relatively stiff phantom tissue.

This work facilitates a broader study to improve the accuracy of needle targeting for clinical and research applications. Future research activities include: (1) determining the relationship of the bevel angle to the steering angle ϕ, (2) creating a noise model that captures the inherent variability of needle insertion, (3) integrating this needle steering model into simulations that include tissue deformation [1,4,5], (4) path planning for steering needles around obstacles (e.g., bones, delicate structures, etc.) in order to acquire targets not previously accessible, (5) selection of optimal insertion points, and (6) development of a robotic system for needle steering that uses feedback from medical imaging to enhance accuracy in the presence of unmodeled tissue deformation and inhomogeneity.

Acknowledgments

This work was supported in part by the National Institutes of Health under grant R21 EB003452 and a National Defense Science and Engineering Graduate Fellowship.

The authors thank Ken Goldberg and Ron Alterovitz for their ideas contributing to this work.

References

1. R. Alterovitz, J. Pouliot, R. Taschereau, I-C Hsu, and K. Goldberg. Sensorless planning for medical needle insertion procedures. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3337–3343, 2003.
2. P. N. Brett, A. J. Harrison, T. A. Thomas, and A. Carr. Simulation of resistance forces acting on surgical needles. *Proceedings of the Institution of Mechanical Engineers*, 211 (H4):335–347, 1997.
3. W. Daum. A deflectable needle assembly, 2003. Patent 5,572,593.
4. S. P. DiMaio and S. E. Salcudean. Needle insertion modeling and simulation. *IEEE Transactions on Robotics and Automation*, 19(5):864–875, 2003.
5. S. P. DiMaio and S. E. Salcudean. Needle steering and model-based trajectory planning. *Medical Image Computing and Computer-Assisted Intervention*, pages 33–40, 2003.
6. R. Ebrahimi, S. Okzawa, R. Rohling, and S. E. Salcudean. Hand-held steerable needle device. *Medical Image Computing and Computer-Assisted Intervention*, pages 223–230, 2003.
7. P. Gorman, T. Krummel, R. Webster, M. Smith, and D. Hutchens. A prototype haptic lumbar puncture simulator. *Proceedings of Medicine Meets Virtual Reality*, pages 106–109, 2000.
8. L. Hiemenz, J. S. McDonald, D. Stredney, and D. Sessanna. A physiologically valid simulator for training residents to perform an epidural block. *Proceedings of the IEEE Biomedical Engineering Conference*, pages 170–173, 1996.
9. H. Kataoka, T. Washio, M. Audette, and K. Mizuhara. A model for relations between needle deflection, force, and thickness on penetration. *Medical Image Computing and Computer-Assisted Intervention*, pages 966–974, 2001.
10. J. C. Lagarias, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.
11. K. Murphy. Department of Radiology, The Johns Hopkins Medical Institutions. Personal Communication, 2003.
12. R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Ann Arbor MI, 1994.
13. M. D. O'Leary, C. Simone, T. Washio, K. Yoshinaka, and A. M. Okamura. Robotic needle insertion: Effects of friction and needle geometry. *IEEE International Conference on Robotics and Automation*, 2003. In press.
14. C. Simone and A. M. Okamura. Haptic modeling of needle insertion for robot-assisted percutaneous therapy. *IEEE International Conference on Robotics and Automation*, pages 2085–2091, 2002.

Microphone Array for 2D Sound Localization and Capture

Satoshi Kagami[1,2], Hiroshi Mizoguchi[1,3], Yuuki Tamai[1,3], and Takeo Kanade[1,2,4]

[1] Digital Human Research Center, National Institute of Advanced Science and Technology
{s.kagami,t.kanade}@aist.go.jp
[2] CREST Program, JST (Japan Science and Technology Corporation)
[3] Dept. of Mechanical Engineering, Tokyo University of Science
hm@rs.noda.tus.ac.jp, j7599107@ed.noda.tus.ac.jp
[4] Robotics Institute, Carnegie Mellon University.

Abstract. This paper describes two circular microphone arrays and a square microphone array which can be used for sound localization and sound capture. Sound capture by microphone array is achieved by Sum and Delay Beam Former (SDBF). Simulation of sound pressure distribution of 32 & 128ch circular microphone array and 128ch square microphone array are shown. According to simulation results, dedicated PCI 128-channel simultaneous input board and Firewire (IEEE1394) 32-channel board are developed with maximum sampling rate of 44.1kHz and 11.025kHz sample respectively.

Then a 32ch circular microphone array and a 128ch square microphone array have been developed. The 32ch circular microphone array can capture sound from an arbitrary direction. The 128ch square microphone array can capture sound from a specific point. Both systems are evaluated by using frequency components of the sound. The circular type system will be used on a mobile robot including humanoid robot, and square type will be extend towards room coverage type application.

1 Introduction

Sound source localization and capture are one of fundamental function for a robot that behaves in a human world. Not only voice capture by omitting background noise for a voice recognition, but also to notice human position or to catch a cue from human being will be important for a human-robot interaction.

We are performing research which centers around human-machine interaction[1]. Microphone arrays are often used for separating sound signals which are from a specific position and sound source localization[2–4]. The method of sound signal separation by a microphone array is already used for a voice recognition-specific microphone.

In this paper, authors developed a 32ch circular microphone array and a 128ch square microphone array for a robotics application. A circular microphone array can detect and capture arbitrary sound directions, and a square type can localize and capture arbitrary sound positions. Capturing sound by microphone array is achieved by Sum and Delay Beam Former. In order to build a microphone array, it is necessary

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 45–54, 2006.

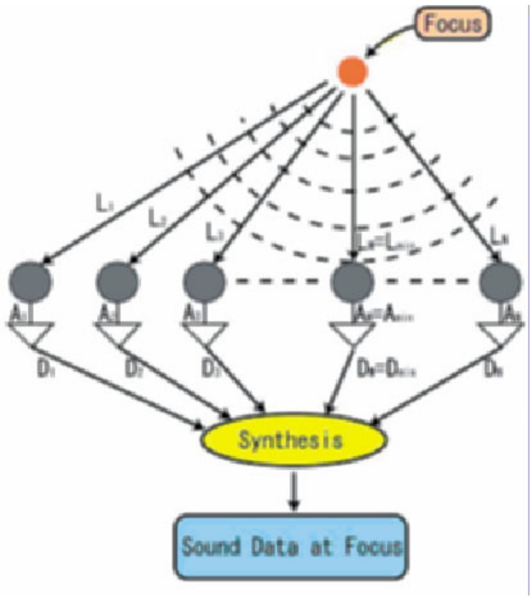


Fig. 1. Sound Capture by Microphone Array

to sample every channels simultaneously at high speeds. Dedicated PCI 128-channel simultaneous input board and Firewire (IEEE1394) 32-channel board are developed with maximum sampling rate of 44.1kHz and 11.025kHz sample respectively.

2 Sound Capture Algorithm

Assume that focusing point C is an ideal omnidirectional sound source (Fig.1 and there are N microphones. Let $L_i(i < N)$ be distance from C and L_{min} be minimum distance among all L_i. Amplitude A_i and phase shift time D_i for each microphone related to the minimum distance microphone are calculated as follows.

$$A_i = \frac{L_{min}}{L_i} \tag{1}$$

$$D_i = \frac{L_{min} - L_i}{V_s} \tag{2}$$

Here V_s is a sound speed. Amplitude A_i was set as each microphone receives the same loudness from the point C. Consider sin curve $\sin(2\pi Ft)$ with frequency F is captured at i^{th} microphone, then following wave was arrived.

$$N_i(t) = A_i \sin(2\pi F(t + D_i)) \tag{3}$$

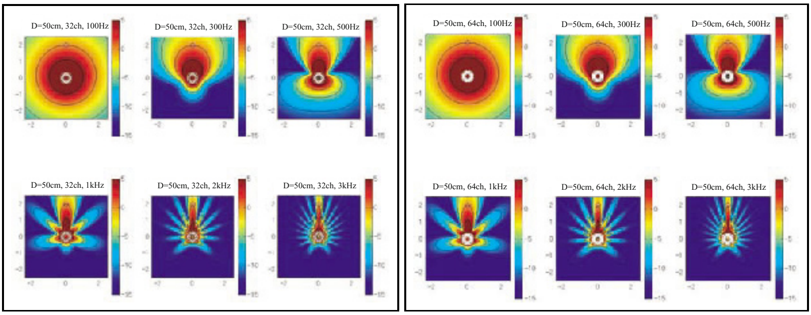


Fig. 2. Simulation result for 32ch and 64ch circle microphone (ϕ 50[cm])

Where decreasing ratio with the minimum distance microphone B_i and time shift E_i at point C are described as follows.

$$B_i = \frac{L_i}{L_{min}} \tag{4}$$

$$E_i = \frac{L_i}{V} \tag{5}$$

Therefore, from the point C, following wave is obtained.

$$O_i(t) = \sin(2\pi F(t + \frac{L_{min}}{V})) \tag{6}$$

Every microphone input wave is synchronized and accumulated at the point C, so that total wave is as follows:

$$Q(t) = \sum_{i=1}^{N} O_i(t) = \sum_{i=1}^{N} \sin(2\pi F(t + \frac{L_{min}}{V})) \tag{7}$$

3 Simulation

In this chapter, simulations by using Sum and Delay Beam Former (SDBF) of 32 & 128ch circular microphone array, and 128ch square microphone array are described.

3.1 Sound Pressure Equations

Simulation can be done by using the following parameters: microphone arrangement, number of microphone, frequency F, location $P(x, y)$. Let R_{iP} is the distance from i^{th} microphone and point P. Then total wave at point P can be described at follows:

$$Q_P(t) = \sum_{i=1}^{N} \frac{L_i}{R_{iP}} \sin(\omega t + \frac{L_{min} + R_{iP} - L_i}{V}) \tag{8}$$

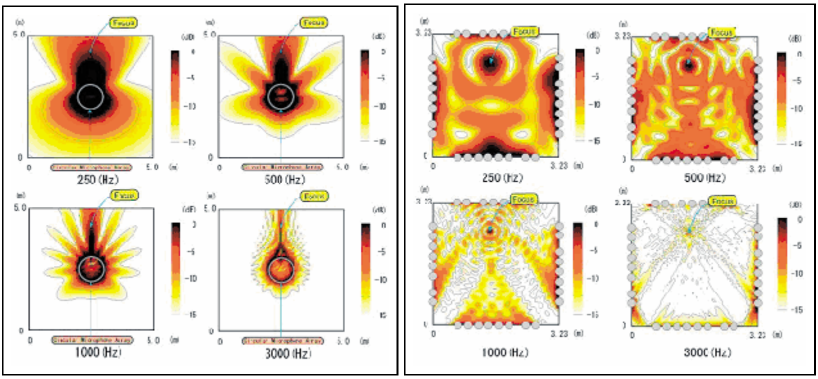


Fig. 3. Simulation result for 128ch circle (ϕ 100[cm]) and square microphone (320[cm] square)

where $\omega = 2\pi F$.
Set $\Theta_{iP} = \frac{L_{min}+R_{ic}-L_i}{v}$ and $F_{iP} = \frac{L_i}{R_{iP}}$. Then

$$
\begin{aligned}
Q_P(t) &= \sum_{i=1}^{N} F_{iP}\sin(\omega t + \Theta_{iP}) \\
&= A_p \sin(\omega t + \alpha)
\end{aligned}
\tag{9}
$$

where,

$$
\begin{aligned}
A_p &= \sqrt{(\sum_{i=1}^{N} F_{iP}\cos(\Theta_{iP}))^2 + (\sum_{i=1}^{N} F_{iP}\sin(\Theta_{iP}))^2} \\
\alpha &= \arctan \frac{\sum_{i=1}^{N} F_{iP}\sin(\Theta_{iP})}{\sum_{i=1}^{N} F_{iP}\cos(\Theta_{iP})}
\end{aligned}
\tag{10}
$$

A_p is a total amplitude of the point w$P(x, y)$, and α is a phase shift.

Sound pressure level S_{PL} (dB) is calculated by using sound pressure P_1 (Pa:N/m^2) and base sound pressure $(P_0)(2 \times 10^{-5} N/m^2)$ as follows:

$$
S_{PL} = 20\log_{10}\frac{P_1}{P_0}
\tag{11}
$$

In this simulation, sound pressure level of the point $P(x, y)$ is

$$
S_{PL_P} = 20\log_{10}\frac{A_p}{Q}
\tag{12}
$$

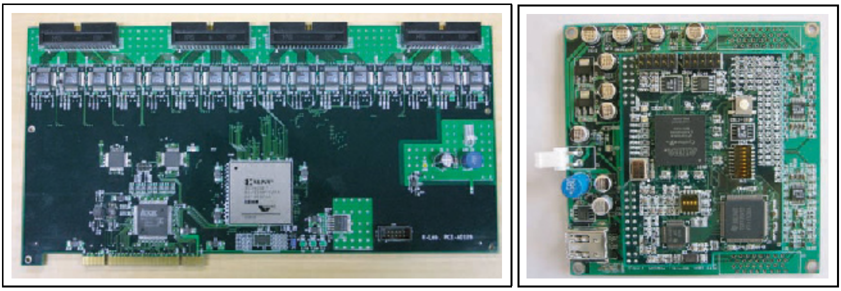


Fig. 4. PCI 128ch AD Board and IEEE1394 32ch Mic board

3.2 Simulation Results

In simulation, sound pressure distribution map is created. Frequency ingredient of human voice is said to be 100-3000Hz, so, Sin wave of those bands are used in simulation.

Fig.2 shows simulation results of 32ch and 64ch circular microphone array with diameter 50[cm]. Fig.3(left) shows simulation results of 128ch microphone array with diameter 100[cm]. Fig.3(right) shows simulation results of 128 ch square microphone array. The length of one side is 320[cm].

In 32ch circular microphone array, above 1[kHz] is focused. At low frequency, high sound pressure area spreads out in wide range. So if sound source only generates low frequency sound, sound capture and sound source localization by SDBF will have limited performance.

In 128ch circular microphone array, only 250[Hz] is not focused. Therefore, sound capture and sound source localization will work above 500[Hz] sound source.

In 128ch square microphone array, all frequency range is focused.

Table 1. PCI 128ch AD board and IEEE1394 32ch Microphone board specs.

	128ch AD	32ch Microphone
Spec	32bit PCI Spec. Rev.2.3	IEEE1394
IF	ADTEC System Science APIC22	TI TSB12LV32+TSB41AB1
FPGA	XILINX XCV600E	ALTERA EP1C6F256
AD	AD7865 x 32 (128ch)	AD7490 x 4 (32ch)
Res.	14bits	12bits
Ratio	x1	x800-5000
Input	-5V to 5V	ECM
Transfer(Typ.)	DMA 5[us] (Typ.)	Isochronous 15[us]

4 System Design and Implementation

A system that all components are driven by software, can have wide flexibility for developing the application. Especially focusing algorithm and digital filtering technique are important. Therefore, in this paper, real-time OS based online sound focus & capture system was developed.

Fig.6 shows 32ch circular and 128ch square microphone array. The 32ch circular system has 50[cm] diameter and Sony “ECM-T145” is adopted as a microphone. The 128ch square system has 320[cm] at one edge length.

4.1 128ch High Speed Simultaneous AD board

In order to develop online sound spot focus & capture system, number of I/O and cycle of the system are the problem. We developed high speed 128ch simultaneous AD board(Fig.4(left)). This board has DMA function that transfer data at once (128ch x 16bits = 256B). Since 32bit/33MHz PCI bus has 133MB/s transfer ratio, 256Bytes x 44Khz = 11.2MB/s can be achieved by normal PC system. The board has 14bits resolution, 128ch simultaneous AD with 44.1[KHz] sampling rate. Table 1(left) shows specifications of the board.

4.2 Online Sound Signal Handling by ART-Linux

44.1[KHz] sampling rate requires 22.7[us] resolution, and it is hard requirement for usual PC based real-time OS. We compared four linux based realtime OS, RT-Linux, RTAI, Timesys Linux and ART-Linux. By examining those real-time OS, ART-Linux[5] has the best performance since it is basically polling based system. Fig.5 shows a ART-Linux system performace on 2.0GHz PentiumIV system. Three sampling cycle (10, 50, 500[us]) with/without load the system. Even 10[us] sampling rate, experiment shows that 2[us] worst case jitter was measured. Therefore, 22.7[us] (44Khz) sampling rate can be achieved.

4.3 32ch Firewire Microphone board

We developed a compact IEEE1394 microphone board for mobile robot purpose. (Fig.4(right)). The board transmits data by using isochronous transmission to host PC, so that no realtime OS is required. Table 1(right) shows specifications of the board.

4.4 Gantry Measurement System

In order to measure the sound field, gantry system has been developed(Fig.7) It has 3 DOFs and it covers $10 \times 7.5 \times 2.2$[m] space. It is controlled by RT-Linux[6]. Maximum speed of the gantry is 2[m/s] and position resolution is about 0.1[mm]. Each sampling point was measured about 5-10[s]. Therefore, covering 5×5[m] space in 25[cm] grid, 5-10[hr] is required to measure sound pressure level.

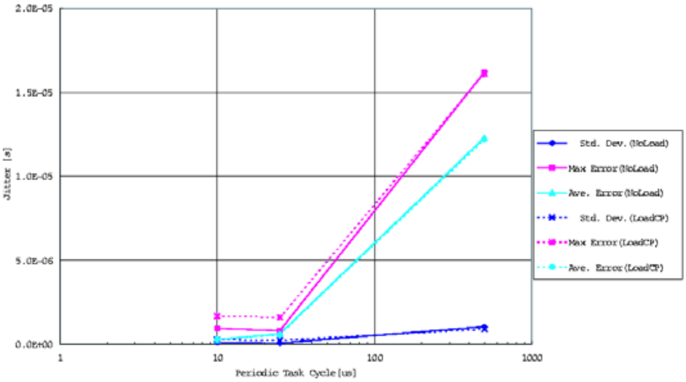


Fig. 5. ART-Linux System Performance with/without loading the system (LoadCP means accessing heavily to NFS mounted HDD)

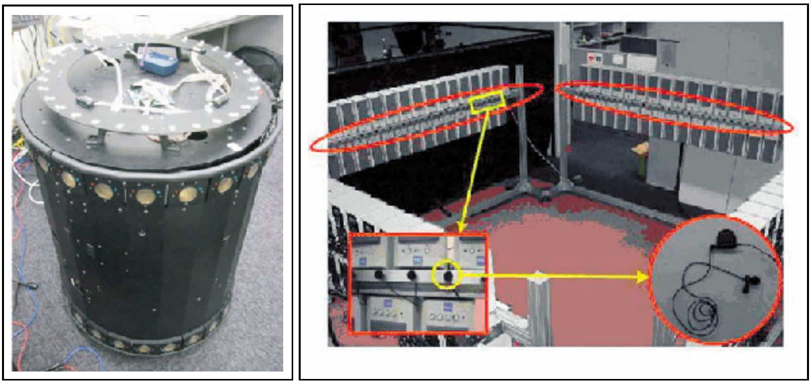


Fig. 6. 32ch circular & 128 square microphone array

5 Experimental Results

The performance of sound capture are evaluated by FFT results, and of sound source separation are done by sound pressure distribution.

5.1 Sound Capture Experiments

In 32ch circular microphone array, origin is set to its center. Two sound sources are placed at point A (1.0, 1.0) [m] and B (-1.0, 1.0) [m] and each sound sources generates 1.0[kHz] and 1.6[kHz] Sin wave simultaneously.

Fig.8(left) shows the experimental results of sound capture by 32ch circular microphone array. The system focuses on point A and B, and two sound data are captured. They are covered over FFT. At point A, the power spectrum of 1.0[kHz] is

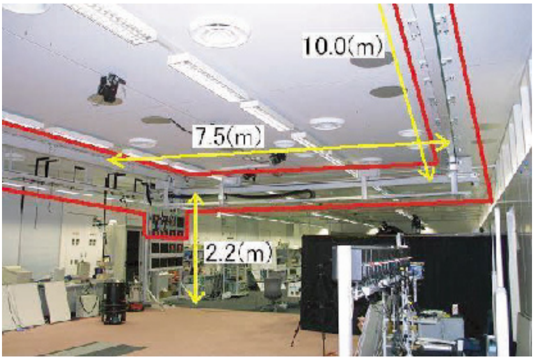


Fig. 7. Gantry System

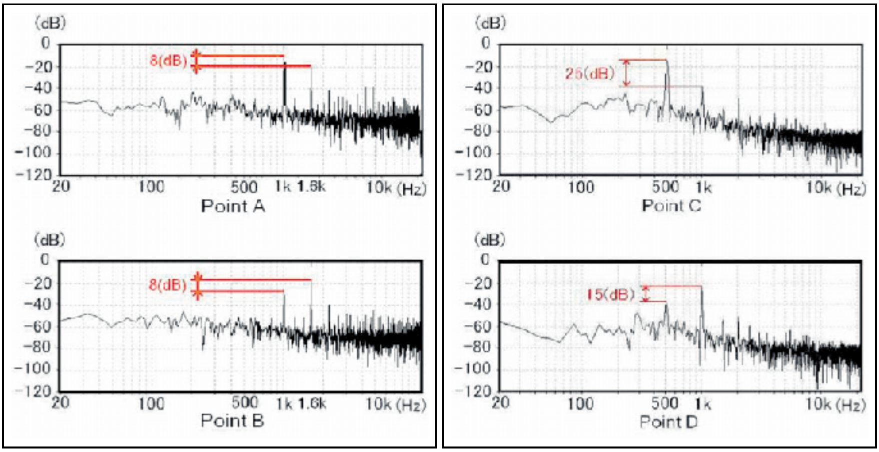


Fig. 8. Sound Capture Experiment by using 32ch Circle Microphone and 128ch Square Microphone Array

about 8[dB] larger than one of 1.6[kHz]. At point B, one of 1.6[kHz] is about 8[dB] larger than one of 1.0[kHz]. The performance of sound capture by the microphone array is a little weak, but the sound can be selectively captured.

In 128ch square microphone array, origin is set to its one vertex. Two sound sources are placed at point C (1.0, 1.0) [m] and point D (2.2, 2.2) [m] and each sound sources generates 0.5[kHz] and 1.0[Hz] Sin wave simultaneously.

Fig.8(right) shows the experimental results of sound capture by 128 channel square microphone array. The system focuses on point C and point D, and two sound data are captured. They are covered over FFT. At point C, the power spectrum of 0.5[kHz] is about 25[dB] larger than one of 1.0[kHz]. At point D, one of 1.0[kHz] is about 15[dB] larger than one of 0.5[kHz].

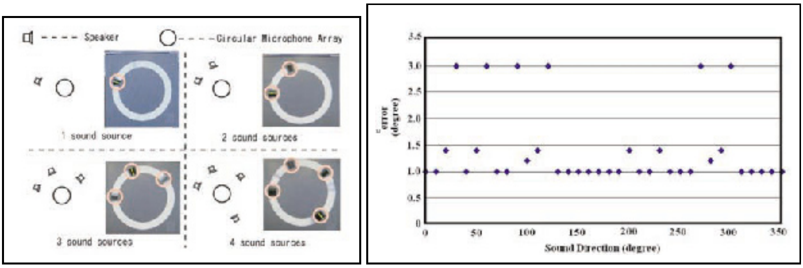


Fig. 9. Sound Localization Experiment and Detection Error by using 32ch Circle Microphone Array

5.2 Sound Localization Experiments

In 32ch circular microphone array, we put one to four speakers around the system. Fig.9(left) shows those conditions. Normal conversation by people are played from those speakers. Fig.9(right) shows the accuracy of sound source direction when only one speaker is put around the system. Maximum error is about 3[deg].

In 128 channel square microphone array, we put two speakers at Focus A (2.2, 2.2) [m] and B (0.8, 0.8) [m], and did the same experiment. The system calculates position of the most loud source by every 1[s] for 10[s].

Fig.10 shows the experimental results of sound source localization by 128 channel square microphone array. 8 sound pressure distribution maps are generated from 10[s] input. There are two areas of high sound pressure in each sound pressure sensitive distribution maps. Accuracy was about 10[cm].

6 Concluding Remarks

In this paper, we described about online sound source localization and capture by using SDBF with number of microphones. Phase shifting parameter is calculated from the distance in between destination point and each microphone. More than one spot can be captured simultaneously. We developed 128ch simultaneous AD board driven by real-time OS (ART-Linux) and 44.1khz sampling cycle was achieved.

In simulation, we evaluate 32 & 128ch circular microphone array, and 128ch square microphone array. Then we implemented 32ch circular and 128ch square microphone array.

The performance of sound capture by 32ch circular microphone array is thin yet, and only direction can be calculated. On the other hand, 128ch microphone array has enough performance to capture and localize the sound source.

In this paper, speech recognition was not able to be evaluated. For a robotics application, cooperation of speech recognition software with the system is indispensable. Improvement in a S/N ratio is a subject too.

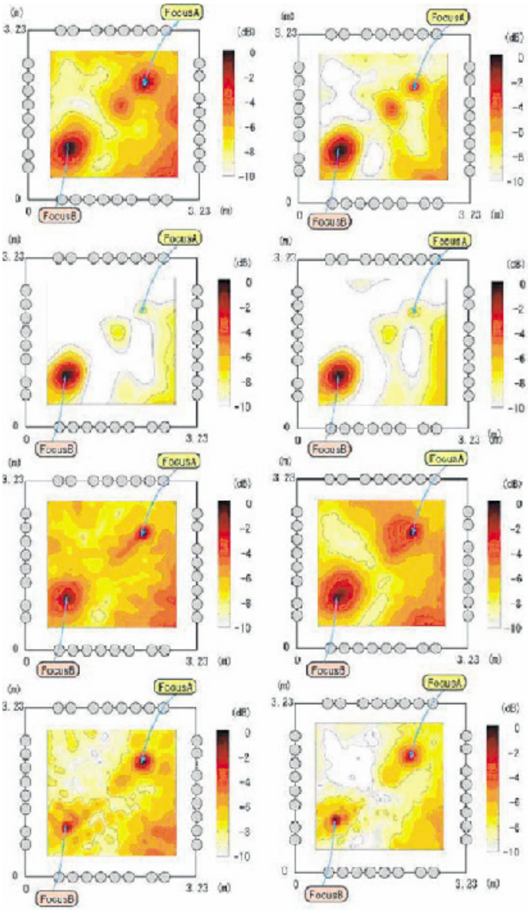


Fig. 10. Sound Localization Experiment by using 128ch Square Microphone Array

References

1. Y. Tamai S. Kagami H. Mizoguchi, K. Shinoda and K. Nagashima. Invisible messenger: Visually steerable sound beam forming system based on face tracking and speaker array. In *Proceedings of the SICE Annual Conference 2003 (SICE2003)*.
2. H. Nomura, Y. Kaneda, and J. Kozima. Microphone array for near sound field. *Journal of Acoustic Society of Japan*, Vol. 53, No. 2, pp. 110–116, 1997.
3. Futoshi Asano, Shiro Ikeda, Michiaki Ogawa, Hideki Asoh and Nobuhiko Kitawaki. A Combined Approach of Array Processing and Independent Component Analysis for Blind Separation of Acoustic Signals.
4. Don H. Johnson and Dan E. Dundgeon. *Array Signal Processing: Concepts and Techniques*. Prentice Hall, 1993. ISBN:0-13-048513-6.
5. Moving Eye Co. *ART-Linux*. http://www.movingeye.co.jp/ỹou1/art-linux/download.html.
6. V. Yodaiken and M.Barabanov. *RT-Linux*. http://www.rtlinux.org.

III. Identification and Control

Control is one of the fundamental building blocks of robotics. To properly execute the desired behaviors, mechanisms are to be controlled efficiently. Though it has been widely researched, the topic still presents plenty of challenges as new applications prompted more rigorous requirements of control.

This chapter presents different facets of control problems: handling of model uncertainties such as friction, new applications of control and robustness to changes in the environment.

The first article by Cheah *et al*, deals with the uncertainties in the kinematic and dynamic model of the robot. The paper argues that no kinematic and dynamic models, including actuator parameters, can be identified and measured correctly. It then proposes a Jacobian-based adaptive controller that updates the estimated task-space velocity and the Jacobian online through a kinematic parameter update law. The proposed algorithm was shown to work on a two-link direct drive robot.

Control application on Shape Memory Alloy (SMA) is explored by Featherstone and Teh, where the material is used both as actuator and feedback sensor. The speed at which the material is heated and cooled is pushed to the safety limit. The electrical resistance of the material is used to keep the material within its safe limits to prevent overheating. The result was demonstrated to be able to control a five-bar-linkage mechanism.

In the third article, Xia *et al*, present an observer-based control algorithm to identify friction parameters of the robot joints online and to compensate these disturbances. Friction has always been difficult to compensate in robotic applications, as it varies considerably with joint position due to tolerances and fiction dependencies to factors such as the room temperature, the state of maintenance of the robot, and how long the robot has been running that day when the experiment is conducted. The method proposed in the article was shown to improve the performance of a PUMA robot.

The last article by Otake, *et al* presents a rather unique way of viewing robot control, as the robot is a lump of polymer gel and the control is done by varying the wave-shape of an electric field. The main challenge was to identify the mapping between the input command in terms of the wave-shape pattern of the electric field, and the output being the shape of the polymer gel ‘robot’.

Experiments on Adaptive Control of Robots with Uncertain Kinematics and Dynamics

C. C. Cheah[1], C. Liu[1] and J.J.E. Slotine[2]

[1] School of Electrical and Electronic Engineering,
Nanyang Technological University,
Block S1, Nanyang Avenue, S(639798), Singapore.

[2] Nonlinear System Laboratory
Massachusetts Institute of Technology,
77 Massachusetts Ave, Cambridge, MA 02139 USA.

Abstract. Most research so far on trajectory tracking control of robot has assumed that the kinematics of the robot is known exactly. In this paper, a new approximate Jacobian adaptive controller is proposed for trajectory tracking of robot with uncertain kinematics, dynamics and actuator model. Experimental results are presented to illustrate the performance of the proposed controller.

1 Introduction

Many robot controllers [1]-[13] have been proposed for robot with dynamics uncertainties and much progress has been made in respect to understanding how the robot can cope with the uncertainties. However, most of these robot controllers have assumed that the kinematics of the robot is known exactly. Unfortunately, no physical parameters could be derived exactly. Even if the parameters could be obtained with a sufficient accuracy, the robot is required to interact with its environment and hence the overall parameters would change according to different tasks. For example, when a robot picks up objects or tools of different lengths, unknown orientations and gripping points, the overall kinematics and dynamics are changing and therefore difficult to derive exactly. To overcome the uncertainties in both kinematics and dynamics, several Approximate Jacobian setpoint controllers [14,15] have been proposed. However, the results in [14,15] are focusing on setpoint control of robot.

Recently, an approximate Jacobian adaptive controller was proposed for trajectory tracking control of robot manipulators [16]. The controller does not require the exact knowledge of kinematics and Jacobian matrix that is assumed in the literature of tracking control. However, it is assumed in [16] that the actuator dynamics is known exactly. Since the actuator model may be uncertain in practice, calibration is necessary to identify the exact parameters of the actuator in implementing the robot controllers. In addition, the actuator parameters could change as temperature varies due to overheating of motor or changes in ambient temperature. Hence, in the presence of the modeling uncertainty or calibration error, the convergence of the tracking error may not be guaranteed. This paper extends the algorithm of [16] to include adaptation to actuator parameters. Experimental results are presented to illustrate the performance of the proposed controller.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 57–66, 2006.

2 Robot Dynamics and Kinematics

If a DC motor driven by an amplifier is used as actuator at each joint of the robot, the dynamics of the robot can be expressed as [2,12]:

$$M(q)\ddot{q} + (\frac{1}{2}\dot{M}(q) + S(q,\dot{q}))\dot{q} + g(q) = Ku, \tag{1}$$

where $M(q) \in R^{n\times n}$ is the inertia matrix, $u \in R^n$ is either a voltage or current inputs to the amplifiers, $K \in R^{n\times n}$ is a diagonal transmission matrix that relates the actuator input u to the control torque, $g(q) \in R^n$ is the gravitational force and

$$S(q,\dot{q})\dot{q} = \frac{1}{2}\dot{M}(q)\dot{q} - \frac{1}{2}\{\frac{\partial}{\partial q}\dot{q}^T M(q)\dot{q}\}^T.$$

Several properties of the dynamic equation (1) are given as follows [2,4,12,19]:
Property 1 The inertia matrix $M(q)$ is symmetric and uniformly positive definite.
Property 2 The matrix $S(q,\dot{q})$ is skew-symmetric.
Property 3 The dynamic model as described by equation (1) is linear in a set of physical parameters $\theta_d = (\theta_{d1}, \cdots, \theta_{dp})^T$ as

$$M(q)\ddot{q} + (\frac{1}{2}\dot{M}(q) + S(q,\dot{q}))\dot{q} + g(q) = Y_d(q,\dot{q},\dot{q},\ddot{q})\theta_d, \tag{2}$$

where $Y_d(\cdot) \in R^{n\times p}$ is called the regressor matrix. ♢

In most applications of robot manipulators, a desired path for the end-effector is specified in task space [2,14,17]. Let $x \in R^n$ be a task space vector defined by,

$$x = h(q), \tag{3}$$

where $h(\cdot) \in R^n$ is a transformation describing the relation between the joint space and task space. The task-space velocity $\dot{x}$ is related to joint-space velocity $\dot{q}$ as:

$$\dot{x} = J(q)\dot{q}, \tag{4}$$

where $J(q) \in R^{n\times n}$ is the Jacobian matrix from joint space to task space.

If cameras are used to monitor the position of the end-effector, the task space is defined as image space in pixels. Let r represents the position of the end-effector in Cartesian coordinates and x represents the vector of image feature parameters [18]. The image velocity vector $\dot{x}$ is related to the joint velocity vector $\dot{q}$ as [18],

$$\dot{x} = J_I(r)J_e(q)\dot{q}, \tag{5}$$

where $J_I(r)$ is the image Jacobian matrix [18] and $J_e(q)$ is the manipulator Jacobian matrix of the mapping from joint space to Cartesian space. In the presence of uncertainties in the camera parameters, the exact image Jacobian matrix and the manipulator Jacobian matrix cannot be obtained. If a position sensor is used to monitor the position of the end-effector, the task space is defined as Cartesian space and hence $J(q) = J_e(q)$ where $J_e(q)$ is the manipulator Jacobian.

A property of the kinematic equation (4) is stated as follows:
Property 4 The right hand side of equation (4) is linear in a set of **kinematic parameters** $\theta_k = (\theta_{k1}, \cdots, \theta_{kq})^T$. Hence, equation (4) can be expressed as,

$$\dot{x} = J(q)\dot{q} = Y_k(q,\dot{q})\theta_k, \tag{6}$$

where $Y_k(q,\dot{q}) \in R^{n\times q}$ is called the kinematic regressor matrix.

3 Adaptive Jacobian Tracking Control

In this section, we present an Adaptive Jacobian Tracking Controller for robot with uncertain kinematic, dynamic and actuator parameters. The main idea is to introduce an adaptive sliding vector using an estimated task-space velocity. The uncertain kinematic parameters of the estimated task-space velocity and the Jacobian matrix are being updated online by a kinematic parameter update law, using the kinematic regressor that we defined in *Property 4*.

Let us define a vector $\dot{x}_r \in R^n$ as,

$$\dot{x}_r = \dot{x}_d - \alpha(x - x_d), \tag{7}$$

where x or $x - x_d$ is measured from a position sensor and $x_d \in R^n$ is a desired trajectory specified in task space and $\dot{x}_d = \frac{dx_d}{dt} \in R^n$ is the desired velocity specified in task space. Differentiating equation (10) with respect to time, we have,

$$\ddot{x}_r = \ddot{x}_d - \alpha(\dot{x} - \dot{x}_d), \tag{8}$$

where $\dot{x} = \frac{dx}{dt}$ is the task-space velocity and $\ddot{x}_d = \frac{d\dot{x}_d}{dt} \in R^n$ is the desired acceleration in task space.

In the presence of kinematic uncertainty, the parameters of the Jacobian matrix is uncertain and hence equation (6) can be expressed as

$$\hat{\dot{x}} = \hat{J}(q,\hat{\theta}_k)\dot{q} = Y_k(q,\dot{q})\hat{\theta}_k, \tag{9}$$

where $\hat{\dot{x}} \in R^n$ denotes an estimated task-space velocity, $\hat{J}(q,\hat{\theta}_k) \in R^{n\times n}$ is an approximate Jacobian matrix and $\hat{\theta}_k \in R^q$ is an estimated kinematic parameters.

To proof the stability of the tracking control system in presence of uncertain kinematics, we define a **adaptive task-space sliding vector** using equation (9) as,

$$\hat{s}_x = \hat{\dot{x}} - \dot{x}_r = \hat{J}(q,\hat{\theta}_k)\dot{q} - \dot{x}_r, \tag{10}$$

where $\hat{J}(q,\hat{\theta}_k)\dot{q} = Y_k(q,\dot{q})\hat{\theta}_k$ as indicated in equation (9). The above vector is adaptive in the sense that the parameters of the approximate Jacobian matrix will be updated by a parameter update law to be defined later. Accordingly, define

$$\dot{\hat{s}}_x = \hat{\ddot{x}} - \ddot{x}_r = \hat{J}(q,\hat{\theta}_k)\ddot{q} + \dot{\hat{J}}(q,\hat{\theta}_k)\dot{q} - \ddot{x}_r, \tag{11}$$

where $\hat{\ddot{x}}$ denotes the derivative of $\hat{\dot{x}}$. Next, let

$$\dot{q}_r = \hat{J}^{-1}(q,\hat{\theta}_k)\dot{x}_r, \tag{12}$$

where $\hat{J}^{-1}(q,\hat{\theta}_k)$ is the inverse of the approximate Jacobian matrix $\hat{J}(q,\hat{\theta}_k)$. In this paper, we assume that the robot is operating in a finite task space such that the approximate Jacobian matrix is of full rank. From equation (12), we have

$$\ddot{q}_r = \hat{J}^{-1}(q,\hat{\theta}_k)\ddot{x}_r + \dot{\hat{J}}^{-1}(q,\hat{\theta}_k)\dot{x}_r, \tag{13}$$

where $\dot{\hat{J}}^{-1}(q,\hat{\theta}_k) = -\hat{J}^{-1}(q,\hat{\theta}_k)\dot{\hat{J}}(q,\hat{\theta}_k)\hat{J}^{-1}(q,\hat{\theta}_k)$. Hence, we have an adaptive sliding vector in joint space as,

$$s = \dot{q} - \dot{q}_r = \hat{J}^{-1}(q,\hat{\theta}_k)((\dot{\hat{x}} - \dot{x}_d) + \alpha(x - x_d)) = \hat{J}^{-1}(q,\hat{\theta}_k)\hat{s}_x, \tag{14}$$

and

$$\dot{s} = \ddot{q} - \ddot{q}_r = \hat{J}^{-1}(q,\hat{\theta}_k)\dot{\hat{s}}_x + \dot{\hat{J}}^{-1}(q,\hat{\theta}_k)\hat{s}_x. \tag{15}$$

Substituting equations (14) and (15) into equation (1) and using *Property 3* yields,

$$M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q,\dot{q}))s + Y_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\theta_d = Ku. \tag{16}$$

We propose an adaptive controller based on the approximate Jacobian matrix and an approximate transmission matrix $\hat{K}$ as,

$$\begin{aligned} u = \hat{K}^{-1}(&-\hat{J}^T(q,\hat{\theta}_k)(K_v\Delta\dot{x} + K_p\Delta x) - \hat{J}^T(q,\hat{\theta}_k)K\hat{s}_x \\ &+Y_d(q,\dot{q},y_r,\dot{y}_r)\hat{\theta}_d + Y_a(\tau_o)\hat{\theta}_a), \end{aligned} \tag{17}$$

where

$$\hat{s}_x = Y_k(q,\dot{q})\hat{\theta}_k - \dot{x}_r, \tag{18}$$

and $\Delta\dot{x} = \dot{x} - \dot{x}_d$, $\Delta x = x - x_d$, $K_v \in R^{n\times n}$, $K_p \in R^{n\times n}$ and $K \in R^{n\times n}$ are symmetric and positive definite matrices, $Y_a(\tau_o) = diag\{-\tau_{o1}, -\tau_{o2}, \ldots, -\tau_{on}\}$ and τ_{oi} denotes the i^{th} element of the vector τ_o which is defined as

$$\tau_o = \hat{J}^T(q,\hat{\theta}_k)(K_v\Delta\dot{x} + K_p\Delta x) + \hat{J}^T(q,\hat{\theta}_k)K_s\hat{s} - Y_d(q,\dot{q},y_r,\dot{y}_r)\hat{\theta}_d. \tag{19}$$

In the proposed controller (17), $\hat{K}^{-1}$ is used to transform the control torque to an approximate actuator input. The first term is an approximate Jacobian transpose sensory feedback law of the task-space velocity and position errors, the second term is an approximate Jacobian transpose feedback law of the adaptive sliding vector (18), the third term is used to compensate the dynamics uncertainty and the last term is used to compensate the uncertainty introduced by the estimated transmission matrix $\hat{K}$. Note that $\hat{\theta}_k$ (like q and $\dot{q}$) is just part of the states of the adaptive control system and hence can be used in the control variables even if it is nonlinear in the variables (provided that a linear parameterization can be found else where in the system model i.e. equation (6)). The estimated kinematic parameters $\hat{\theta}_k$ of the approximate Jacobian matrix $\hat{J}(q,\hat{\theta}_k)$ are updated by,

$$\dot{\hat{\theta}}_k = L_k Y_k^T(q,\dot{q})(K_v\Delta\dot{x} + K_p\Delta x), \tag{20}$$

and the estimated dynamics parameters $\hat{\theta}_d$ of the dynamic model are updated by,

$$\dot{\hat{\theta}}_d = -L_d Y_d^T(q, \dot{q}, \dot{q}_r, \ddot{q}_r)s, \tag{21}$$

where $L_k \in R^{q \times q}$ and $L_d \in R^{p \times p}$ are positive definite matrices. The estimated actuator parameters $\hat{\theta}_a$ are updated by,

$$\dot{\hat{\theta}}_a = -L_a Y_a(\tau_o)s, \tag{22}$$

where $L_a \in R^{n \times n}$ is a positive definite and diagonal matrix.

Substituting equation (17) into (16) and rearranging it, we have

$$\begin{aligned} &M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q, \dot{q}))s + Y_d(q, \dot{q}, y_r, \dot{y}_r)\Delta\theta_d \\ &\quad + \hat{J}^T(q, \hat{\theta}_k)(K_v \Delta\dot{x} + K_p \Delta x) + \hat{J}^T(q, \hat{\theta}_k)K\hat{s}_x \\ &\qquad + (K\hat{K}^{-1} - I)\tau_o - K\hat{K}^{-1}Y_a(\tau_o)\hat{\theta}_a = 0, \end{aligned} \tag{23}$$

where τ_o is defined in equation (19). Since K, $\hat{K}$ and $Y_a(\tau_o)$ are diagonal matrices, the last two terms of equation (23) can be expressed as

$$(K\hat{K}^{-1} - I)\tau_o - K\hat{K}^{-1}Y_a(\tau_o)\hat{\theta}_a = Y_a(\tau_o)(\bar{\theta}_a - K\hat{K}^{-1}\hat{\theta}_a), \tag{24}$$

where $\bar{\theta}_{ai} = 1 - \frac{k_i}{\hat{k}_i}$ and $k_i, \hat{k}_i$ are the i^{th} diagonal elements of K, $\hat{K}$ respectively.

Substituting equation (24) into (23), we have,

$$\begin{aligned} &M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q, \dot{q}))s + Y_d(q, \dot{q}, y_r, \dot{y}_r)\Delta\theta_d \\ &+ \hat{J}^T(q, \hat{\theta}_k)(K_v \Delta\dot{x} + K_p \Delta x) + \hat{J}^T(q, \hat{\theta}_k)K\hat{s}_x + Y_a(\tau_o)\Delta\bar{\theta}_a = 0, \end{aligned} \tag{25}$$

where $\Delta\bar{\theta}_a = \bar{\theta}_a - K\hat{K}^{-1}\hat{\theta}_a$ and hence $\Delta\dot{\bar{\theta}}_a = -K\hat{K}^{-1}\dot{\hat{\theta}}_a$.

Let us define a Lyapunov-like function candidate as

$$\begin{aligned} V &= \tfrac{1}{2}s^T M(q)s + \tfrac{1}{2}\Delta\theta_d^T L_d^{-1}\Delta\theta_d + \tfrac{1}{2}\Delta\theta_k^T L_k^{-1}\Delta\theta_k \\ &\quad + \tfrac{1}{2}\Delta\bar{\theta}_a^T L_a^{-1}\hat{K}K^{-1}\Delta\bar{\theta}_a + \tfrac{1}{2}\Delta x^T(K_p + \alpha K_v)\Delta x, \end{aligned} \tag{26}$$

where $\Delta\theta_k = \theta_k - \hat{\theta}_k$. Differentiating equation (26) with respect to time and using equations (25), 20), (21), (22), (14) and *Property 1*, *Property 2*, we have,

$$\begin{aligned} \dot{V} &= -\hat{s}_x^T K\hat{s}_x - \hat{s}_x^T(K_v \Delta\dot{x} + K_p \Delta x) + \Delta x^T(K_p + \alpha K_v)\Delta\dot{x} \\ &\quad - \Delta\theta_k^T Y_k^T(q, \dot{q})(K_v \Delta\dot{x} + K_p \Delta x). \end{aligned} \tag{27}$$

From equations (18), (6) and (7), we have

$$\hat{s}_x = \Delta\dot{x} + \alpha\Delta x - Y_k(q, \dot{q})\Delta\theta_k, \tag{28}$$

where

$$Y_k(q, \dot{q})\Delta\theta_k = J(q)\dot{q} - \hat{J}(q, \hat{\theta}_k)\dot{q} = \dot{x} - \dot{\hat{x}}, \tag{29}$$

Substituting equation (28) into equation (27) yields,

$$\begin{aligned}\dot{V} &= -(\Delta\dot{x} + \alpha\Delta x - Y_k(q,\dot{q})\Delta\theta_k)^T K(\Delta\dot{x} + \alpha\Delta x - Y_k(q,\dot{q})\Delta\theta_k) \\ &\quad -\Delta\dot{x}^T K_v \Delta\dot{x} - \alpha\Delta x^T K_p \Delta x \le 0. \end{aligned} \tag{30}$$

We are now in a position to state the following Theorem:

Theorem The approximate Jacobian adaptive control law (17) and the parameter update laws (20), (21) and (22) for the robot system (1) result in the convergence of position and velocity tracking errors. That is, $x - x_d \to 0$ and $\dot{x} - \dot{x}_d \to 0$, as $t \to \infty$. In addition, the estimated task-space velocity converges to the actual task-space velocity, i.e. $\hat{\dot{x}} \to \dot{x}$, $t \to \infty$.

Proof:

Since $M(q)$ is uniformally positive definite, V in equation (26) is positive definite in s, Δx, $\Delta\theta_k$, $\Delta\theta_d$ and $\Delta\bar{\theta}_a$. Since $\dot{V} \le 0$, V is also bounded, and therefore s, Δx, $\Delta\theta_k$, $\Delta\theta_d$ and $\Delta\bar{\theta}_a$ are bounded vectors. This implies that $\hat{\theta}_k$, $\hat{\theta}_d$, $\hat{\theta}_a$ are bounded, x is bounded if x_d is bounded, and $\hat{s}_x = \hat{J}(q,\hat{\theta}_k)s$ is bounded as seen from equation (14). Since Δx is bounded, $\dot{x}_r$ in equation (7) is also bounded if $\dot{x}_d$ is bounded. Therefore, $\dot{q}_r$ in equation (12) is also bounded if the approximate Jacobian matrix is non singular. From equations (14), $\dot{q}$ is bounded and the boundedness of $\dot{q}$ means that $\dot{x}$ is bounded since the Jacobian matrix is bounded. Hence, $\Delta\dot{x}$ is bounded and $\ddot{x}_r$ in equation (8) is also bounded if $\ddot{x}_d$ is bounded. From equation (20), $\dot{\hat{\theta}}_k$ is therefore bounded since Δx, $\Delta\dot{x}$, $\dot{q}$ are bounded and $Y_k(\cdot)$ is a trigonometric function of q. Therefore, $\ddot{q}_r$ in equation (13) is bounded. From the closed-loop equation (25), we can conclude that $\dot{s}$ is bounded. The boundedness of $\dot{s}$ imply the boundedness of $\ddot{q}$ as seen from equation (15). From equation (11), $\dot{\hat{s}}_x$ is therefore bounded. Finally, differentiating equation (28) with respect to time and re-arranging yields,

$$\Delta\ddot{x} + \alpha\Delta\dot{x} = \dot{\hat{s}}_x + \dot{Y}_k(q,\dot{q},\ddot{q})\Delta\theta_k - Y_k(q,\dot{q})\dot{\hat{\theta}}_k, \tag{31}$$

which means that $\Delta\ddot{x} = \ddot{x} - \ddot{x}_d$ is also bounded.

To apply Barbalat's lemma, let us check the uniform continuity of $\dot{V}$. Differentiating equation (30) with respect to time gives,

$$\ddot{V} = -2\hat{s}_x^T K \dot{\hat{s}}_x - 2\Delta\dot{x}^T K_v \Delta\ddot{x} - 2\alpha\Delta x^T K_p \Delta\dot{x}. \tag{32}$$

This shows that $\ddot{V}$ is bounded since $\hat{s}_x$, $\dot{\hat{s}}_x$ and Δx, $\Delta\dot{x}$, $\Delta\ddot{x}$ are all bounded. Hence, $\dot{V}$ is uniformly continuous. Using Barbalat's lemma, we have $\Delta x = x - x_d \to 0$, $\Delta\dot{x} = \dot{x} - \dot{x}_d \to 0$, and $\hat{s}_x \to 0$ as $t \to \infty$. Finally, from equation (29), the convergence of Δx, $\Delta\dot{x}$ and $\hat{s}_x$ imply that $\dot{x} - \hat{\dot{x}} \to 0$, as $t \to \infty$. △△△

Remark 1. Note that the need for using task-space velocity can be eliminated by using a filtered differentiation of the measured position x. That is, one can create a known signal y using,

$$y = \frac{\lambda p}{p + \lambda}\, x = W_k(t)\theta_k, \tag{33}$$

where $W_k(t) = \frac{\lambda}{p+\lambda} Y_k(q, \dot{q})$, p is the Laplace variable, $y(0) = 0$ and $W_k(0) = 0$. The actual task-space velocity error $\Delta\dot{x}$ in the feedback law (17) is also replaced by the estimated task-space velocity error $\Delta\hat{\dot{x}} = Y_k(q, \dot{q})\hat{\theta}_k - \dot{x}_d$. The estimated kinematic parameters $\hat{\theta}_k$ of the approximate Jacobian matrix $\hat{J}(q, \hat{\theta}_k)$ are updated by the following **composite kinematic update law**,

$$\dot{\hat{\theta}}_k = -L_k W_k^T(t) K_v (W_k(t)\hat{\theta}_k - y) + L_k Y_k^T(q, \dot{q}) K_p \Delta x. \tag{34}$$

The proof can be established similarly by defining a Lyapunov-like function as

$$\begin{aligned} V = {} & \tfrac{1}{2} s^T M(q) s + \tfrac{1}{2}\Delta\theta_d^T L_d^{-1}\Delta\theta_d + \tfrac{1}{2}\Delta\theta_k^T L_k^{-1}\Delta\theta_k \\ & + \tfrac{1}{2}\Delta\bar{\theta}_a^T L_a^{-1}\hat{K}K^{-1}\Delta\bar{\theta}_a + \tfrac{1}{2}\Delta x^T K_p \Delta x, \end{aligned} \tag{35}$$

and noting that,

$$\dot{V} = -[\Delta x^T \, \Delta\hat{\dot{x}}^T] \begin{bmatrix} \alpha K_p & \frac{\alpha K_v}{2} \\ \frac{\alpha K_v}{2} & K_v \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\hat{\dot{x}} \end{bmatrix} - \Delta\theta_k^T W_k^T(t) K_v W_k(t) \Delta\theta_k, \tag{36}$$

where $K_p > \frac{\alpha}{4} K_v$ yields $\dot{V} \leq 0$.

Remark 2. In this paper, we assume that the robot is operating in a finite task space such that the approximate Jacobian matrix is of full rank. Note from equation (12) that $\hat{J}^{-1}(q, \hat{\theta}_k)$ is used only in the definition of control variable $\dot{q}_r$. Therefore, we should be able to control this by bounding the variable or using a singularity-robust inverse of the approximate Jacobian matrix [20]. Often, singularities only depend on q, not $\hat{\theta}_k$. It may also be possible to avoid boundary singularities by originally overestimating the size of the robot.

4 Experimental Results

To illustrate the performance of the approximate Jacobian tracking controllers, we implemented the proposed controller on a 2-link direct drive robot as shown in figure 1, using a personal computer (PC) with Pentium II processor. The controllers are running in Microsoft Windows NT together with VenturCom's RTX, a hard real-time extension. The sampling period is set as 3 ms. The PC is mounted with a servo-to-go I/O card which consists of a digital to analog (DA) converter, analog to digital (AD) converter and a pulse counter. Joint angles are obtained from two incremental encoders located on the motors. Joint velocities are obtained from differentiation of the joint angles. A PSD camera (position sensitive detector) manufactured by Hamamatsu is used to measure the position of the robot end effector. The task-space velocities are obtained from differentiation of the measured position.

The robot is required to hold an object with uncertain length and grasping angle and follow a circular trajectory specified in Cartesian space as:

$$X_d = 0.33 + 0.1\sin(0.54 + 3t), \quad Y_d = 0.41 + 0.1\cos(0.54 + 3t). \tag{37}$$

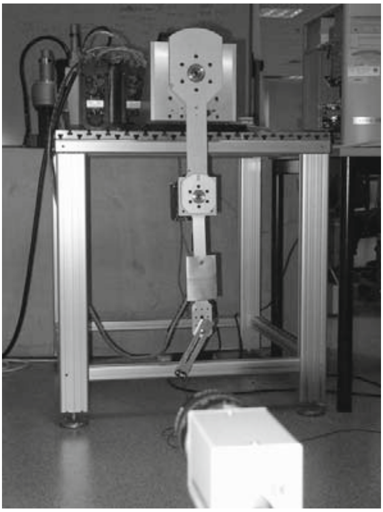


Fig. 1. A 2-link Direct Drive Robot

The relationship between the velocities in task space and the velocities of the joints is given by [16]

$$\begin{aligned}\dot{x} = J(q)\dot{q} &= \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_o s_{12o} & -l_2 s_{12} - l_o s_{12o} \\ l_1 c_1 + l_2 c_{12} + l_o c_{12o} & l_2 c_{12} + l_o c_{12o} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \\ &= \begin{bmatrix} -s_1 \dot{q}_1 & -s_{12}(\dot{q}_1 + \dot{q}_2) & -s_{12}(\dot{q}_1 + \dot{q}_2) & -c_{12}(\dot{q}_1 + \dot{q}_2) \\ c_1 \dot{q}_1 & c_{12}(\dot{q}_1 + \dot{q}_2) & c_{12}(\dot{q}_1 + \dot{q}_2) & -s_{12}(\dot{q}_1 + \dot{q}_2) \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ l_o c_o \\ l_o s_o \end{bmatrix},\end{aligned} \tag{38}$$

where l_1, l_2 are the link lengths, l_0 and q_0 are the length and grasping angle of the object respectively, $c_1 = \cos q_1, s_1 = \sin q_1, c_{12} = \cos(q_1 + q_2), s_{12} = \sin(q_1 + q_2)$, $c_{12o} = \cos(q_1 + q_2 + q_o)$, $s_{12o} = \sin(q_1 + q_2 + q_o)$, $c_o = \cos q_o$, $s_o = \sin q_o$,.

The proposed controller in the Theorem was implemented on the robot holding objects with different lengths and grasping angles. In the first experiment, the length of the object was approximately set as $0.06m$ and the grasping angle was approximately set as 45^o. The object length and grasping angle were estimated as $\hat{l}_o(0) = 0.12m$ and $\hat{q}_o(0) = 20^o$ respectively and the link lengths were set as $\hat{l}_1(0) = 0.25m$, $\hat{l}_2(0) = 0.27m$. The initial position of the robot end effector was specified as $(X(0), Y(0)) = (0.28, 0.52)$. Uncertainty in actuator model is also introduced by changing the estimated actuator model from $K = [2.8, 1.09]$ to $\hat{K} = [2.45, 0.95]$. Experimental results with $L_a = diag\{0.15, 0.10\}$, $L_k = diag\{0.13, 0.13, 0.02, 0.015\}$, $L_d = diag\{0.01, 0.002, 0.002, 0.002, 0.015, 0.01, 0.01\}$, $K_v = diag\{2, 2\}$, $K_p = diag\{420, 380\}$, $K = diag\{1, 1\}$, $\alpha = 1.2$ are presented in figure 2. As seen from the results, the tracking errors converge with updating of the estimated actuator, kinematic and dynamic parameters.

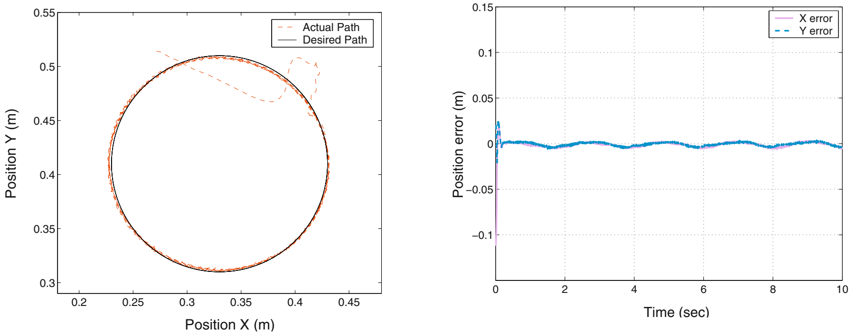


Fig. 2. Path and position errors of the end effector holding first object

In the second experiments, the length of the object was approximately set as $0.1m$ and the grasping angle was approximately set as 60^o. The experiment results are shown in figure 3.

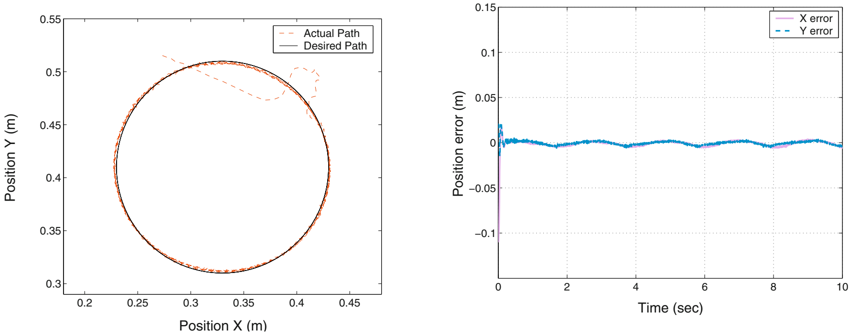


Fig. 3. Path and position errors of the end effector holding second object

5 Conclusion

We have proposed an approximate Jacobian adaptive controller for the tracking control of robot with uncertain kinematics, dynamics and actuator model. Novel parameter update laws are proposed to update uncertain kinematics, dynamics and actuator parameters. We have shown that the robot end effector is able to track a desired trajectory with the uncertain parameters being updated online by the proposed parameter update laws. Experimental results illustrate the performance of the proposed controller.

References

1. M. Takegaki and S. Arimoto, “A new feedback method for dynamic control of manipulators,” *ASME J. of Dynamic Systems, Measurement and Control*, vol. 102, pp. 119–125, 1981.
2. S. Arimoto, *Control Theory of Nonlinear Mechanical Systems - A Passivity-Based and Circuit-Theoretic Approach*. Oxford: Clarendon Press, 1996.
3. J. J. Craig, P. Hsu, and S. S. Sastry, “Adaptive control of mechanical manipulators,” *Int. J. Robotics Research*, vol. 6, no. 2, pp. 16–28, 1987.
4. J. J. E. Slotine and W. Li, “On the adaptive control of robot manipulators,” *Int. J. Robotics Research*, no. 6, pp. 49–59, 1987.
5. J. J. E. Slotine and W. Li, “Adaptive manipulator control: A case study,” *IEEE Trans. on Automatic Control*, vol. 33, no. 11, pp. 995–1003, 1988.
6. R. H. Middleton and G. C. Goodwin, “Adaptive computed torque control for rigid link manipulators,” *Systems and Control Letter*, vol. 10, pp. 9–16, 1988.
7. D. E. Koditschek, “Adaptive techniques for mechanical systems,” in *Fifth Yale Workshop on Applications of Adaptive Systems Theory*, (New Haven, CT), pp. 259–265, 1987.
8. R. Ortega and M. W. Spong, “Adaptive motion control of rigid robots: a tutorial,” *Automatica*, vol. 25, no. 6, pp. 877–888, 1989.
9. N. Sadegh and R. Horowitz, “Stability and robustness analysis of a class of adaptive controllers for robotic manipulators,” *Int. J. Robotics Research*, vol. 9, no. 3, pp. 74–92, 1990.
10. H. Berghuis, R. Ortega and H. Nijmeijer, "A robust adaptive robot controller", *IEEE Trans. on Robotics and Automation*, vol. 9, no. 6, pp 825 - 830, 1993.
11. L. L. Whitcomb, A. Rizzi, and D.E. Koditschek, "Comparative Experiments with a New Adaptive Controller for Robot Arms", *IEEE Trans. on Robotics and Automation*, vol. 9, no. 1, pp 59-70, 1993.
12. F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators*. New York: Macmillan Publishing Company, 1993.
13. L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. New York: Springer-Verlag, 2000.
14. C. C. Cheah, S. Kawamura, and S. Arimoto, “Feedback control for robotic manipulators with an uncertain jacobian matrix,” *J. of Robotic System*, vol. 12, no. 2, pp. 119–134, 1999.
15. C. C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, “Approximate jacobian control for robots with uncertain kinematics and dynamics,” *IEEE Trans. on Robotics and Automation*, vol. 19, no. 4, pp. 692–702, 2003.
16. C. C. Cheah, C. Liu and J.J.E. Slotine, "Approximate Jacobian Adaptive Control for Robot Manipulators", in *Proc. of IEEE Int. Conference on Robotics and Automation*, (New Orleans, USA), pp 3075-3080, 2004.
17. O. Khatib, “A unified approach for motion and force control of robot manipulators” *IEEE J. Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
18. G. H. S. Hutchinson and P. Corke, “A tutorial on visual servo control,” *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651 – 670, 1996.
19. J. J. E. Slotine and W. Li, *Applied nonlinear control*. Englewood Cliffs, New Jersey: Prentice Hall, 1991.
20. Y. Nakamura, *Advanced Robotics*. Reading MA: Addison-Wesleyn, 1985.

Improving the Speed of Shape Memory Alloy Actuators by Faster Electrical Heating

Roy Featherstone[1] and Yee Harn Teh[2]

[1] Dept. Systems Engineering
Australian National University
Canberra ACT 0200, Australia
roy@syseng.anu.edu.au
http://www.syseng.anu.edu.au/~roy/

[2] Dept. Systems Engineering
Australian National University
yee.teh@rsise.anu.edu.au

Abstract. This paper describes a method for improving the speed of actuators based on shape memory alloys (SMA) by increasing the rate at which an SMA element can safely be heated. The method consists of measuring the electrical resistance of an SMA element, calculating a maximum safe heating current as a function of measured resistance, and ensuring that the actual heating current does not exceed this maximum value. In effect, resistance is being used as a form of temperature measurement, and the maximum safe heating current is designed to prevent overheating. This method has been incorporated into a two-stage relay controller that controls the motion of a pantograph robot actuated by two antagonistic pairs of SMA wires. Experimental results show a substantial increase in the maximum velocity attainable by this robot, without any change in the cooling regime, purely as a result of faster heating.

1 Introduction

Actuators based on shape memory alloys (SMA) have a variety of uses in robotics [4,7–9]. Their advantages include mechanical simplicity, high power-to-weight ratio, small size, and clean, silent, spark-free operation. On the other hand, they also possess some serious disadvantages, including inefficiency, hysteresis and slow speed.

A typical SMA-based actuator contains an element, comprising one or more wires, coils, or formed shapes of SMA material. These elements can be stretched easily when cool, but contract forcibly when hot. In robotic applications, they are typically arranged in antagonistic pairs. An element is usually heated by means of joule heating (i.e., passing an electrical current through the element), and cooled by means of heat transfer to the environment. The limiting factors on the speed of such an actuator are the heating and cooling rates of the SMA elements. The cooling rate can be increased by various means, including forced-air cooling, oil or water cooling, and using thinner SMA wires; but the heating rate can be increased simply by passing a larger current through the element.

In principle, an SMA element can be heated arbitrarily quickly by passing a sufficiently large current through it. However, currents beyond a certain magnitude have the capacity to overheat the SMA, causing permanent damage. Thus, any strategy

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 67–76, 2006.

for electrical heating of an SMA element must take steps to avoid overheating. The simplest strategy is to limit the heating current to a known safe value, such as can be found in an SMA data sheet. A better strategy is to monitor the state of the SMA, and apply a current that is safe for the SMA in its current state. If the SMA is cool, for example, then it is safe to apply a very large heating current for as long as the SMA remains below some threshold temperature.

One such method has been proposed by Kuribayashi [6]. His method involves:-

1. measuring the temperature of the SMA element directly by means of a miniature thermocouple attached to the SMA element;
2. allowing a large heating current whenever the temperature is below a threshold value; and
3. setting the heating current to zero whenever the temperature is above the threshold.

This heating strategy was incorporated into a simple motion control system, and experimental results showed a large improvement in actuator response times.

We present an alternative heating method in this paper, that uses a resistance measurement of the SMA instead of a direct temperature measurement [2]. The main advantage of the new method is that it dispenses with the need for a special sensor co-located with the SMA element. To demonstrate the practical benefit of this method, it was incorporated into a simple motion control system based on Grant's two-stage relay controller [3], and used to control the motions of a pantograph robot actuated by four SMA wires arranged in antagonistic pairs. Experimental results show that the new heating method was able to double the maximum speed of the actuators, compared with a heating regime in which the heating current is limited to the data-sheet value. This improvement was achieved without any change in the cooling regime, and can therefore be attributed entirely to faster heating.

In the rest of this paper, we describe the principle of operation of the new method, the experimental hardware used to test it, and experimental results from the motion controller showing the resulting improvement in speed.

2 Principle of Operation

The shape memory effect is caused by a phase transition between two crystalline phases: a low-temperature phase called martensite, and a high-temperature phase called austenite. In a typical SMA, like nitinol, the two phases have different resistivities. This causes a phase-related change in resistance as an SMA element is heated or cooled. This phenomenon is well known, and has been used before as a means of estimating the martensite ratio for the purpose of servoing the phase transition directly [5].

Figure 1 shows a (simplified) plot of resistance versus temperature that would be typical of an alloy like nitinol. As the alloy is heated from cold, it begins to transform into the austenite phase at a temperature called the *austenite start* temperature, A_s, and the transformation is essentially complete at a temperature called the *austenite*

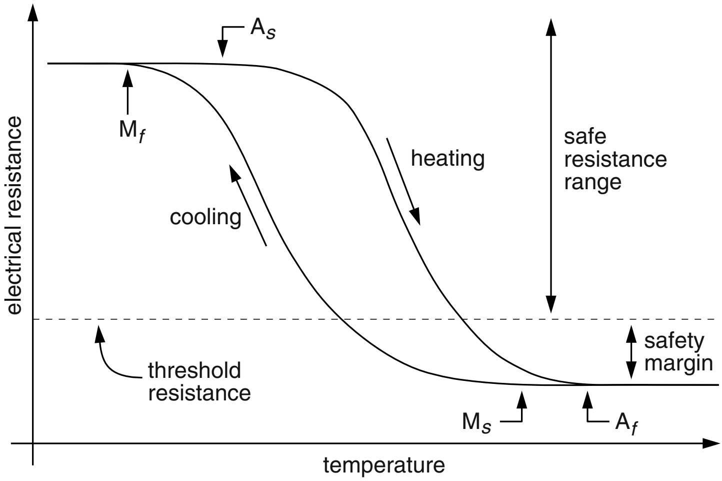


Fig. 1. Typical plot of resistance versus temperature for nitinol.

finish temperature, A_f. Likewise, as the hot alloy is cooled, it transforms back into the martensite phase between the martensite start and finish temperatures, M_s and M_f. These temperatures are lower than those for the heating curve, because of thermal hysteresis in the transformation. For an alloy like nitinol, the resistance changes by about 20%.

The presence of thermal hysteresis means that we cannot deduce the exact temperature from the resistance; but we can identify a threshold resistance, R_{thresh}, that marks the boundary between 'safe' resistances and 'possibly unsafe' resistances. This quantity is defined to be the resistance of the hot SMA element, plus a safety margin that accounts for resistance measurement errors and strain-induced resistance changes. If R_{thresh} applies to a batch of SMA elements, rather than an individual element, then the safety margin must also account for possible variations in hot resistance from one element to the next.

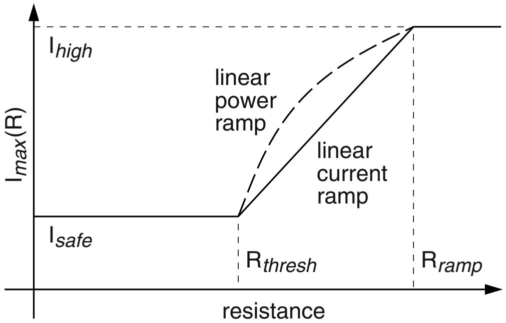


Fig. 2. Maximum safe current, $I_{max}(R)$, vs. SMA element resistance.

Given R_{thresh}, we can define a maximum safe heating current, $I_{max}(R)$, which is a function of the measured resistance of the SMA element, as shown in Figure 2. In this figure, I_{safe} is a current that can safely be applied to the element indefinitely without risk of overheating, but is sufficient to keep the temperature above A_f. Suitable values for I_{safe} can be found in data sheets for SMA actuator wires, e.g. [1]. I_{high} is the maximum value of the rapid-heating current. I_{high} will typically be determined by limits on the available power, e.g. the maximum voltage or current available from the power supply, or the current-handling capacity of the individual regulators supplying each SMA element. R_{ramp} is an optional additional resistance threshold on the safe side of R_{thresh}, and its purpose is to allow a smooth transition between I_{safe} and I_{high}. Two obvious possibilities are shown in the figure: a linear current ramp and a linear power ramp. If a ramp is not required, then I_{max} is a discontinuous function of SMA resistance, given by

$$I_{max}(R) = \begin{cases} I_{high} & \text{if } R \geq R_{thresh} \\ I_{safe} & \text{otherwise.} \end{cases} \tag{1}$$

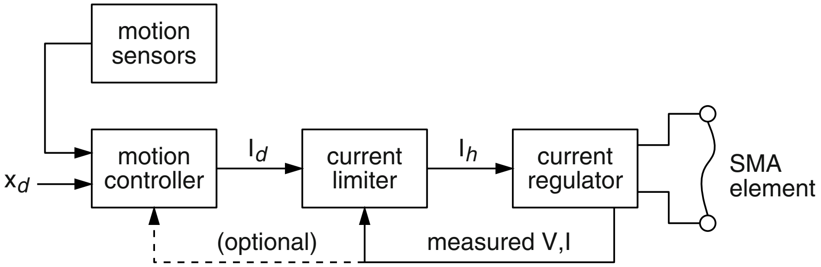
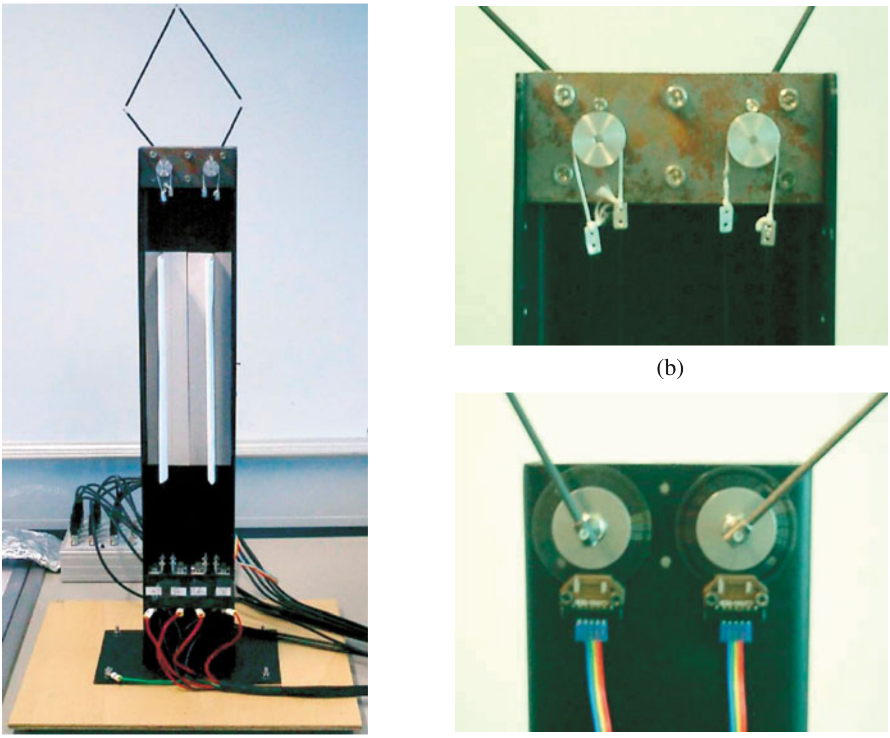


Fig. 3. Motion control system with current limiter.

Figure 3 shows one possible way to incorporate the rapid heating method into a motion control system. In this case, a motion controller receives a desired position input signal, x_d, and signals from one or more motion sensors. Based on this data, it calculates an output signal, I_d, that is interpreted as the desired heating current for a particular SMA element. This signal is input to a current limiter that calculates an actual heating current, I_h, according to the formula

$$I_h = \min(I_d, I_{max}(R_{meas}))\,. \tag{2}$$

This signal is then passed to a current regulator, which causes a current of I_h to pass through the SMA element. Measurements of the actual voltage across the SMA element, and the actual current passing through it, are passed back to the current limiter so that it can work out the resistance of the SMA element. These signals could also be passed back to the motion controller, if it has been designed to use them. Feedback of the actual SMA current is not necessary if the current regulator is sufficiently precise.



(b)

(a)

(c)

Fig. 4. The experimental rig (a), and detailed views of the pulleys (b) and optical encoders (c).

3 Experimental Hardware

The experimental rig is shown in Figure 4(a), and schematically in Figure 5. It consists of a vertical metal C-beam, about 0.7m high, supporting two horizontal shafts at the top and eight anchor points at the bottom. Each shaft rotates freely on ball bearings, and carries a small pulley at the front and an optical encoder wheel at the rear, as shown in Figures 4(b) and 4(c). The two shafts terminate in small sockets, welded to the rear end of each shaft, which hold the ends of a pantograph linkage made from carbon tubes. The pantograph serves as a mechanical load.

A short chord is wrapped 1.5 times around each pulley, and is affixed to the pulley at its centre so that it cannot slip relative to the pulley. Each end of each chord terminates in an eyelet. Four Flexinol (tm) wires are strung between the eight anchor points and four eyelets, as shown in Figure 5, to form two antagonistic pairs of SMA elements. The wires are approximately 1m long and 0.1mm in diameter, and they are too thin to be visible in the photographs in Figure 4. The white objects half way up the column in Figure 4(a) are separators, made from paper, which prevent the wires from making physical (and therefore electrical) contact with their neighbours in the event they go slack.

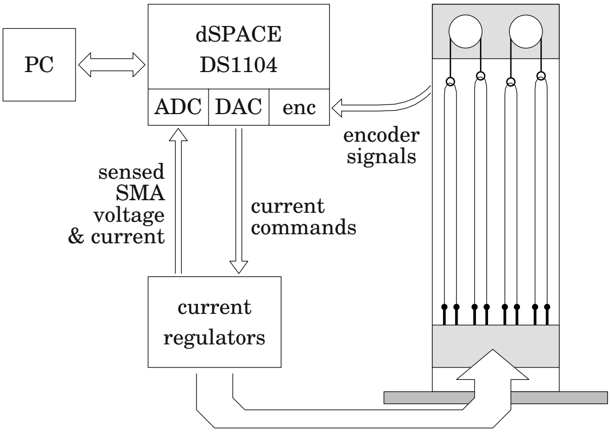


Fig. 5. Schematic diagram of experimental hardware.

Figure 5 shows a schematic of the complete experimental setup. All real-time computation and data capture functions are performed on a DS1104 board from dSPACE, which communicates with and is programmed from a PC. The four DAC outputs from the DS1104 are connected to four current regulators, which supply electrical power to the SMA wires. Each regulator is capable of delivering more than 0.65A (40W) to its load, which is more than enough to burn out the SMA wires. The actual voltage across each wire, and the actual current passing through it, are sensed at taps in the current regulator circuits, and the signals are passed back to the ADC inputs on the DS1104. The magnitudes of these signals are such that we have an effective resolution of 10.5 bits on the SMA voltage signals and 12 bits on the SMA current signals. However, all the signals are polluted with noise, especially the current signals, and we found it necessary to pass them through low-pass filters before we could get accurate resistance measurements.

The optical shaft encoders also connect directly to the DS1104. They have a resolution of 8192 counts per revolution, but the motion range of the pulleys is only slightly more than 90°, so we have an effective resolution of 11 bits on shaft angle measurements.

4 Experimental Results

We have implemented a version of Grant's two-stage relay controller for antagonistic pairs of SMA elements [3], and incorporated the rapid-heating method into the controller, essentially as shown in Figure 3. The two-stage relay controller implements the following control law:

$$(I_{Fd}, I_{Rd}) = \begin{cases} (0, I_H) & \theta_{err} < -\phi \\ (0, I_L) & -\phi \le \theta_{err} < 0 \\ (I_L, 0) & 0 \le \theta_{err} < \phi \\ (I_H, 0) & \phi \le \theta_{err} \end{cases} \tag{3}$$

where θ_{err} is the position error, I_{Fd} and I_{Rd} are the desired heating currents for the forward and reverse SMA elements, respectively, and I_H, I_L and ϕ are parameters of the controller. I_H is the current to apply when the position error is large; I_L is the current to apply when the position error is small; and ϕ defines the boundary between small and large position errors. The forward and reverse SMA elements pull in the positive and negative directions, respectively, as measured by the position sensor. The two desired heating currents, I_{Fd} and I_{Rd}, are sent to two current limiters, one for each SMA element, which calculate the actual heating currents to be sent to each element.

We have tested this controller under a variety of conditions. Overall, the rapid heating mechanism substantially improves the speed, but the tracking accuracy is poor because of large limit cycles. The two-level relay controller is known to suffer from large limit cycles [3], but the larger velocities produced by the rapid heating mechanism appear to exacerbate the problem.

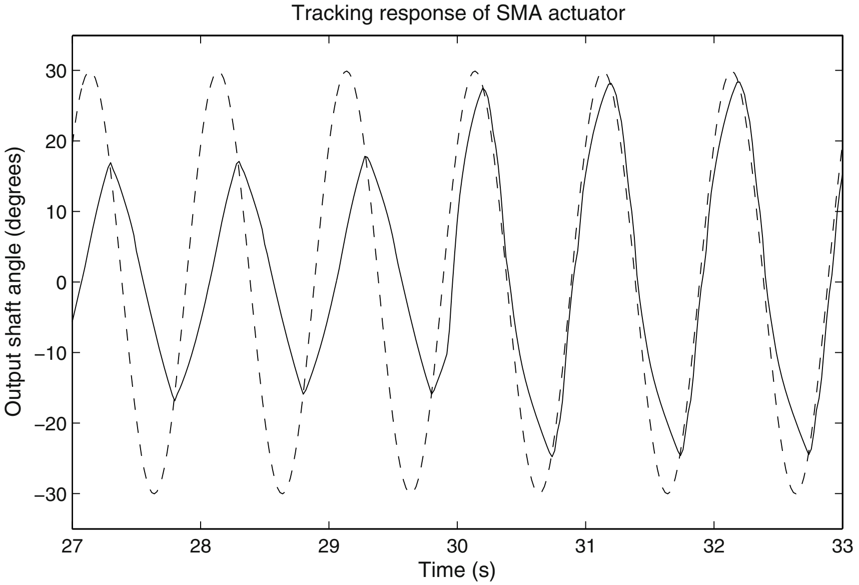


Fig. 6. Tracking response (solid line) of an antagonistically actuated pulley to a 1Hz sinusoidal position command (dashed line).

A representative example of improved tracking speed is shown in Figure 6. This graph shows the tracking response of a single antagonistic pair, under no-load conditions (i.e., without the pantograph), to a 1Hz sine wave input. (Limit cycles are much smaller under no-load conditions.) The $60°$ peak-to-peak magnitude of the sine wave corresponds to a 1.6% strain in the SMA wires.

The parameters used in this experiment were: $I_H = I_{high}$, $I_L = I_{safe}$ and $\phi = 3°$ for the controller, and $I_{high} = 0.42\text{A}$, $I_{safe} = 0.18\text{A}$, $R_{thresh} = 105\Omega$

and $R_{ramp} = 118\Omega$ for the two current limiters. A linear power ramp was used for $I_{max}(R)$ between R_{thresh} and R_{ramp}.

These parameters were selected as follows. I_{high} is the current required to produce approximately 20W of joule heating in the SMA wires used. This is about half the power available from the current regulators, but many times more than the safe heating power. I_{safe} is taken directly from the data sheet for 0.1mm Flexinol wires [1]. ϕ was chosen to minimise limit cycle magnitude, and was determined empirically. R_{thresh} is the measured hot resistance of one of the SMA wires, plus a 4% safety margin to account for measurement errors, strain-induced variations in resistance, and variations in hot resistance from one wire to the next. (We did not measure the hot resistance of every individual wire.) R_{ramp} was set equal to the cold resistance of an unstretched SMA wire, for no particularly good reason. A lower value for R_{ramp} might have produced better results.

In separate experiments, we determined the largest measurement noise source to be the SMA current measurement, which exhibited an RMS noise of 1.4mA. This amounts to a 0.8% error at currents close to I_{safe} and a proportionately larger error at smaller currents. We removed some of this noise by passing both the SMA current signal and the SMA voltage signal through identical low-pass filters before using them. We also measured the effect of a full load (3 Newtons) on both the strain and resistance of a wire heated by I_{safe}, and found that it caused 0.2% strain and a 0.9% increase in resistance.

Figure 6 shows the tracking response both with and without the benefit of rapid heating. For the first 30 seconds, I_{high} is set equal to I_{safe}, which effectively disables the rapid-heating mechanism. During this period, the controller is unable to track the input signal accurately because the actuator moves too slowly. Then, on the 30 second mark, I_{high} is set to the value given above. This causes an immediate increase in maximum actuator velocity, although it is still not quite fast enough to track the whole of the sine wave accurately.

With the parameters used in this experiment, the rapid-heating mechanism increases the peak actuator velocity by a factor of 2, from approximately $90°/s$ to $180°/s$. Note that this speed-up is due entirely to faster heating. The cooling regime (ambient air cooling) does not change.

The tracking response shows a noticeable asymmetry after the rapid heating mechanism is enabled, and also a substantial reduction in the contraction rate of each wire as it gets shorter. The former is probably due either to one wire being slightly shorter than the other, or to the two eyelets not being exactly level when the pulley angle is zero. The latter suggests there may be some speed advantage, at the expense of motion range, in pre-straining the wires at installation.

Figure 7 shows the actual power delivered to the two SMA wires during this experiment, in the vicinity of the 30-second mark. The data in this graph comes from the SMA voltage and current measurements, which have been low-pass filtered as explained above. The spike in the centre corresponds to the moment when the rapid-heating mechanism is enabled. Prior to this moment, the tracking errors are large, and so the relay controller requests a current of I_H to be sent to each wire in

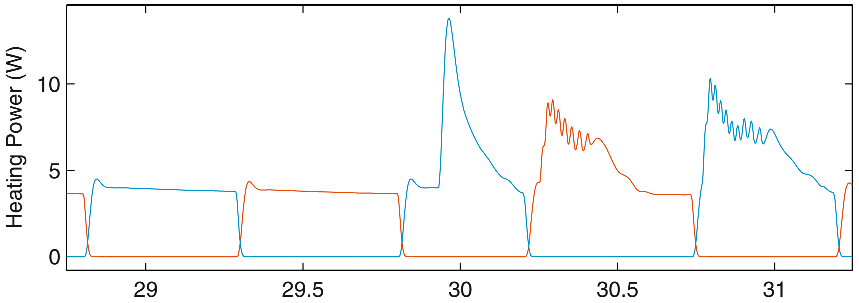


Fig. 7. Actual power delivered to the two actuator wires during the motion shown in Figure 6.

turn; but the actual current is limited to I_{safe} by the current limiter. Immediately after rapid heating is enabled, the tracking error is still large, so the relay controller continues to request I_H be sent to the contracting wire; but now the actual current is computed according to the resistance of the wire. Initially, the wire is relatively cool, and the heating power rises to 14W. The wire then heats very rapidly, but the heating power quickly decays to a safe level as the wire's resistance drops to R_{thresh}.

On subsequent cycles, the relay controller is able to track at least part of the sine wave. The small oscillations visible in the graph are caused by the relay controller hunting around the boundary between large and small position errors, so that its output oscillates between I_H and I_L. The frequency of oscillation is above the cut-off frequency of the voltage and current signal filters, so it appears greatly attenuated in the graph.

5 Conclusion and Future Work

This paper has described a method for rapid electrical heating of SMA elements without risk of overheating. It involves measuring the electrical resistance of an SMA element, calculating a safe upper limit to the heating current as a function of the measured resistance and element-specific resistance and current parameters, and ensuring that the actual heating current never exceeds the calculated upper limit. It improves upon the conceptually-similar method of Kuribayashi by dispensing with the need for special sensors (miniature thermocouples) co-located with the SMA elements; and it is the subject of a provisional patent application in Australia [2].

To demonstrate the effectiveness of this new method, it was incorporated into a motion control system based on Grant's two-stage relay controller, and applied to the task of controlling a pantograph linkage actuated by two antagonistic pairs of Flexinol (tm) wires. Experimental results for the tracking response of a single antagonistic pair under no-load conditions show that the rapid heating method doubles the peak actuator speed, compared with heating via the data-sheet value for safe heating current. This improvement is obtained purely by faster heating—the cooling regime does not change.

Although we have been successful in improving the speed of SMA-based actuators, the tracking accuracy is low (because of large limit cycles), and we suspect that the control system is over-stressing the SMA wires. (We have worn out several so far). We therefore intend to investigate alternatives to the two-stage relay controller, and to conduct fatigue tests on the wires. We are currently designing a new test rig that will include load cells for measuring the tension on each wire. We intend to investigate the use of load-cell data to implement force control systems and more accurate motion control systems.

References

1. Dynalloy, Inc., "Flexinol (tm) Technical Data," `http://www.dynalloy.com/-TechnicalData.html`, accessed May 2004.
2. R. Featherstone & Y. H. Teh, "A Shape Memory Alloy Actuator," Australian Provisional Patent Application no. 2004900618, 9^{th} Feb. 2004.
3. D. Grant, "Accurate and Rapid Control of Shape Memory Alloy Actuators," Ph.D. Thesis, Centre for Intelligent Machines, McGill University, TR-CIM-99-11, 1999.
4. S. Hirose, K. Ikuta & Y. Umetani, "A New Design Method of Servo-actuators Based on the Shape Memory Effect," in A. Morecki, G. Bianchi & K. Kędzior (Eds), *Theory and Practice of Robots and Manipulators*, MIT Press, Cambridge, MA, 1985, pp. 339–349.
5. K. Ikuta, M. Tsukamoto & S. Hirose, "Shape Memory Alloy Servo Actuator System with Electric Resistance Feedback and Application for Active Endoscope," *Proc. IEEE Int. Conf. Robotics & Automation*, Philadelphia, PA, pp. 427–430, 1988.
6. K. Kuribayashi, "Improvement of the Response of an SMA Actuator Using a Temperature Sensor," *Int. J. Robotics Research*, vol. 10, no. 1, pp. 13–20, 1991.
7. M. J. Mosley & C. Mavroidis, "Experimental Nonlinear Dynamics of a Shape Memory Alloy Wire Bundle Actuator," *Trans. ASME, J. Dynamic Systems, Measurement & Control*, vol. 123, no. 1, pp. 103–112, 2001.
8. D. Reynaerts & H. Van Brussel, "Design Aspects of Shape Memory Actuators," *Mechatronics*, vol. 8, pp. 635–656, 1998.
9. N. Troisfontaine, Ph. Bidaud & P. Dario, "Control Experiments on Two SMA Based Micro-actuators," in A. Casals & A. T. de Almeida (Eds), *Experimental Robotics V*, Springer, London, 1998, pp. 490–499.

Adaptive Friction Compensation Using a Velocity Observer

Qing Hua Xia[1], Ser Yong Lim[2], Marcelo H Ang Jr[1], and Tao Ming Lim[2]

[1] Mechanical Engineering Department, National University of Singapore
[2] Singapore Institute of Manufacturing Technology

Abstract. In this paper, adaptive friction compensation and identification issues are investigated. The friction adaptation law is formulated by utilizing both observed and desired velocity information. The overall adaptive observer-controller system is semi-global asymptotically stable. Experimental results verify the effectiveness of the the proposed adaptive observer-controller.

1 Introduction and State-of-Art

Friction is always present in robot servo-mechanisms and causes tracking lags, steady state errors and undesired stick-slip motion. In most industrial robots, motors torques are transformed through gears to links. The dynamical behavior of robots is significantly affected by gears, and a high amount of torque is needed to overcome friction.

In general, tracking errors due to friction cannot always be compensated by high gains. Therefore, friction identification and compensation issues have been investigated extensively. For example, Tomei developed a robust adaptive controller for friction compensation in the presence of bounded disturbances and/or modeling uncertainties [1]; Zhu et al. presented an adaptive controller that consider both static and dynamic friction effects [2]; and Azenha et al. investigated a variable structure control scheme for the robot with nonlinear friction and dynamic backlash [3]. By utilizing the velocity observer structure proposed in [4], we developed an adaptive controller that combined an adaptive friction control law with an observer-controller to achieve higher tracking accuracy [5]. Based on the adaptive controller structure presented in [5], we designed a new adaptive controller for friction identification and compensation. The friction adaptation algorithm is constructed by using both observed and desired velocity information. Experimental results using PUMA 560 indicated that the proposed observer-controller could achieve higher tracking accuracy than the one without friction compensation.

2 Open-Chain Robot Dynamic Model

The open-chain n degree-of-freedom robot dynamic model in joint space can be expressed as:

$$A(q)\ddot{q} + B(q,\dot{q})\dot{q} + g(q) + \tau_f = \Gamma \tag{1}$$

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 77–86, 2006.

where $\varGamma$ is the $n \times 1$ vector of joint torques, q is the $n \times 1$ vector of joint positions, $A(q)$ is the $n \times n$ inertial matrix, $B(q, \dot{q})$ is the $n \times n$ centrifugal and Coriolis matrix expressed in joint space, $g(q)$ is the $n \times 1$ vector of gravitational torques, τ_f is the $n \times 1$ vector of friction torques.

In our proposed adaptive controller, we use the following friction model [6]:

$$\tau_f = \tau_{vis}\dot{q} + \left[\tau_{cou} + \tau_{sti}\exp(-\tau_{dec}\dot{q}^2)\right]\mathrm{sgn}(\dot{q}) \tag{2}$$

where τ_{vis} denotes the diagonal coefficient matrix of viscous friction; τ_{cou} denotes the Coulomb friction-related diagonal coefficient matrix; τ_{sti} denotes the static friction-related diagonal coefficient matrix; τ_{dec} is a positive diagonal coefficient matrix corresponding to Stribeck effect; and the signum function sgn(.) is defined as:

$$sgn(\dot{q}) = \begin{cases} +1, & \dot{q} > 0 \\ 0, & \dot{q} = 0 \\ -1, & \dot{q} < 0 \end{cases} \tag{3}$$

The end-effector equation of motion in operational space can be expressed as [7]:

$$\varLambda(x)\ddot{x} + \varPsi(x, \dot{x})\dot{x} + p(x) + f = F \tag{4}$$

where F is the $n \times 1$ operational forces vector, x is the $n \times 1$ vector describing the position and orientation of the end-effector, $\varLambda(x)$ is the $n \times n$ kinetic energy matrix, $\varPsi(x, \dot{x})$ is the $n \times n$ centrifugal and Coriolis matrix expressed in operational space, $p(x)$ is the $n \times 1$ vector of gravitational forces, and f is the $n \times 1$ friction vector expressed in operational space.

For the proposed observer-controller stability analysis, the following properties of the robot dynamic model need to be used:

Property 1-The $n \times n$ kinetic energy matrix $\varLambda(x)$ defined in (4) satisfies the following inequality [8]:

$$m_1\|z\|^2 \leq z^T\varLambda(x)z \leq m_2\|z\|^2 = \|\varLambda(x)\|_{i2}\|z\|^2 \quad \forall z \in \Re^n \tag{5}$$

where m_1 and m_2 are known positive scalar constants. $\|.\|$ represents the standard Euclidean norm, and $\|.\|_{i2}$ represents the matrix induced two norm [9].

Property 2-In joint space dynamic model (1), the centrifugal and Coriolis matrix satisfies the following relationship [10]:

$$V_m(q, y)z = V_m(q, z)y \quad \forall y, z \in \Re^n$$

Property 3-In operational space dynamic model (4), the centrifugal and Coriolis matrix $\varPsi(x, \dot{x})$ satisfies the following relationships:

$$z^T\left[\frac{1}{2}\dot{\varLambda}(x) - \varPsi(x, \dot{x})\right]z = 0 \quad \forall z \in \Re^n \tag{6}$$

$$\Psi(x,y)z = \Psi(x,z)y \quad \forall y,z \in \Re^n \tag{7}$$

and

$$\|\Psi(x,\dot{x})\|_{i\infty} \leq \zeta_c \|\dot{x}\| \tag{8}$$

where ζ_c is a known positive scalar constant and $\|.\|_{i\infty}$ represents the matrix induced infinity norm.

3 Formulation of Adaptive Controller

Our proposed adaptive observer-controller consists of a model-based velocity observer, an operational space controller, plus friction adaptation law.

3.1 Formulation of Velocity Observer

In our velocity observer formulation, the following velocity observer is utilized to estimate the end-effector velocity:

$$\dot{\hat{x}} = y + k\tilde{x}, y(0) = -k\tilde{x}(0) \tag{9}$$

$$\dot{y} = \Lambda(x)^{-1}\left[F - \Psi(x,\dot{\hat{x}})\dot{\hat{x}} - p(x) - \hat{f} + k_i\tilde{x}\right] \tag{10}$$

where

$$\tilde{x} = x - \hat{x} \tag{11}$$

F is the force control input to the observer. $\hat{f}$ is the estimated friction term. k_i is a positive scalar constant to be decided. k is a positive scalar constant defined by:

$$k = \frac{1}{m_1}\left[\zeta_c\zeta_d + \zeta_c k_0 + \zeta_c k_s k_0 + k_s + 2k_n + \zeta_e\right] \tag{12}$$

where k_0, k_s and k_n being positive scalar control gains, ζ_c is declared in (8), ζ_e is defined in (16), and ζ_d is a known positive scalar constant defined by:

$$\|\dot{x}_d\| \leq \zeta_d \tag{13}$$

where $\dot{x}_d$ represents the desired end-effector velocity.

3.2 Formulation of Friction Adaptation Law

We assume that the friction term τ_f in (2) is uncoupled among the joints, so that

$$\tau_f = \text{vec}\{\tau_{f_i}(\dot{q}_i)\} \equiv \begin{bmatrix} \tau_{f_1}(\dot{q}_1) \\ \cdot \\ \cdot \\ \cdot \\ \tau_{f_n}(\dot{q}_n) \end{bmatrix} \tag{14}$$

with $\tau_{f_i}(.)$ known scalar functions that may be determined for any given arm. Here we have defined the vec{.} function for future use.

We assume that the viscous friction have the form:

$$\tau_{vis}\dot{q} = \text{vec}\{\tau_{vis_i}\dot{q}_i\} \tag{15}$$

with τ_{vis_i} constant coefficients. Then τ_{vis} =diag$\{\tau_{vis_i}\}$, a diagonal matrix with entries τ_{vis_i}.

The viscous friction term has the following property [5]:

$$\|\tau_{vis}\dot{q}\|\leq\zeta_e\|\dot{x}\| \tag{16}$$

Assume that the Coulomb friction has the form:

$$\tau_{cou}\text{sgn}(\dot{q}) = \text{vec}\{\tau_{cou_i}\text{sgn}(\dot{q}_i)\} \tag{17}$$

with τ_{cou_i} constant coefficients, and τ_{cou} =diag$\{\tau_{cou_i}\}$.

The friction term $\tau_{sti}\exp(-\tau_{dec}\dot{q}^2)\text{sgn}(\dot{q})$ in (2) is the combination of static friction and Stribeck effect, and we assume that it can be written in the following form [6]:

$$\tau_{sti}\exp(-\tau_{dec}\dot{q}^2)\text{sgn}(\dot{q}) = \text{vec}\{\tau_{sti_i}\exp(-\tau_{dec_i}\dot{q}_i^2)\text{sgn}(\dot{q}_i)\} \tag{18}$$

with τ_{sti_i} and τ_{dec_i} constant coefficients, and τ_{sti} =diag$\{\tau_{sti_i}\}$, τ_{dec} =diag$\{\tau_{dec_i}\}$.

The joint space robot friction model (2) can be written in the following linearity-in-the-parameters form:

$$\tau_f = W_j(\dot{q})\theta \tag{19}$$

where $W_j(\dot{q})$ is the $n \times 3n$ regression vector given by:

$$\begin{aligned} W_j(\dot{q}) &= \begin{bmatrix} w_{j1}(\dot{q}) \; w_{j2}(\dot{q}) \; w_{j3}(\dot{q}) \end{bmatrix} \\ w_{j1}(\dot{q}) &= \text{diag}(\dot{q}) \\ w_{j2}(\dot{q}) &= \text{diag}(\text{sgn}(\dot{q})) \\ w_{j3}(\dot{q}) &= \text{diag}(\text{sgn}(\dot{q})\exp(-\tau_{dec}\dot{q}^2)) \end{aligned} \tag{20}$$

and θ is the $3n \times 1$ vector of constant parameters defined as:

$$\theta = \begin{bmatrix} \text{vec}^T\{\tau_{vis_i}\} \; \text{vec}^T\{\tau_{cou_i}\} \; \text{vec}^T\{\tau_{sti_i}\} \end{bmatrix}^T \tag{21}$$

Here we assume that the coefficients τ_{vis}, τ_{cou} and τ_{sti} are unknown constants, but the Stribeck parameters τ_{dec} are assumed to be known.

The friction parameter estimate vector $\hat{\theta}$ is updated using the following adaptation algorithm:

$$\dot{\hat{\theta}} = -K_{ad} W_j(\dot{\hat{q}}, \dot{\hat{q}}_d)\dot{\tilde{q}} \tag{22}$$

where K_{ad} is a $3n \times 3n$ diagonal, positive-definite, adaptation gain matrix; the joint velocity observation error $\dot{\tilde{q}}$ is defined as:

$$\dot{\tilde{q}} = \dot{q} - \dot{\hat{q}} \tag{23}$$

In the non-singular region of a robot, the observed joint velocity $\dot{\hat{q}}$ can be obtained by:

$$\dot{\hat{q}} = J^{-1}\dot{\hat{x}} \tag{24}$$

where $\dot{\hat{x}}$ is calculated by (9), and $W_j(\dot{\hat{q}}, \dot{\hat{q}}_d)$ is the $n \times 3n$ regression vector given by:

$$\begin{aligned} W_j(\dot{\hat{q}}, \dot{\hat{q}}_d) &= \begin{bmatrix} \hat{w}_{j1}(\dot{\hat{q}}_d) \ \hat{w}_{j2}(\dot{\hat{q}}) \ \hat{w}_{j3}(\dot{\hat{q}}) \end{bmatrix} \\ \hat{w}_{j1}(\dot{\hat{q}}_d) &= \operatorname{diag}(\dot{\hat{q}}_d) \\ \hat{w}_{j2}(\dot{\hat{q}}) &= \operatorname{diag}(\operatorname{sgn}(\dot{\hat{q}})) \\ \hat{w}_{j3}(\dot{\hat{q}}) &= \operatorname{diag}(\operatorname{sgn}(\dot{\hat{q}}) \exp(-\tau_{dec}\dot{\hat{q}}^2)) \end{aligned} \tag{25}$$

The estimated joint frictions are obtained by:

$$\hat{\tau}_f = W_j(\dot{\hat{q}}, \dot{\hat{q}}_d) \tag{26}$$

From (26), the estimated frictions in operational space can be obtained by:

$$\hat{f} = J^{-T}\hat{\tau}_f \tag{27}$$

3.3 Formulation of Operational Space Controller

Utilizing the estimated velocity $\dot{\hat{x}}$ proposed in section 3.1, we can formulate the following controller:

$$F = (k_s + k_{nd})\eta_p + w_e - k_i\tilde{x} \tag{28}$$

where k_{nd} is a positive controller gain defined as:

$$k_{nd} = 2k_n + \zeta_c k_0 + (k_s m_2 + k m_2)^2 k_n \tag{29}$$

the $n \times 1$ observed filtered tracking error signal η_p is defined as:

$$\eta_p = \dot{x}_d + k_s e - \dot{\hat{x}} \tag{30}$$

and the $n \times 1$ auxiliary vector w_e is defined as:

$$w_e = \Lambda(x)[\ddot{x}_d + k_s(\dot{x}_d - \dot{\hat{x}})] + \Psi(x, \dot{\hat{x}})(\dot{x}_d + k_s e) + p(x) + \hat{f} \tag{31}$$

where the $n \times 1$ end-effector position and orientation tracking error e is defined as:

$$e = x_d - x \tag{32}$$

The generated F will be used by the velocity observer indicated by (10). And the corresponding torque commands for driving the robot can be obtained by:

$$\Gamma = J^T F$$

4 Overall System Stability Result

Theorem 1. *Under the assumption that the exact model of a robot except friction is known, if the observer-controller gains satisfy the following sufficient conditions :*

$$\begin{array}{l} k_s > 1/k_n + \eta_e \\ k_s > 2/k_n \\ k_0 > \|err(0)\| \end{array} \tag{33}$$

the closed-loop tracking error system is asymptotically stable as illustrated by:

$$\lim_{t\to\infty} e(t) = 0, \lim_{t\to\infty} \dot{e}(t) = 0 \tag{34}$$

$$\lim_{t\to\infty} \tilde{x}(t) = 0, \lim_{t\to\infty} \dot{\tilde{x}}(t) = 0 \tag{35}$$

$$\lim_{t\to\infty} \hat{\theta}(t) = \theta, \tag{36}$$

where

$$err = \begin{bmatrix} \eta_p^T & e^T & \dot{\tilde{x}}^T & \tilde{x}^T \end{bmatrix}^T \in \Re^{4n} \tag{37}$$

5 Overall System Stability Analysis

First, we define the following Lyapunov function.

$$V = V_0 + V_1 + V_2 \tag{38}$$

where

$$V_0 = \frac{1}{2}\dot{\tilde{x}}^T \Lambda(x)\dot{\tilde{x}} + \frac{1}{2}\tilde{x}^T k_i \tilde{x} + \frac{1}{2}\tilde{\theta}^T K_{ad}^{-1}\tilde{\theta} \tag{39}$$

where

$$\tilde{\theta} = \theta - \hat{\theta} \tag{40}$$

and

$$V_1 = \frac{1}{2} e^T e \tag{41}$$

$$V_2 = \frac{1}{2} \eta_p^T \Lambda(x) \eta_p \tag{42}$$

To form the bound of $\dot{V}_0$, first, take the time derivative of (9) and then substitute (10) into the resulting expression to yield:

$$\Lambda(x)\ddot{\hat{x}} + \Psi(x, \dot{\hat{x}})\dot{\hat{x}} + p(x) + \hat{f} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x} = F \tag{43}$$

Subtract (43) from (4) to yield the following closed-loop observer error system:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x, \dot{x})\dot{\tilde{x}} + \Psi(x, \dot{\hat{x}})\dot{\tilde{x}} + k\Lambda(x)\dot{\tilde{x}} + k_i\tilde{x} + f - \hat{f} = 0 \tag{44}$$

Differentiate V_0 along (44) to get:

$$\begin{aligned} \dot{V}_0 = \dot{\tilde{x}}^T &\left[-\Psi(x, \dot{x})\dot{\tilde{x}} - \Psi(x, \dot{\hat{x}})\dot{\tilde{x}} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x} \right] \\ &+ \frac{1}{2}\dot{\tilde{x}}^T \dot{\Lambda}(x)\dot{\tilde{x}} + \dot{\tilde{x}}^T k_i\tilde{x} - \dot{\tilde{x}}^T (f - \hat{f}) \end{aligned} \tag{45}$$

The upper bound for the time derivative of V_1 can be expressed as:

$$\dot{V}_1 \leq -k_s\|e\|^2 + \|e\|\|\eta_p\| + \|e\|\|\dot{\tilde{x}}\| \tag{46}$$

Differentiating V_2 with respect to time and utilizing (6) to get:

$$\begin{aligned} \dot{V}_2 = &-(k_s + k_{nd})\eta_p^T\eta_p - (k + k_s)\eta_p\Lambda(x)\dot{\tilde{x}} \\ &+ \eta_p^T\Psi(x, \dot{\tilde{x}})\eta_p \end{aligned} \tag{47}$$

Finally, the upper bound of $\dot{V}$ can be formed as:

$$\begin{aligned} \dot{V} \leq &-(k_s - \tfrac{2}{k_n})\left\|e\right\|^2 - (k_s - \tfrac{1}{k_n})\left\|\eta_p\right\|^2 - (k_s - \tfrac{1}{k_n} - \zeta_e)\left\|\dot{\tilde{x}}\right\|^2 \\ &- (k_o - \|err\|)\left(\zeta_c\left\|\dot{\tilde{x}}\right\|^2 + \zeta_c k_s\left\|\dot{\tilde{x}}\right\|^2 + \zeta_c\left\|\eta_p\right\|^2\right) \end{aligned} \tag{48}$$

From (48) we can see that, if $k_s > 1/k_n + \zeta_e$, $k_s > 2/k_n$, and $k_0 \geq ||err||$, we can get:

$$\dot{V} \leq 0 \tag{49}$$

From (49), we can get the conclusion that, friction coefficients estimation error $\tilde{\theta}$, position tracking errors e, position estimation errors $\widetilde{x}$, velocity estimation errors $\dot{\widetilde{x}}$, observed filtered tracking error signal η_p of the observer-controller are all asymptotically stable, i.e. $\lim_{t \to \infty} \hat{\theta} = \theta$, $\lim_{t \to \infty} e(t) = 0$, $\lim_{t \to \infty} \widetilde{x}(t) = 0$, $\lim_{t \to \infty} \dot{\widetilde{x}}(t) = 0$ and $\lim_{t \to \infty} \eta_p(t) = 0$. Furthermore, the end-effector velocity tracking error is also asymptotically stable, i.e. $\lim_{t \to \infty} \dot{e}(t) = 0$. In fact, after adding and subtracting $\dot{x}$ to the right-hand side of (30) and rearranging the terms, we can formulate the following inequality:

$$\left\|\dot{e}\right\| = \left\|\dot{x_d} - \dot{x}\right\| \leq \left\|\eta_p\right\| + k_s \left\|e\right\| + \left\|\dot{x}\right\| \tag{50}$$

Since each of the terms on the right-hand side of the above equation is asymptotically stable, it is easy to see that $\left\|\dot{e}\right\|$ is also asymptotically stable. This yields the result indicated by *Theorem 1.*

6 Experimental Results

The experiments were performed using PUMA 560 robot. The defined trajectory is to move the end-effector in XYZ direction with the desired position trajectory indicated by (51), while maintaining the initial end-effector orientation constant all the time.

$$\begin{aligned}
p_{x_d} &= p_{x_0} + 50.0 \sin(2\pi f t) \left(1 - e^{-0.05t^3}\right) \text{mm} \\
p_{y_d} &= p_{y_0} + 50.0 \cos(2\pi f t) \left(1 - e^{-0.05t^3}\right) \text{mm} \\
p_{z_d} &= p_{z_0} + 50.0 \cos(2\pi f t) \left(1 - e^{-0.05t^3}\right) \text{mm} \\
f &= 0.2\text{Hz}
\end{aligned} \tag{51}$$

where p_{x_0}, p_{y_0} and p_{z_0} are the initial positions of the robot. The controller gains were selected as diagonal gains matrices as following:

$$\begin{aligned}
k_{nd} &= \text{diag}\{120, 120, 120, 35, 35, 35\} \\
k &= \text{diag}\{108, 108, 108, 32, 32, 32\} \\
k_s &= \text{diag}\{97, 97, 97, 30, 30, 30\} \\
k_i &= \text{diag}\{2000, 2000, 2000, 3000, 3000, 3000\}
\end{aligned} \tag{52}$$

We set all the diagonal terms of the 18×18 friction adaptation gains K_{ad} to be 500, all the diagonal terms of τ_{dec} to be 1, and all the initial estimated friction coefficients to be zeros. The experimental result is shown in Fig. 1, and Table 1 shows the tracking errors after the robot ran for about two minutes. J_i stands for Joint i, and e_x, e_y, and e_z are the position tracking errors along X, Y, and Z axis, and e_{ϕ_x}, e_{ϕ_y}, and e_{ϕ_z} are the orientation tracking errors about X, Y, and Z axis, respectively.

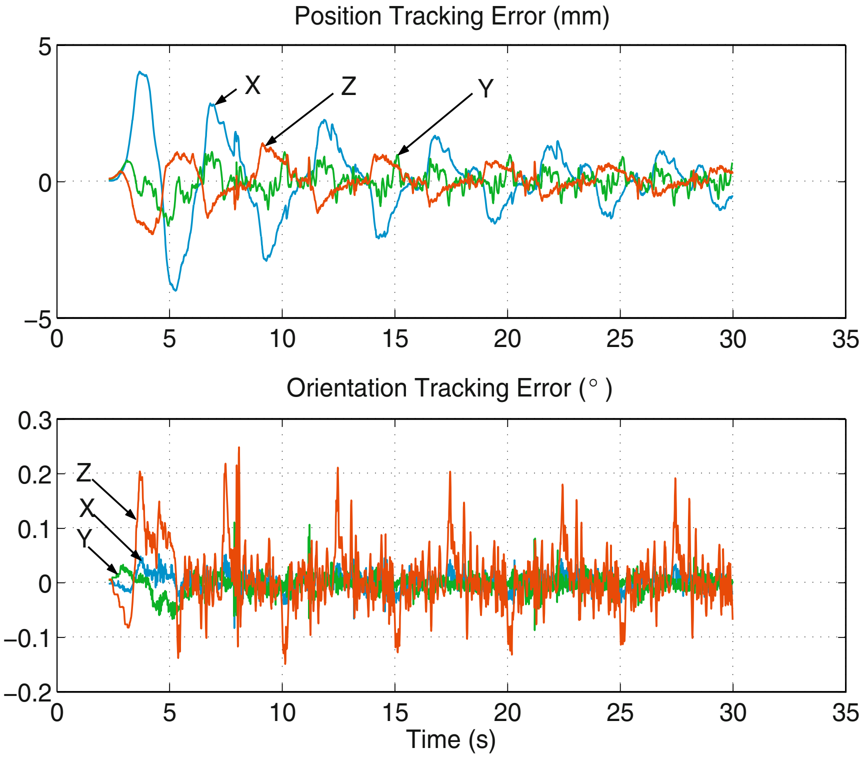


Fig. 1. Tracking errors with adaptive friction compensation

Table 1. Maximum tracking errors with adaptive friction compensation

e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
0.31mm	0.49mm	0.29mm	0.05^o	0.04^o	0.07^o

Table 2. Maximum tracking errors without friction compensation

e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
1.84mm	1.58mm	1.30mm	0.11^o	0.09^o	0.19^o

Using the same controllers gains listed in (52), the experimental result without friction compensation is shown in Tabel 2 The results indicate that the tracking errors of the controllers with adaptive friction compensation is about 2 to 6 times smaller than the controller without friction compensation, which verify the effectiveness of the proposed adaptive controller.

7 Conclusion and Research Perspective

We proposed an operational space observer-controller with adaptive friction compensation capability. The friction adaptation law is formulated by utilizing both observed and desired velocity information. Experimental results indicate that, the proposed adaptive controller is able to achieve higher tracking accuracy than the one without friction compensation, which verifies the effectiveness of the control algorithm.

References

1. P. Tomei, "Robust adaptive friction compensation for tracking control of robot manipulators," *IEEE Trans. Automat. Contr.*, vol. 45, no. 11, pp. 2164–2169, 2000.
2. Y. L. Zhu and P. R. Pagilla, "Static and dynamic friction compensation in trajectory tracking control of robots," in *Proc. IEEE Int. Conf. Rob. Autom.*, vol. 3, Washington, DC, USA, May 2002, pp. 2644–2649.
3. A. Azenha and J. A. T. Machado, "Variable structure control of robots with nonlinear friction and backlash at the joints," in *Proc. IEEE Int. Conf. Rob. Autom.*, vol. 1, Minneapolis, MN, USA, Apr. 1996, pp. 366–371.
4. Q. H. Xia, S. Y. Lim, Marcelo H. Ang Jr, and T. M. Lim, "Implementation of an output feedback controller in operational space," in *Proc. IEEE/RSJ Int. Conf. Intelli. Robots and Syst.*, Las Vegas, NV, USA, Oct. 2003, pp. 2761–2766.
5. ——, "Adaptive joint friction compensation using a model-based operational space velocity observer," in *Proc. IEEE Int. Conf. Rob. Autom.*, New Orleans, USA, June/July 2004, pp. 3081–3086.
6. M. S. de Queiroz, D. M. Dawson, and F. M. Zhang, *Lyapunov-Based Control of Mechanical Systems.* Boston: Birkhauser, 2000.
7. O. Khatib, "A unified approach to motion and force control of robot manipulators: the operational space formulation," *IEEE Trans. Robot. Automat.*, vol. 3, no. 1, pp. 43–53, 1987.
8. M. Spong and M. Vidyasagar, *Robot Dynamics and Control.* NY: Wiley, 1989.
9. M. Vidyasagar, *Nonlinear Systems Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1978.
10. S. Nicosia and P. Tomei, "Robot control by using only joint position measurements," *IEEE Trans. Automat. Contr.*, vol. 35, no. 9, pp. 1058–1061, 1990.

Wave-shape pattern control of electroactive polymer gel robots

Mihoko Otake, Yoshihiko Nakamura, Masayuki Inaba, and Hirochika Inoue

Department of Mechano-Informatics, Graduate School of Information Science and
Technology, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
E-mail: otake@ynl.t.u-tokyo.ac.jp

Abstract. This paper proposes wave-shape pattern control method for whole-body deformable robots containing electroactive polymers. Mechanisms consisting of a typical electroactive polymer gel containing poly 2 -acrylamido -2- methylpropane sulfonic acid (PAMPS), named 'gel robots', have been designed, developed, and controlled experimentally. We faced a common problem called, the degrees-of-freedom problem, namely controlling many points on the gel surface with a small number of inputs. In order to solve the problem, we once reduced the number of inputs to one, and discovered that wave-shape pattern evolves for the beam-shaped gel even in a constant uniform electric field. This method makes use of the pattern formation. Wave-shaped gels with varying curvature are obtained by switching the polarity of a spatially uniform electric field. The method is verified through experiments which are carefully designed based on numerical simulations.

1 Introduction

Electroactive polymers [1] change their shapes in the electric fields. The fact motivated us to design the whole-body deformable robots. The purpose of this study is to establish methods for deriving a variety of shapes and motions of deformable robots whose bodies are made of active materials. Shape control of such mechanisms is difficult, which is commonly called the degrees-of-freedom problem [2], namely a problem of controlling many points of continuum with a small number of inputs. In order to explore the high dimensional configuration space of elastic objects [3], probabilistic roadmaps was applied.

Osada et al. developed a surfactant-driven ionic polymer gel which is made of poly 2-acrylamido-2-methylpropane sulfonic acid (PAMPS), and well known for its large deformation[4]. Although the gel reversibly bends and stretches by alternating the polarity of the electric fields, the spatiotemporal pattern formation has not been studied in depth. We have been developing deformable machines utilizing the gel, hereafter called "gel robots", and proposed methods for exploiting a variety of shapes and motions of the gel[5–8]. The methods were applied and succeeded in generating variety of motions such as beam-shaped gels curling around an object and starfish-shaped gels turning over[9]. The pattern formation process of the gel was analyzed refering to the living things that also have deformable bodies. They accumulate the temporal sequence of input and build their bodies in the developmental process called morphogenesis. Uniform structures are converted to complex organs in a uniform

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 87–96, 2006.

gradient field. Learning from morphogenesis, we applied a uniform gradient electric field to a straight beam of gel, which exhibited wave-shape pattern formation[10]. The experimental results suggest that complex shapes can be generated by simple electric fields.

In this paper, a method for generating a variety of wave-shapes is proposed making use of pattern formation theory of electroactive polymer gels. The wave-shape pattern formation experiments are described in the following section. Then, the wave-shape pattern control method is derived and verified through control experiments.

2 Wave-shape pattern formation of electroactive polymer gel

2.1 Experimental methods

The gel was prepared by radical copolymerization at 323K for 48 hours. The total monomer concentration in N, N-dimethylformamide was kept at 3.0M in the presence of 0.01M N,N'- methylenebisacrylamide (MBAA) as a cross-linking agent and 0.01M α, α' - azobis (isobutyronitrile) (AIBN) as an initiator. Monomers were 2 - acrylamido -2- methylpropanesulfonic acid (AMPS), n-stearyl acrylate (SA), and acrylic acid (AA) with the composition (AMPS: SA: AA) = (20: 5: 75). After the polymerizations, the gel was immersed in a large amount of pure water to remove un-reacted reagents until it reached an equilibrium state. The degree of swelling of the gel was 20. This number is determined as a weight ratio of the water-swollen gel to its dry state, which characterizes the gel. In order to apply the electric field, the gel was immersed in a dilute solution of 0.01M lauryl pyridinium chloride containing 0.03M sodium sulphate. All experiments were carried out at a room temperature of 25 °C.

The experimental setup included a pair of parallel platinum plate electrodes of 25 [mm] wide and 40 [mm] long each, which were horizontally placed with 40 [mm] vertical spacing between them. A beam-shaped gel of 4 [mm] wide, 21 [mm] long, and 1[mm] thick was also horizontally placed inbetween with one end fixed for 5 [mm] and the other end free. The vertical section of the experimental setup is seen in Fig. 1. The two electrodes and the gel were immersed in the solution. A uniform electric field was applied by the electrodes. The current density was kept constant by a galvanostat at 0.15 [mA/mm^2] for 600 [s]. The deformation of the gel was monitored and recorded by a video microscope. The fixture size was set small so as not to disturb the electric field. As shown in Fig.1, the x-axis was chosen as the horizontal line going through the fixed end of the gel, while the y-axis was the vertical one also going through the fixed end. The electrodes were placed at y=$\pm$20 [mm]. Let ϕ be an angle between the tangential line of the gel at the free end and the x-axis.

2.2 Experimental results

With the constant electric field, the gel showed waving motion and a wave shape was eventually generated after a while. First the gel bent toward the anode side. When ϕ

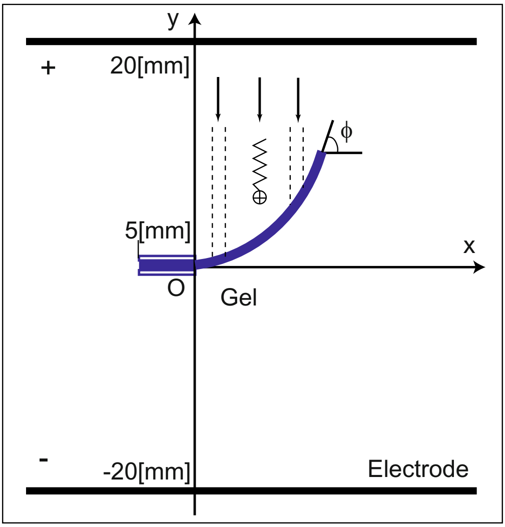


Fig. 1. Deformation process of a surfactant driven ionic polymer gel illustrated with experimental setup.

went over $\pi/2$, a portion of gel near the free end started to bend in the other direction. The deformation of root portion remained same. Again, when ϕ went under $\pi/2$, a smaller portion of the gel near the free end started to bend the first direction. Likewise the free end of the gel showed an oscillation. Fig. 2 shows the angle of the tip of the gel ϕ during motion. In the uniform electric field, the beam-shaped gel bent toward the anode side. The tip of the gel showed unclockwise motion in the x-y plane from the initial point (16, 0) [mm]. Angle ϕ increased from 0 and reached the local maximum of ϕ_1=2.11[rad] at $t_1 = 50$[s]. It didn't stop when the tip became vertical to the electrode and parallel to the electric field. Then the direction of the movement of the tip reversed. The angle decreased to the local minimum of $\phi_2 = 1.02$[rad] at $t_2 = 210$[s]. The direction of motion of the tip again reversed to increase ϕ. The angle of the tip became the local maximum of $\phi_3 = 1.80$[rad] at $t_3 = 420$[s], after that it gradually decreased in the course of experiment. The transitional shapes of the gel are shown in Fig. 3. Fig. 3(a) represents the shape of the gel at t_1 when the direction of the tip reversed for the first time. The second extremum of the gel near the tip appeared after t_1. The second extremum grew until the direction of the tip movement reversed in t_2 whose shape are shown in Fig. 3(b). After that, the third extremum appeared. When the tip of the gel reversed at t_3, the gel had three extrema (Fig. 3(c)). The maximum curvature decreased as the numbers of extrema increased. After the third reversing point of the tip, it became difficult to count the numbers of extrema. This is because the curvature became small and exceeded the maximum accuracy of measurement. The whole shape of the gel became mostly vertical to the electrode. In summary, wave-shape pattern formation of the gel observed. The waving rhythm generation was accompanied it.

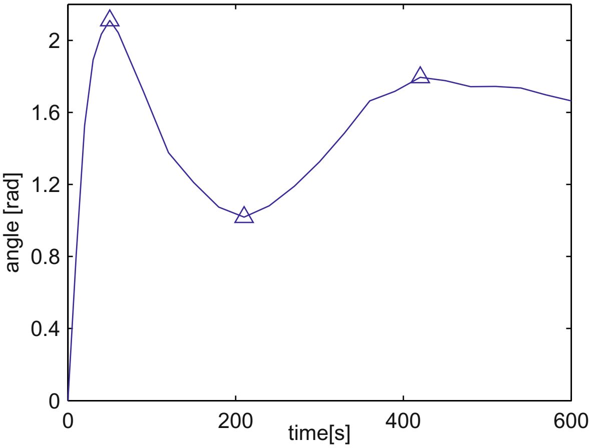


Fig. 2. The angle of the tip of the beam-shaped gel ϕ in a spatio-temporally uniform electric field.

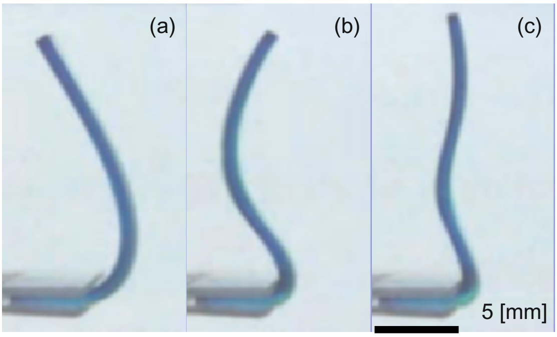


Fig. 3. Deformation of the beam-shaped gel in a spatio-temporally uniform electric field.

2.3 Mechanism of pattern formation

The nonlinearity of the electroactive polymers is the key to understand the mechanism. An ionic polymer gel in an electric field deforms through penetration of the surfactant solution [11]. This process is characterized by the following three steps.

1. Migration of surfactant molecules into the gel driven by the electric field
2. Adsorption of surfactant molecules to the polymers
3. Gel deformation caused by adsorption of surfactant molecules

The state of the gel is characterized by the distribution of adsorbed molecules, which determines its overall shape. Adsorption state transition of the gel is approximated in the following local nonlinear differential equation[10]:

$$\frac{\mathrm{d}\alpha}{\mathrm{dt}} = v_a - v_d = -a\boldsymbol{i} \cdot \boldsymbol{n} - d\alpha. \tag{1}$$

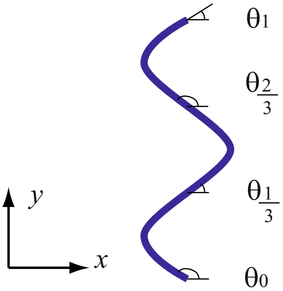


Fig. 4. Definition of the objective shapes with tangential angles in the case of three-half-waved shape

where α is the adsorption rate defined as the molar ratio of bound surfactants to the local sulfonates group of the polymer chains inside the gel; $\boldsymbol{i}$ is the current density vector on the gel surface; $\boldsymbol{n}$ is the normal vector of the gel surface; a and d are association and dissociation constants. The equation shows that the effect of an electric field to a gel is determined by the geometry of the equi-potential surface and the gel surface.

Since the electrodes were placed parallel to the x-axis, the current density vector is written by:

$$\boldsymbol{i} = (i_x, i_y) = i_c(0, -1), \tag{2}$$

with the constant $i_c = 0.15[\mathrm{mA/mm^2}]$. If we take the tangent angle on the gel surface for θ, the normal vector on the surface becomes as follows:

$$\boldsymbol{n} = (n_x, n_y) = (-\sin\theta, \cos\theta). \tag{3}$$

We can obtain the specific expression of equation (1) by substituting equations (2) and (3) into it,

$$\frac{\mathrm{d}\alpha}{\mathrm{dt}} = a i_c \cos\theta - d\alpha. \tag{4}$$

The second term of the right hand side of equation (4) is damping, which originates from the dissociation term in equation (1). If the first term of the right hand side of equation (4) is constant, the answer of the equation is a simple exponential function, which converses to zero. Inclusion of cosine function to the first-order partial differential equation of equation (4) causes rich pattern development. Adsorption reaction causes the mechanical deformation. The deformation determines the subsequent reaction. This interaction of deformation and reaction brings out the wave-shape pattern formation.

3 Wave-shape pattern control of electroactive polymer gel

3.1 Numerical simulation for experiments

Wave-shaped gels with varying curvature were obtained by switching the polarity of a spatially uniform electric field. The period for reversing the polarity was explored through numerical simulation. The polarity of one of the electrodes was either anodic(0) or cathodic(1). A control sequence is described with a time interval and its sequence. A time interval of 10 second was initially selected and its sequence of eight intervals were enumerated from (00000000) to (11111111). Other intervals were also considered every 10 seconds from 20 to 120 seconds. We determined (00001111) with 120 seconds time interval as the best input sequence, which generated three-half-waved shape with large curvature. The performance function was defined with the tangential angle of the gel for determinating input sequence. That of the three-half-waved shape f_3 is described by

$$f_3 = (\theta_{\frac{2}{3}} - \theta_1) + (\theta_{\frac{2}{3}} - _{\frac{1}{3}}) + (\theta_0 - _{\frac{1}{3}}) = \theta_0 - 2\theta_{\frac{1}{3}} + 2\theta_{\frac{2}{3}} - \theta_1, \tag{5}$$

where the suffix of the orientation θ is the normalized length from the root to the arbitrary point. The orientation of the root is θ_0, that of the tip is θ_1, that of the center is $\theta_{\frac{1}{2}}$, that of one third from the root is $\theta_{\frac{1}{3}}$, that of two thirds from the root is $\theta_{\frac{2}{3}}$. Performance function of the shape with x half-waves is noted as f_x. The objective shape with tangential angles is illustrated in Fig. 4. Simulated forms of the gel at every time interval are shown in Fig. 5.

Experiments were conducted. Experimental setup was the same as that of wave-shape pattern formation. The current density was kept constant by a galvanostat at 0.10 [mA/mm^2]. The polarity of the electric field was reversed from anodic(0) to cathodic(1) when the tangential angle at the tip of the gel reached the same values as that of the simulation.

3.2 Experimental results

The experimental snapshots corresponding simulation show the initial (1), transitional (2-8), and final (9) forms of the gel(Fig. 6). Let θ be an angle between the tangential line of the gel at the free end and one of the fixing ends. First, the gel bent toward the anode side (2). A portion of gel near the free end started to bend in the other direction when $shape$ went over π/2 (3). The deformation of root portion remained same. Again, when θ went under π/2, a smaller portion of the gel near the free end started to bend the first direction (4). The polarity of the electric field was reversed at (5). Finally, the gel reached the desired shape (9).

3.3 Reachable wave-shapes

Different shapes were generated through alternating the performance functions. The performance functions of one-half-waved shape (f_1) and two-half-waved shape (f_2) are defined as same as that of the three-half-waved shape.

$$f_1 = \theta_0 - \theta_1 \tag{6}$$

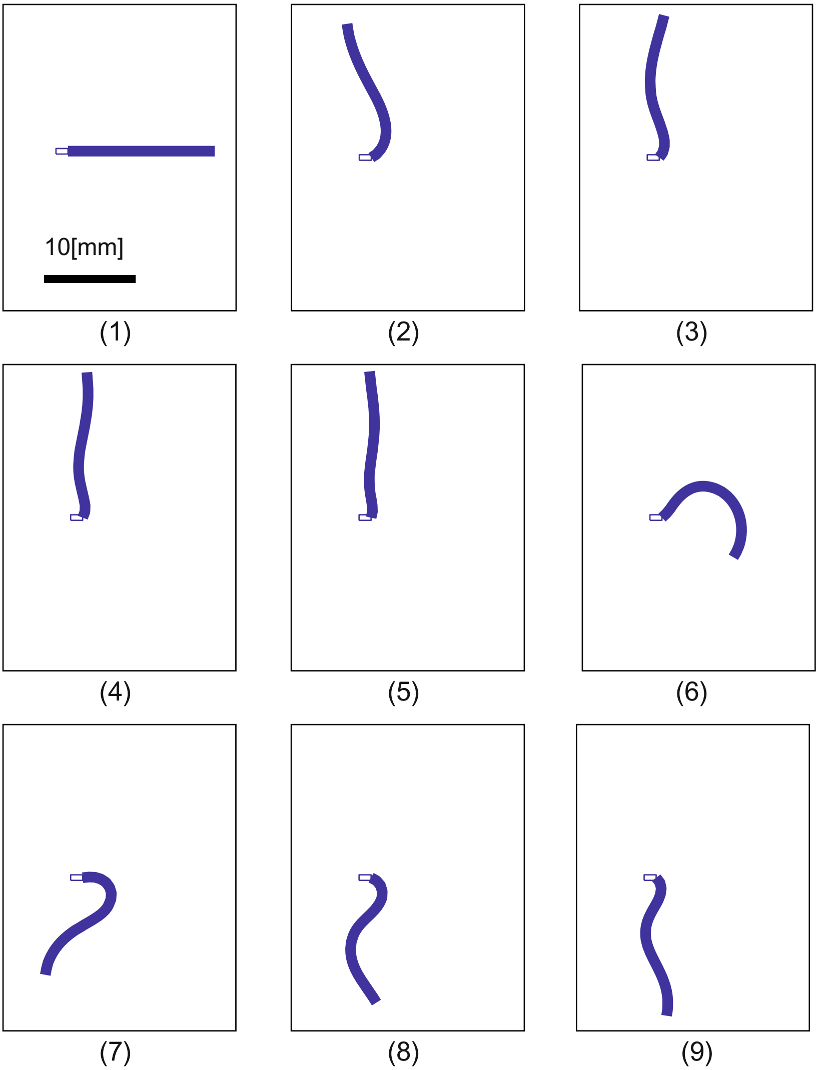


Fig. 5. Simulated process of the gel which deforms into three-half-waved shape

$$f_2 = (\theta_{\frac{1}{2}} - \theta_0) + (\theta_{\frac{1}{2}} - \theta_1) = 2\theta_{\frac{1}{2}} - \theta_0 - \theta_1 \qquad (7)$$

The input sequesnces which maximize the performance functions were obtained. One-half-waved shape was generated by (1100011) with 90 secondes time interval. Likewise, two-half-waved shape was generated by (11111000) with 90 seconds time interval. Simulated and obtained shapes are shown in Fig. 7.

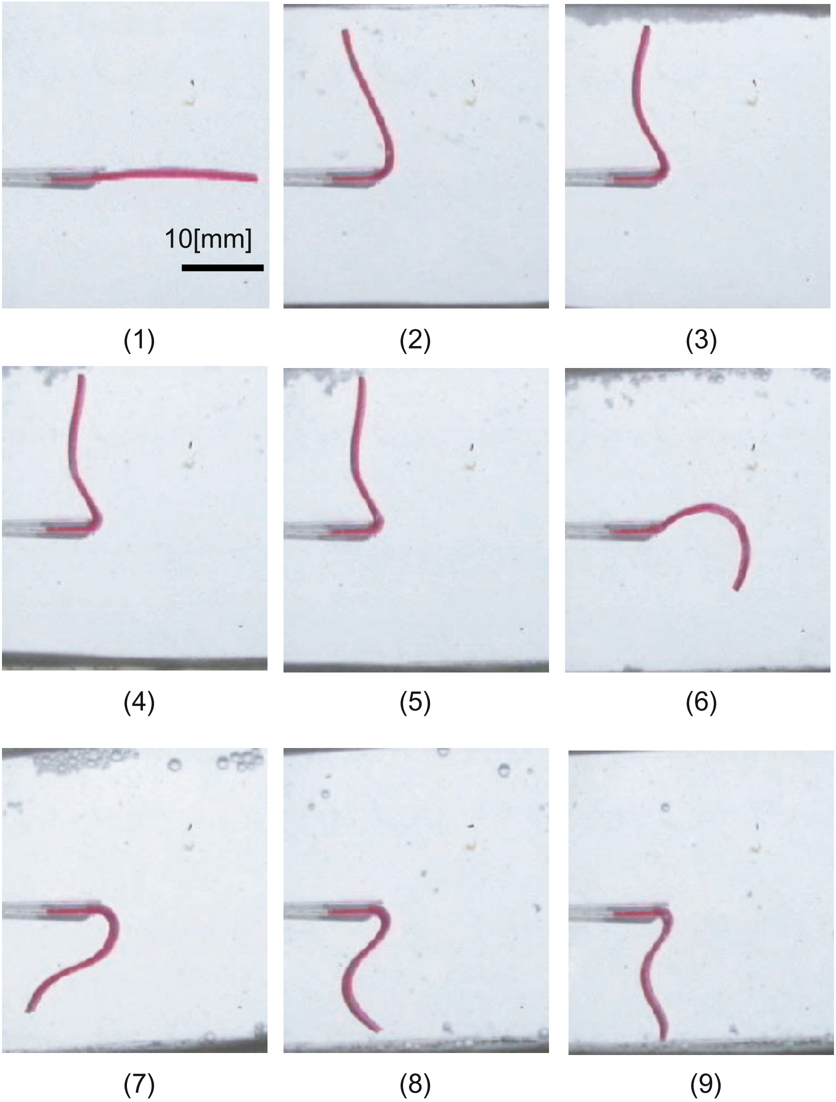


Fig. 6. Experimental results of the gel which deforms into three-half-waved shape

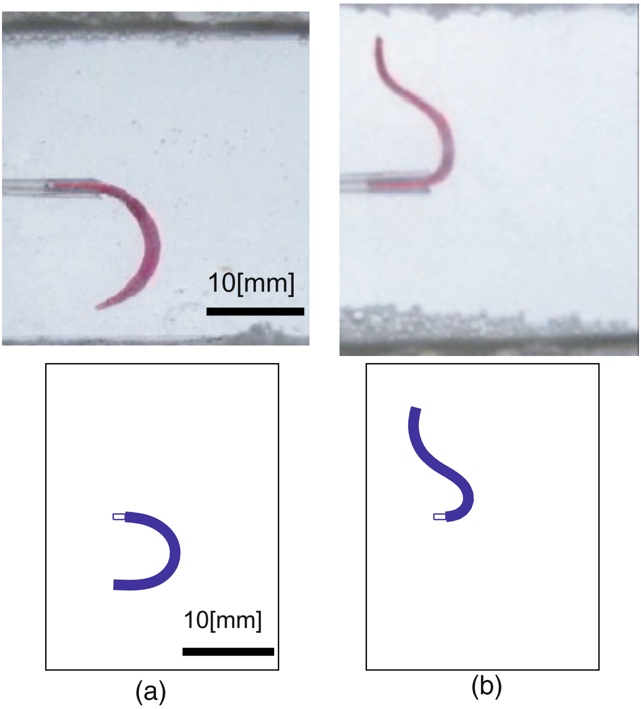


Fig. 7. Simulated and experimental results of the gel which deformed into (a)one-half-waved shape and (b)two-half-waved shape

4 Conclusion

The conclusions of this paper are summarized as follows:

1. Wave-shape pattern formation of a surfactant driven ionic polymer gel in a uniform electric field was described. A beam of ionic gel in a uniform electric field develops wave form through penetration of the surfactant solution.
2. The wave-shape pattern control method was proposed. Wave-shaped gels with varying curvature were obtained by switching the polarity of a spatially uniform electric field: 1) The period for reversing the polarity was explored through numerical simulation; 2) The performance function was defined with the tangential angle of the gel for determinating input sequence; 3) The period for reversing the polarity of the electric field was regulated when the numerical simulations are applied to experiments.
3. The method was experimentally verified. Originally straight-shaped gel deformed into the shapes containing one, two and three half-waves.

Further investigation is required in order to make clear the reachable shapes by this method. Polarity switching of spatially varying electric fields would generate more complex shapes. From the mechanism of deformation and experimental results, we can consider gel as an integrator of the series of input. Molecules are adsorbed on the surface of the gels, which deform the shape of the gels. The deformed surface forms a reaction field in the next step. In this way, series of input by a time varying electric field are accumulated. Hysteretic property of the materials helps to solve the degrees-of-freedom problem of deformable robots containing electroactive polymers.

Acknowledgement

The authors would like to thank Prof. Y. Kakazu, Prof. Y. Osada with Hokkaido University, who provided us valuable suggestions and discussions.

References

1. Y. E. Bar-Cohen, *Electroactive Polymer (EAP) Actuators as Artificial Muscles - Reality, Potential and Challenges*. SPIE Press, Bellingham, WA, 2001.
2. N. Bernstein, *The Co-ordination and Regulation of Movments*. Pergamon Press, 1967.
3. L. Kavraki, E. Lamiraux, and C. Holleman, "Towards planning for elastic objects," in *Robotics: The Algorithmic Perspective, P. Agarwal, L. E. Kavraki, and M. Mason (Eds.)*. AK Peters, Natick, MA, 1998, pp. 313–326.
4. Y. Osada, H. Okuzaki, and H. Hori, "A polymer gel with electrically driven motility," *Nature*, vol. 355, pp. 242–244, 1992.
5. M. Otake, M. Inaba, and H. Inoue, "Kinematics of Gel Robots made of Electro-Active Polymer PAMPS Gel," in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, 2000, pp. 488–493.
6. M. Otake, Y. Kagami, M. Inaba, and H. Inoue, "Dynamics of Gel Robots made of Electroactive Polymer Gel," in *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, 2001, pp. 1457–1462.
7. M. Otake, Y. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Inverse Kinematics of Gel Robots made of Electroactive Polymer Gel," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, 2002, pp. 3224–3229.
8. M. Otake, Y. Kagami, Y. Kuniyoshi, M. Inaba and H. Inoue, "Inverse Dynamics of Gel Robots made of Electroactive Polymer Gel," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, 2003, pp. 2299–2304.
9. M. Otake, Y. Kagami, M. Inaba, and H. Inoue, "Motion design of a starfish-shaped gel robot made of electro-active polymer gel," *Robotics and Autonomous Systems*, vol. 40, pp. 185–191, 2002.
10. M. Otake, Y. Nakamura, and H. Inoue, "Pattern Formation Theory for Electroactive Polymer Gel Robots," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 2004, pp. 2782–2787.
11. H. Okuzaki and Y. Osada, "Effects of hydrophobic interaction on the cooperative binding of a surfactant to a polymer network," *Macromolecules*, vol. 27, pp. 502–506, 1994.

IV. Localization and Map Building

Navigation of a mobile robot in an unknown environment is a challenge that has been studied extensively in the past. The ability of a mobile robot to localize itself with respect to the environment is dependent on its external sensor and the availability of the information on its surrounding environment – or a map. A map could be geometric or topological in nature as long as it represents the necessary features in the environment that enable the robot to localize itself. The absence of a map requires the robot to also create a model of its surrounding environment. This resulting chicken-and-egg problem has presented a lot of research opportunities. A single mobile robot in an unknown indoor environment is commonly thought of as the starting level of the problem. The level of difficulty increases with outdoor environment and multi-robot problems.

The first article by Tapus and co-workers presents an indoor localization and map building approach, which utilizes the fingerprint concept with uncertainty modeling. Fingerprint concept is one where the robot obtains all the features in the circle surrounding it at every point – as opposed to having a directional view. In the second article, Williams and Mahon present the results of Terrain Aided Navigation and Simultaneous Localization and Map-building problem in the Great Barrier Reefs and Sydney Harbour, Australia. The system utilizes the highly irregular and complex natural features on the sea bed as navigational landmarks. A successful implementation of the concept would remove the need of artificial beacons for an autonomous underwater exploration by robotic vehicles.

The third and fourth articles present two very impressive examples of experiments with very large scale robotic teams. These experiments were carried out by the University of Southern California / University of Tennessee team and the SRI / University of Washington team. Both teams of robots performed the same tasks, in the same building, as a part of DARPA's Software for Distributed Robotics (SDR) Project. It is therefore very interesting to gain insight into the problem from two different points of view of the research teams. Each team fielded between 80 – 100 mobile robots which were instructed to map the large building, establish a sensor network, locate the targeted object and detect any intruders. The reports from such rare opportunities of testing the algorithms in very large scale and realistic scenarios is a very valuable step in the progress of this field.

Another large scale and ambitious project is presented in the last article by Montemerlo and Thrun at Stanford University. The aim was to rapidly acquire an accurate 3D map of an urban environment, seamlessly integrating indoor and outdoor structures. The autonomous robot was realized using a Segway scooter mounted with a SICK laser sensor mounted sideways and servo driven around a vertical axis. The results shown was an impressive 3D map of several structures, including Stanford University's main campus, including the palm trees that line the walkways!

Topological Global Localization and Mapping with Fingerprints and Uncertainty

Adriana Tapus, Nicola Tomatis and Roland Siegwart

Autonomous Systems Lab
Swiss Federal Institute of Technology, Lausanne (EPFL)
1015 Lausanne, Switzerland
{Adriana.Tapus, Nicola.Tomatis, Roland.Siegwart}@epfl.ch

Abstract. Navigation in unknown or partially unknown environments remains one of the biggest challenges in today's mobile robotics. Environmental modeling, perception, localization and mapping are all needed for a successful approach. The contribution of this paper resides in the extension of the fingerprint concept (circular list of features around the robot) with uncertainty modeling, in order to improve localization and allow for automatic map building. The uncertainty is defined as the probability of a feature of being present in the environment when the robot perceives it. The whole approach is presented in details and viewed in a topological optic. Experimental results of the perception and localization capabilities with a mobile robot equipped with two 180° laser range finders and an omni-directional camera are reported.

1 Introduction

Navigation, described by Gallistel in [5] as the capacity to localize itself with respect to a map, is an elementary task that an autonomous mobile robot must carry out. Both, accurate perception and a reliable environmental modeling are needed, in order to localize a mobile robot and to build a map of its environment. Many methods have been proposed to represent environments in the framework of autonomous navigation, from precise geometric maps based on raw data or lines up to purely topological maps using symbolic descriptions. Each one of these methods is optimal concerning some characteristics but can be very disappointing with respect to other requirements. Metric maps are suited when the robot needs to know its location accurately in terms of metric coordinates. However, in office buildings with corridors and rooms, or roads, the topology of important locations and their connections might be sufficient for navigation. Topological maps are less complex and permit more efficient planning than metric maps. Moreover, it is easier to generate and maintain global consistency for topological maps than for metric maps. Even though research has recently leaded to successful solutions, robust perception for robot localization in unmodified, dynamic, real-world environments is still a challenge.

In this paper we concentrate on how multimodal perception combined with the uncertainty modeling of the features increases the reliability for topological localization and permits improving map building (map update). For the

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 99–110, 2006.

topological framework the fingerprint concept is used. This type of representation permits a reliable and distinctive environment modeling. The goal is to model the error-prone measurements from the imperfect exteroceptive sensors by means of uncertainty associated to their data.

Early works in topological localization [6] presented experiments in simulations, which avoided facing the perception problem. Following works as [10] were concerned with controlled environments, where perception with sonars was enough for the navigation purpose. Only more recent works address the perception problem in its whole complexity in the real world. Successful vision-based navigations are currently limited to indoor navigation because of its dependence on ceiling features [14], room geometry, or artificial landmark [12]. Other means for visual localization are applicable both indoors and outdoors, however they are designed to collect image statistics while foregoing recognition of specific scene features, or landmarks [13, 16]. In this context [7] and [8] introduced the fingerprint concept. Here, we show how the extension of the fingerprint concept with uncertainty modeling improves the topological global localization and mapping.

The remainder of this paper is organized as follows. We present in Section 2 the fingerprint concept, the way it is encoded and generated. In Section 3, we define the uncertainty model for the features present in the fingerprint. Section 4 is dedicated to the new method used for the fingerprint matching. In Section 5 a brief description of the localization and mapping is depicted. Experimental results are presented in Section 6. The system will use both, a laser scanner and an omni-directional camera for feature extraction. To conclude, Section 7 contains a discussion of the proposed approach and further research directions.

2 The Fingerprint Concept in a Topological Framework

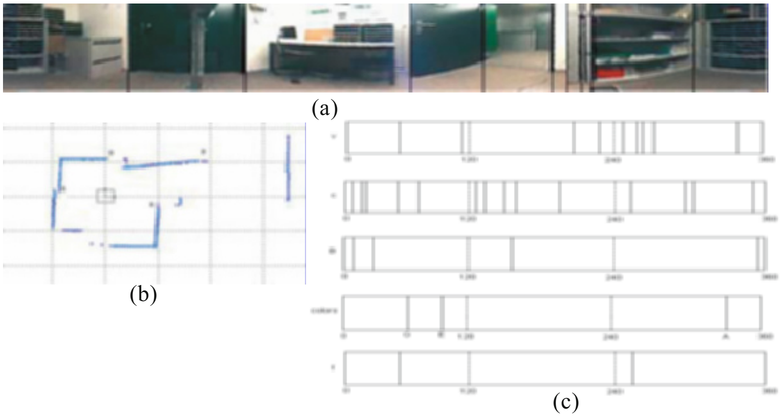
The topological approach yields a compact representation and allows high-level symbolic reasoning for map building and navigation. With this method we try to eliminate the perceptual aliasing (i.e. distinct locations within the environment appearing identical to the robot's sensors) and to improve the distinctiveness of places in the environment. To maximize the reliability in navigation, the information from all sensors available to the robot must be used. For this, the notion of fingerprint as described in [7, 8] is used. This characterization of the environment is especially interesting when used within topological localization and multiple sensor modalities.

2.1 Fingerprint Encoding

A fingerprint is a circular list of features, where the ordering of the set matches the relative ordering of the features around the robot. We denote the fingerprint sequence using a list of characters, where each character represents the instance of a specific feature type. In our case we choose to extract color patches and vertical edges from visual information and corners and beacons from laser scanner. We use the letter ′v′ to characterize an edge, the letters ′A′, ′B′, ′C′, ... , ′P′ to represent hue bins, the letter ′c′ to characterize a corner feature and the letter ′b′ to characterize a beacon feature. Details about the visual features extraction can be found in [7] and laser scanner features extraction in [2].

2.2 Fingerprint Generation

The fingerprint generation is performed in three steps (see Figure 1). The extraction of the different features (e.g. vertical edges, corners, color patches, beacons) from the sensors is the first phase of the fingerprint generation. The order of the features, given by their angular positions (0 … 360°) is kept in an array. At this stage a new type of feature, the virtual feature ′f′ is introduced. It reflects a correspondence between a corner and an edge. The ordering of the features in a fingerprint sequence is highly informative and for that reason the notion of angular distance between two consecutive features will be added. This adds geometric information and increases once again the distinctiveness between fingerprints. Furthermore, we introduced an additional type of feature, the empty space feature ′n′, for reflecting angular distances. Each ′n′ covers the same angle of the scene (20°). This insertion is the last step of the fingerprint generation. More details can be found in [8].


(a)

(b)

(c)

Figure 1. Fingerprint generation. (a) Panoramic image with the vertical edges and color patches (′v′ and color); (b) laser scan with extracted corners ′c′ and beacons ′b′; (c) images one to four depict the position (0 to 360°) of the vertical edges, the corners, the beacons and the colors (G-green, E-light green, and A-red). The fifth image describes the correspondence between the vertical edges and the corners. By regrouping all this together and by adding the empty space features, the final fingerprint is: cbccbnfGcnEnvccncbcvncnnfvvvnccAcb.

3 Uncertainty Modeling in the Fingerprint Approach

The interaction between the mobile robot and its surroundings is performed by means of exteroceptive sensor data. Of course, the sensors are imperfect devices, and thus the measurements always contain errors. This can be modeled by associating uncertainty to their data. For that reason, the probability theory will be used to model the uncertainty of the geometric features extracted from the environment. We define the uncertainty as the probability of a feature of being present in the environment when the robot perceives it. The main idea is to introduce a new element in the fingerprint approach that specifies this uncertainty. Such uncertainty is modeled by experience, for each type of feature presented in Figure 1: vertical edges, colors, corners (extremities of the segments), beacons, ′f′ feature and respectively ′n′ feature. In the following, it will be shown how the uncertainty, denoted by the symbol *u*, is calculated for each one of the features:

- For the first three types of features (vertical edges, colors and corners) the uncertainty is calculated by using the following schema.

$$u = \begin{cases} \min, if\ extraction\_value \le low\_bound \\ linear\ interpolation, if\ low\_bound \le extraction\_value \le high\_bound \\ \max, if\ extraction\_value \ge high\_bound \end{cases}$$

  The *extraction_value* variable changes in function of the type of the feature. For the vertical edges, *extraction_value* corresponds to the gradient value. For the colors, the *extraction_value* is represented by the hue value of the color. In the case of the corner features, the *extraction_value* is identified as the distance between the robot and the extremities of the segments. The values of the *low_bound* and *high_bound* are experimentally determined for each type of feature. The *low_bound* represents the bound below which the feature has a low probability of existence so that the robot may not see it at the next passage. Another important element is the *high_bound*, above which a feature has a high certainty to exist and to be in the place where it was found (extracted). The *low_bound* and the *high_bound* are determined for each feature at the extraction level. The extraction of the vertical edges consists of the application of a threshold function on the gradient values. Since all edges below the threshold are ignored, the *low_bound* is used as the threshold value. The *high_bound* must be high, but not max_gradient, and so it has been fixed experimentally at *(threshold + mean_gradient)*. To extract the colors, a threshold function on the hue values has been applied and a similar method to that applied for the vertical edges has been chosen. Other two important elements in the schema described before are the values of *min* and *max*, which are fixed to 0.6, respectively 0.99. A method for calculating the value in-between these two values (*min* and *max*), by knowing the *low_bound*, *high_bound* and the *extraction_value* can be obtained by linear interpolation. In this case, *u* will be equal to:

$$u = \min + \frac{extraction\_value - low\_bound}{High\_bound - low\_bound} \times (\max - \min)$$

- The beacons are artificial landmarks (i.e. reflectors) and they are extracted with the help of the laser range finders. The experiments showed that the beacons are detected all the times, and for that reason the uncertainty has been fixed at a high value.
- As the ′f′ feature reflects the correspondence between a corner and an edge, its uncertainty is defined as the mean value between the uncertainty of the corner and the uncertainty of the vertical edge feature.
- The last feature is the ′n′ feature (i.e. the empty space feature that represents the angular distance between the features). The uncertainty of this feature is proportional to the distance between the features.

In this way, the uncertainty of the features used in the fingerprints is calculated. The limitation of this method resides in the models, which are difficult to define, especially for our definition of uncertainty, which cannot be directly derived from the physical characteristics of the sensors.

4 Fingerprint Matching

The string-matching problem is not easy. Usually strings do not match exactly because the robot may not be exactly located on a map point and/or some changes in the environment or perceptual errors occurred. Many string-matching algorithms can be found in the literature but they generally require the strings to have the same length. Some of them allow a level of mismatch, such as the k-mismatch matching algorithms and string matching with k differences [1, 3]. The first allows matches where up to k characters in the pattern do not match the text and the second requires that the pattern have an edit distance from the text of k or less. One of the main problems of the above methods is that they do not consider the nature of features and specific mismatches. We wish to consider the likelihood of specific types of mismatch errors. For instance confusing a red patch with a blue patch is more egregious than confusing the red patch with a yellow patch. The standard algorithms are quite sensitive to insertion and deletion errors, which cause the string lengths to vary significantly. The methods adopted previously in the fingerprint approach for sequence matching are the minimum energy algorithm used in stereovision [5] and the global alignment used usually for D.N.A. sequences [9]. Our current approach is an extension of the global alignment algorithm considering uncertainties and it is described below.

4.1 Global Alignment with Uncertainty

The global alignment algorithm finds an alignment between two strings so that the cost is minimal by using the cost function for aligning two characters.

Before starting describing the algorithm, the idea of aligning two strings and calculating the cost will be illustrated with an example (see Figure 2):

string1 := « abcd »
string2 := « bbc »

$$\mathrm{cost}(x,y) := \begin{cases} x=\varepsilon \,|\, y=\varepsilon : & 0.6 \\ x=y & :0.0 \\ \text{else} & :1.0 \end{cases}$$

The cost of alignment $\begin{bmatrix} abcd \\ bbc\varepsilon \end{bmatrix}$ is calculated as:

$$= \mathrm{cost}('a','b') + \mathrm{cost}('b','b') + \mathrm{cost}('c','c') + \mathrm{cost}('d','\varepsilon')$$

$$1 \quad + \quad 0 \quad + \quad 0 \quad + \quad 0.6 \quad = \quad 1.6$$

Figure 2. An example of calculating the cost between two strings.

More formally, we can distinguish five elements, which form the global alignment algorithm (see Figure 3). The first element is an alphabet A, typically a set of letters, which is not empty. The second element corresponds to the two strings which are to be aligned: the first is composed of *m*, the second of *n* letters of the alphabet. The occlusion symbol is used to represent a space inserted into the string. The cost function gives the cost for the match between two symbols of the alphabet, included the occlusion symbol. Finally, the cost matrix is used to keep the minimal cost of a match between the first *i* letters of the first string with the first *j* letters of the second string, keeping this value in the element (*i,j*) of matrix V.

Alphabet	A, A ≠ { }
Strings	S1$\in A^m$, S2$\in A^n$, m, n ∈ N
Occlusion symbol	ε, ε ∉ A
Cost function	$f_{\cos t} : a \in A \cup \varepsilon, b \in A \cup \varepsilon \rightarrow \Re$
Cost Matrix	$V_{(i,j)} \in \Re$, i∈{0, 1, ..., m}, j∈{0, 1, ..., n}

Figure 3. The main elements of the Global Alignment algorithm.

The values of the cost function *$f_{cost}(a, b)$*, are calculated experimentally in function of the similarity between characters *a* and *b*, in other words the more similar the characters are, the lower will be the penalty for mismatching. It only remains to calculate the values of the elements of the cost matrix, which is constructed by a technique named "dynamic programming". Initially the edges of the matrix are initialized with the cumulative cost of occlusions. (That reflects the fact that we do not know, a priori, how much letters must be jumped in one or the other string in order to obtain the best solution.).
The base conditions of the algorithm are:

- $V(0,j) = \sum_{1 \le k \le j} f_{\cos t}(\varepsilon, S2(k))$
- $V(i,0) = \sum_{1 \le k \le i} f_{\cos t}(S1(k), \varepsilon)$

For *i* and *j* both strictly positive, the recurrence relation is:

$$V(i,j) = \min\begin{cases} V(i-1, j-1) + f_{\cos t}(S1(i), S2(j)) \\ V(i-1, j) + f_{\cos t}(S1(i), \varepsilon) \\ V(i, j-1) + f_{\cos t}(\varepsilon, S2(j)) \end{cases}$$

The three cases that can be distinguished from the above relation are:

- **Aligning** *S*1*(i)* **with** *S*2*(j)*: The score in this case is the score of the operation $f_{cost}(S1(i),S2(j))$ plus the score of aligning *i*-1 elements of *S*1 with *j*-1 elements of *S*2, namely, $V(i\text{-}1,j\text{-}1)+f_{cost}(S1(i),S2(j))$
- **Aligning** *S*1*(i)* **with an occlusion symbol in string** *S*2: The score in this case is the score of the operation f_{cost} ($S1(i)$, ε) plus the score of aligning the previous *i*-1 elements of *S*1 with *j* elements of *S*2 (Since the occlusion is not an original character of *S*2), $V(i\text{-}1,j)+f_{cost}$ ($S1(i)$, ε)
- **Aligning** *S*2*(j)* **with an occlusion symbol in string** *S*1: Similar to the previous case, the score will be $V(i,j\text{-}1)+f_{cost}$ (ε,$S2(j)$).

If strings *S*1 and *S*2 are of length *n* and respectively *m*, then the value of their optimal alignment with the global alignment is the value of the cell (*n,m*).

The global alignment with uncertainty changes only the cost function described earlier. The cost function is adapted in order to take into account the corresponding uncertainty of features. The goal of adding the uncertainty in the string matching algorithm is to improve the distinctiveness of places. Next, a small example of global alignment algorithm with uncertainty will show the improvement of the matching (see Figure 4).

The example depicted in Figure 4 shows the improvement gained by the new fingerprint matching with uncertainty algorithm. Even if the two fingerprints from the map are similar (i.e. string1 and string2), the uncertainty of features will determine the map fingerprint that matches best the observed fingerprint (i.e. stringObs).

string1 := « abcde »
uncertainty_s1 := « 1 1 0.1 1 1 »
string2 := « abcde »
uncertainty_s2 := « 1 1 1 1 1 »

stringObs := « abde »
uncertainty_sObs:= « 1 1 1 1 »

The alignment between the string1 and stringObs is the same with the alignment of the string2 and stringObs:

$$\begin{bmatrix} a\,b\,\varepsilon\,d\,e \\ ab\,c\,d\,e \end{bmatrix}$$

The probability that the stringObs matches the string1 is of 95% and the probability that the stringObs matches the string2 is of 75%.

Figure 4. An example of the Global Alignment algorithm with the uncertainty.

5 Topological Localization and Mapping

In this section a brief description of the global topological localization and map building approach is presented.

For the topological navigation a Partially Observable Markov Decision Process (POMDP) model is used.

A POMDP is defined as <*S,A,T,O*>, where *S* is a finite set of environment states; *A* is a finite set of actions; *T(s,a,s')* is a transition function between the environment states based on the action performed. A finite set *O* of possible observations and an observation function *OS* will be added. With this information, the probability of being in a state *s'* (belief state of *s'*) after having made observation *o*, while performing action *a*, is given by:

$$SE_{s'}(k+1) = \frac{OS(o,s',a)\sum_{s\in S} T(s,a,s')SE_s(k)}{P(o\ a, SE(k))}$$

The key idea is to compute a discrete approximation of a probability distribution over all possible poses in the environment. An important feature of this localization technique is the ability to globally localize the robot within the environment. More details about this approach can be found in [4].

The information for the observation function within the topological framework is given by the fingerprint matching algorithm, described in the previous section.

While navigating in the environment, the robot firstly creates and then updates the global topological map. Each node contains the topology and door situation (i.e. corridor, T intersection, + Intersection, L Intersection, room, closed door, opened door, partially left opened door, partially right opened door and no door. For the doors, with the direction: in front, behind, on the left or on the right of the robot.) and the associated fingerprint. More details about the topology can be found in [12].

The entropy of a probability distribution p is

$$H(p) = -\sum_{s \in S} ps \log ps,$$

where $p_s \log p_s = 0$ when $p_s = 0$. The lower the value, the more certain the distribution. When the robot is "confused", the entropy is high. Therefore, the strategy of updating the map will be the following:

- When the entropy of the belief state is low enough, the map will be updated and so the fingerprint and the uncertainty of the features will also be updated.
- If the entropy is above a threshold α, then the updating will not be allowed, and we will try to reduce the entropy by continuing the navigation with localization.

Similarly to [15], when the robot feels confident concerning its state, it can decide if an extracted feature is new by comparing the observation fingerprint to the fingerprint from the map, corresponding to the most likely state. This can happen either in an unexplored portion of the environment, or in a known portion where new features appear due to the environmental dynamics. The features from the fingerprint come with their extraction uncertainty u. When a feature is re-observed, the uncertainty of the feature from the map fingerprint is averaged with the uncertainty of the extracted one. Otherwise, if the robot does not see an expected feature the uncertainty is decreasing. When the uncertainty of a feature from a map fingerprint is below a minimum threshold, than the feature is deleted, allowing in this way for dynamics in the environment.

6 Experimental Results

The approach has been tested in a 50 x 25 m^2 portion of our institute building.

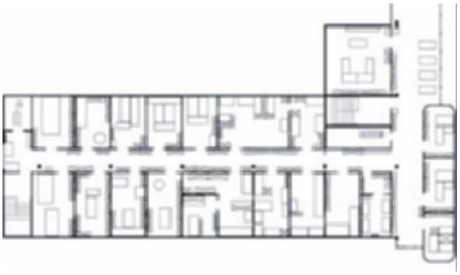


Figure 5. The test environment, with the rooms and corridors in which the experimentation has been done.

For the experiments, the Donald Duck robot (see Figure 6), a fully autonomous mobile robot, has been used.

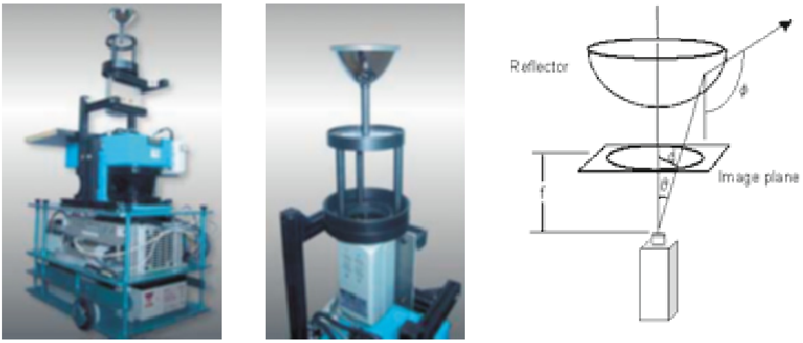


Figure 6. System used for experimentation: The fully autonomous robot Donald Duck and the panoramic vision system. The camera has a 640 x 480 pixel resolution and an equiangular mirror is used so that each pixel in the image covers the same view angle.

Its controller consists of a VME standard backplane with a Motorola PowerPC 604 microprocessor clocked at 300 MHz and running XO/2, a hard real-time operating system. Among its peripheral devices, the most important are the wheel encoders, two 180° laser range finders and an omni-directional camera. The panoramic vision system depicted in Figure 6 uses a mirror-camera system to image 360° in azimuth and up to 110° in elevation.

In order to validate the fingerprint approach with uncertainty, a comparison between the results obtained with the non-probabilistic approaches and the results obtained with the new probabilistic version will be presented. The experiments for all the approaches have been tested in the same environment and under the same conditions.

The test setup was the following: The robot extracted the four features (i.e. vertical edges, colors, corners and beacons) in seven offices at 11 different places. For the new matching approach, the uncertainties of different features have been modeled. One fingerprint per room has been included in a database as reference (map initialization) for the localization approach. The other 70 fingerprints have been matched to the database for testing the localization.

During all measurements, the orientation of the robot was approximately the same. This simplification could be omitted by letting the robot estimate his orientation by considering all rotations of the fingerprint string.

For a given observation (fingerprint) a match is successful if the best match with the database corresponds to the correct room. Table 1 illustrates the percentage of successful matching and the mean rank for three string matching algorithms: minimum energy, global alignment and global alignment with uncertainty. The rank calculates the position of the correct room, with respect to the others, in the classification (e.g. if the match is successful than the rank is 1, if the correct room is detected with the second highest probability the rank is fixed at 2, etc.). 100100

Table 1. Classification using string matching, comparing minimum energy, global alignment and global alignment with uncertainty algorithms.

	right classifications	mean rank
minimum energy	58.82%	1.85
global alignment	75%	1.32
global alignment with uncertainty	83.82%	1.23

In Table 1, one can see the improvement from using global alignment with uncertainty instead of the global alignment or minimum energy algorithm. The results with global alignment with uncertainty algorithm have 83.82% of successful matches, which corresponds to a clear improvement of 8.82% with respect to the standard global alignment (see Table 1). Note that the experimental setup does not include yet the presented Partial Observable Markov Decision Process (POMDP) for localization. However, as soon as the matching information will be integrated by the POMDP, the motion will bring additional information to the system that should allow very reliably navigation.

7 Conclusion and Future Work

This paper has presented a method for topological global localization and mapping by using the fingerprint concept combined with an uncertainty modeling. The fingerprint approach [7, 8] has already shown its capability of representing real world scenes in a robust and flexible manner. The uncertainty model, presented here as the probability of a feature being present in the environment when the robot perceives it, improves the concept and allows for a more stable global localization. The performance of the probabilistic fingerprint approach is shown through experiments comparing the new approach with two old versions. From the experiments we can conclude that the presented method is practical and robust. The successful classification is 83.82% which represents an improvement of 25% in comparison with the minimum energy approach and 8.82% with the standard global alignment. Even though the matching is not yet integrated with POMDP, we can already state that all matching steps will bring important information to the system since the correct fingerprint has a mean rank of 1.23. Future work will focus on the integration of the fingerprint with uncertainty within a POMDP for Simultaneous Localization and Mapping (SLAM) and the extension of the whole approach towards multi-resolution SLAM.

Acknowledgments

This work was sponsored by the European project BIBA IST-2001-32115.

References

[1] Alfred V. Aho. Algorithms for finding patterns in strings. *In J. van Leeuwen, editor, Handbook of Theoretical Computer Science*, chapter 5, pages 254-300. Elsevier Science Publishers B. V., 1990.

[2] Arras, K.O. and Siegwart, R, Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building, In Proceedings of the Symposium on Intelligent Systems and Advanced Manufacturing, Pittsburgh, USA, October 13-17,1997.

[3] Baeza-Yates, R., Navarro, G. Faster Approximate String Matching, Department of Computer Science, University of Chile, Santiago.

[4] Cassandra, A. R., L. P. Kaelbling, et al. (1996). Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation*. IEEE International Conference on Robotics and Automation*, Osaka, Japan.

[5] Gallistel, R. (1990), The Organization of Learning, MIT Press: Cambridge, MA.

[6] Kuipers, B. J. and Y. T. Byun (1987). A qualitative approach to robot exploration and map-learning. Workshop on Spatial Reasoning and Multi-Sensor Fusion, Los Altos, CA, USA, Morgan Kaufmann.

[7] Lamon, P., I. Nourbakhsh, B. Jensen and R. Siegwart (2001). Deriving and Matching Image Fingerprint Sequences for Mobile Robot Localization. IEEE International Conference on Robotics and Automation, Seoul, Korea.

[8] Lamon, P., Tapus A., Glauser E., Tomatis N., Siegwart R. (2003), Environmental Modeling with Fingerprint Sequences for Topological Global Localization, In Proceedings of the IEEE International Conference on Intelligent Robot and Systems, Las Vegas, USA, October 27-30.

[9] Needleman, S. and Wunsch, C. (1970), A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Molecular Biology, 48:443-453.

[10]Nourbakhsh, I. (1998). Dervish: An Office-Navigating Robot. Artificial Intelligence and Mobile Robots. D. Kortenkamp, R. P. Bonasso and R. Murphy, The AAAI Press/ The MIT Press: 73-90.

[11]Nourbakhsh, I., J. Bodenage, et al. (1999). An Affective Mobile Robot Educator with a Full-Time Job. Artificial Intelligence 114(1-2): 95-124.

[12]Tapus A., Ramel G., Dobler L. and Siegwart R. (2004), Topology Learning and Recognition using Bayesian Programming for Mobile Robot Navigation, In Proceedings of the IEEE International Conference on Intelligent Robot and Systems, Sendai, Japan.

[13]Thrun, S., Finding Landmarks for Mobile Robot Navigation, IEEE International Conf. on Robotics and Automation, May 1998.

[14]Thrun, S., Bennewitz, M., Burgard, W., Cremers, A.B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., Schulz, D., MINERVA: a second-generation museum tour-guide robot, IEEE International Conf. on Robotics and Automation (Cat. No.99CH36288C) 1999.

[15]Tomatis, N., I. Nourbakhsh, and R. Siegwart (2003). Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems,* 44:3-14.

[16]Ulrich, I. and I. Nourbakhsh (2000). Appearance-Based Place Recognition for Topological Localization. IEEE International Conference on Robotics and Automation, San Francisco, CA.References

Terrain Aided Localisation and Mapping for Marine Environments

Stefan Williams and Ian Mahon

ARC Centre of Excellence in Autonomous Systems (CAS)
The University of Sydney, Australia
{s.williams, i.mahon}@cas.edu.au
http://www.cas.edu.au

Abstract. This paper presents experimental results of the application of terrain aided localisation and mapping algorithms to vehicle deployments in marine environments. The application of a terrain aided navigation filter to the tracking of a ship operating on Sydney Harbour is described. This approach allows highly unstructured seafloor bathymetric information to be incorporated into the navigation solution. In addition, experimental validation of the Simultaneous Localisation and Mapping algorithm using data collected by an Unmanned Underwater Vehicle operating on the Great Barrier Reef in Australia is reported. By fusing information from the vehicle's on-board sonar and vision systems, it is possible to use the highly textured reef to provide estimates of the vehicle motion as well as to generate models of the gross structure of the underlying reefs. Terrain-aided navigation promises to revolutionise the ability of marine systems to track underwater bodies in many applications. This work represents a crucial step in the development of underwater technologies capable of long-term, reliable deployment.

1 Introduction

This paper presents results of the application of Terrain Aided Navigation and Simulatneous Localisation and Mapping (SLAM) methods to the tracking of marine vehicles operating in unstructured, natural environments. Terrain-aided navigation promises to revolutionise the ability of marine systems to track underwater vehicles in subsea applications. The development of techniques for extracting navigation 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 [11][22]. The ability to use natural features will allow a submersible 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 details demonstrations of map based navigation techniques undertaken using bathymetric maps of Sydney Harbour. Section 4 gives a brief summary of the SLAM algorithm and describes the approach adopted for this deployment. Finally, Section 5 summarizes the findings and presents directions for future work.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 111–120, 2006.

2 Underwater positioning

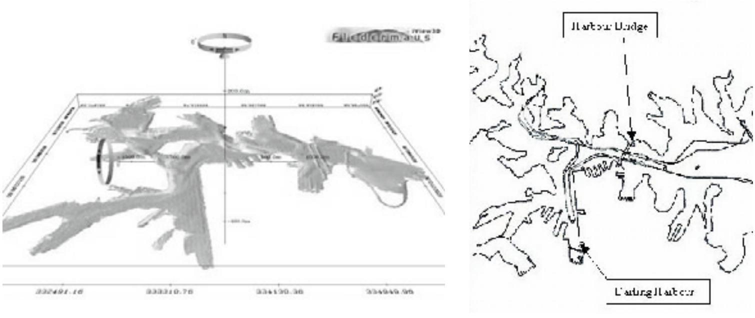
While many land-based tracking systems use GPS to provide accurate position updates for navigation, a body operating underwater does not typically have access to this type of information. Many underwater systems rely on fixed acoustic transponders that are surveyed into the system's work area [19]. These 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.

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 [16] and other work has looked at methods for constructing seafloor terrain elevation maps [5] but these techniques have not been deployed on a system operating in real, unstructured environments. Recent work has also examined the possibility of using Monte Carlo methods for position tracking given a map of the environment, as shown in related work in [17] and [22]. 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.

It is our feeling that a combination of terrain aided and acoustic navigation systems will be required before autonomous marine systems will be able to operate for long periods of time, especially in and around significant seafloor structures such as coral reefs. Acoustic systems will tend to operate at their peak in open ocean environments where the chance of multi-path and occlusions are low. In areas where the vehicle must operate in close proximity to the sea floor or in and around sea floor structures or canyons, the acoustic signals are likely to be corrupted. Conversely, terrain aided systems will perform poorly in open ocean or flat bottomed areas of the seafloor. They will, however, perform very well in areas where there is significant variation or texture on the sea floor. A combination of these two navigation techniques are therefore more likely to yield robust solutions to the problem of positioning of underwater vehicles.

3 Map Based Localisation

In underwater scientific missions, *a priori* maps are often available. Bathymetric data collected by ships can be compiled to generate sea floor elevation maps [6][14] and Geospatial Information Systems (GIS) exist that allow this information to be efficiently accessed. The work undertaken as part of this demonstration allows 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.



(a) (b)

Fig. 1. 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.

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 [18], in road following [4], in aerial navigation [1] and for underwater applications [17]. 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 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 1 (a), and ship transect data, including GPS and depth soundings, shown in Figure 1 (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 [22] have been applied to these data sets. Figure 2 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

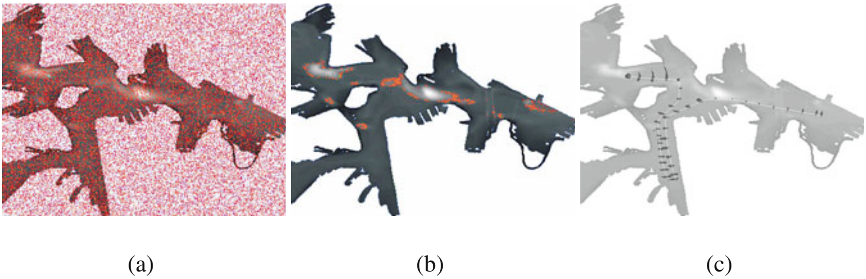


Fig. 2. Monte Carlo localisation example using the Sydney Harbour bathymetric map. The line represents the ship track in this deployment and the particles are shown overlaid on the figure. (a) The particles are initially drawn from the uniform distribution across the extent of the harbour. (b) After taking a depth sounding, the potential location of the ship is reduced to areas of the harbour with a common depth to the start of the trial. (c) The particles converge to the actual position of the ship and tracking its motion 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.

the deployment. The particle clouds converge to the true ship position within the first 20 observations and successfully track the remainder of the ship track. 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.

One of the limiting factors with this approach to the terrain aided navigation problem will be the resolution of the *a prior* map and the variability in the terrain structure. For deepwater applications, the map resolution achievable from the surface may be quite low. In this case we are working towards a mechanism for incorporating higher resolution terrain information taken when the vehicle is in close proximity to the seafloor into the prior, low resolution map. This will require an efficient method for representing and updating the map information.

4 Simultaneous Localisation and Mapping

In cases where *a priori* maps are not available, a mechanism must exist for building a map while simultaneously using that map for the purposes of localisation. Simultaneous Localisation and Mapping (SLAM) is the process of concurrently building a feature based map of the environment and using this map to obtain estimates of the location of the vehicle. We have recently demonstrated the SLAM algorithm operating on data collected by an Unmanned Underwater Vehicle operating on the Great Barrier Reef [21]. The vehicle, shown in Figure 4 (a), is equipped with two scanning low frequency terrain-aiding sonars, a depth sensor, fibre-optic gyroscope, compass with integrated roll and pitch sensors and two colour CCD cameras.

The SLAM algorithm has seen a considerable amount of interest from the mobile robotics community as a tool to enable fully autonomous navigation [2][3]. The prospect of deploying a system that can build a map of its environment while simultaneously using that map to localise itself promises to allow these vehicles to operate autonomously for long periods of time in unknown environments. Much of this work has focused on the use of stochastic estimation techniques to build and maintain estimates of vehicle and map feature locations. In particular, the Extended Kalman Filter (EKF) has been proposed as a mechanism by which the information gathered by the vehicle can be consistently fused to yield bounded estimates of vehicle and landmark locations in a recursive fashion [3][7]. Recent work has concentrated on the development of efficient methods for implementing the algorithm using relative maps [15] and submaps [8] [20].

The SLAM algorithm represents the state of the environment and the state of the vehicle within it by the augmented state vector, $\hat{\mathbf{x}}^+(k)$, consisting of the n_v states representing the vehicle, $\hat{\mathbf{x}}_v^+(k)$, and the n_f states describing the observed features, $\hat{\mathbf{x}}_i^+(k), i = 1, ..., n_f$. The augmented state estimate consists of the current vehicle state estimates as well as those associated with the observed features

$$\hat{\mathbf{x}}^+(k) = \left[\hat{\mathbf{x}}_v^+(k)\ \hat{\mathbf{x}}_1^+(k)\ \ldots\ \hat{\mathbf{x}}_{n_f}^+(k)\right]^T \tag{1}$$

The covariance matrix for this state estimate is defined through

$$\mathbf{P}^+(k) = E[(\mathbf{x}(k) - \hat{\mathbf{x}}^+(k))(\mathbf{x}(k) - \hat{\mathbf{x}}^+(k))^T|\mathbf{Z}^k]. \tag{2}$$

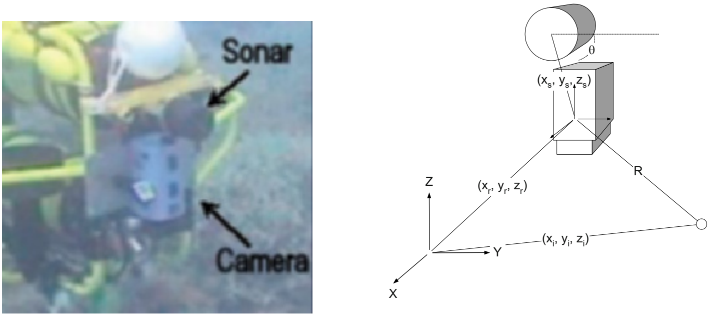
This defines the mean squared error and error correlations in each of the state estimates. For the case of the SLAM filter, the covariance matrix takes on the following form using $\mathbf{P}_{vv}^+(k)$ to represent the vehicle covariances, $\mathbf{P}_{mm}^+(k)$ to represent the map covariances and $\mathbf{P}_{vm}^+(k)$ to represent the cross-covariance between the vehicle and the map.

$$\mathbf{P}^+(k) = \begin{bmatrix} \mathbf{P}_{vv}^+(k) & \mathbf{P}_{vm}^+(k) \\ \mathbf{P}_{vm}^{+T}(k) & \mathbf{P}_{mm}^+(k) \end{bmatrix} \tag{3}$$

The SLAM filter equations are derived directly from the standard Extended Kalman Filter formulation and can be found in numerous texts and papers on the subject [3][13].

In the current implementation of the SLAM algorithm, it was assumed that the roll and pitch of the vehicle were negligible. This assumption is based on observations of the vehicle performance during in-water tests and results primarily from the design of the vehicle, which incorporates a pair of pressure vessels stacked vertically and a keel mounted at the bottom of the vehicle. This configuration of the vehicle body results in a platform that is very stable in pitch and roll. As a result, the vehicle pose at time step k is represented by

$$\hat{\mathbf{x}}_v^+(k) = \left[\hat{x}_v^+(k)\ \hat{y}_v^+(k)\ \hat{z}_v^+(k)\ \hat{\psi}_v^+(k)\right]^T \tag{4}$$



(a) Sonar and Camera

(b) Schematic

Fig. 3. The pencil beam scanning sonar mounted directly above the vehicle's high resolution vision system. The sonar scans the sea floor directly below the vehicle. The returns can be projected into the visual image to initialise high contrast features in the SLAM map.

and the features states are represented by

$$\hat{\mathbf{x}}_i^+(k) = \left[\hat{x}_i^+(k)\ \hat{y}_i^+(k)\ \hat{z}_i^+(k)\right]^T \tag{5}$$

4.1 Fusing sonar and vision

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. In these experiments, a pencil beam scanning sonar is mounted directly above the vehicle's high resolution vision system as shown in Figure 3. The sonar scans the sea floor directly below the vehicle and is used to generate profiles of the surface over which the vehicle travels. The returns received by the sonar are then processed to identify the principal returns which are used to initialise new features in the SLAM map. The observation of range, R, and bearing, θ, are combined with the estimated value of the vehicle pose, $\hat{\mathbf{x}}_v(k)$, and the measured position, $\left[x_s\ y_s\ z_s\right]^T$ of the sonar relative to the camera frame.

$$\begin{bmatrix}\hat{x}_i\\ \hat{y}_i\\ \hat{z}_i\end{bmatrix} = \begin{bmatrix}\hat{x}_v + x_s\cos\hat{\psi}_v - (y_s + R\cos\theta)\sin\hat{\psi}_v\\ \hat{y}_v + x_s\sin\hat{\psi}_v + (y_s + R\cos\theta)\cos\hat{\psi}_v\\ \hat{z}_v + z_s + R\sin\theta\end{bmatrix} \tag{6}$$

4.2 Extracting visual features

Once the sonar returns have been initialised within the SLAM framework, they are tracked using observations from the vehicle's high resolution vision system. When a new feature is initialised, the sonar footprint is projected into the visual frame of the camera. The centre of this footprint is used to identify a high contrast feature in the image within the area insonified by the sonar. While there is no guarantee that the resulting feature will in fact correspond to the return identified within the sonar ping due to the beam width of the sonar, at close range the sonar footprint is relatively small and the features appear to correspond quite closely to the initialised feature locations. Currently, these visual features are tracked from frame to frame using the Lucas and Kanade feature tracking technique [10] to provide the SLAM algorithm with bearing only observations of the feature positions. Observation of elevation, z_e, and azimuth, z_a, within the visual frame are provided to the SLAM algorithm. The estimated elevation and azimuth are computed using the current estimate of the vehicle pose and feature position.

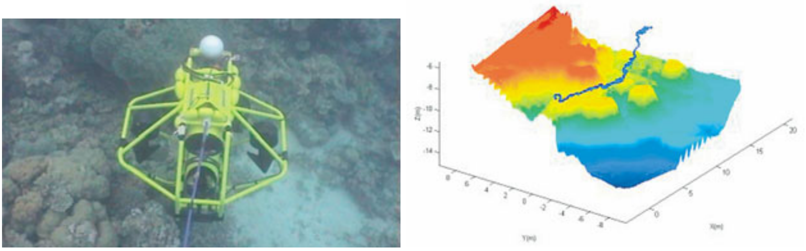
$$\begin{bmatrix} \hat{z}_e \\ \hat{z}_a \end{bmatrix} = \begin{bmatrix} \frac{(\hat{x}_i - \hat{x}_v)\cos\hat{\psi}_v + (\hat{y}_i - \hat{y}_v)\sin\hat{\psi}_v}{\hat{z}_i - \hat{z}_v} \\ \frac{-(\hat{x}_i - \hat{x}_v)\sin\hat{\psi}_v + (\hat{y}_i - \hat{y}_v)\cos\hat{\psi}_v}{\hat{z}_i - \hat{z}_v} \end{bmatrix} \tag{7}$$

The difference between the actual observation received from the camera and the predicted observation is termed the innovation. This innovation is used in computing the posterior state estimate according to the standard EKF update equation [13]. Two classes of features are used in the underwater SLAM implementation. A large number of transient features are tracked to provide accurate short-term motion estimation, and a smaller set of stable features are maintained as landmarks.

The positions of transient features are initialised using the range and bearing estimates from sonar observations. While such features are unlikely to be repeatedly observed in the sonar sensor's narrow footprint, their location in the camera's reference frame can be calculated using knowledge of the sensors' geometry. Additional azimuth and elevation observations can then be obtained to aid localisation by tracking a feature through subsequent image frames. The Lucas-Kanade algorithm [10] is used in our implementation to track rectangular regions of pixels around the transient features. Work is underway to identify higher order, natural features in the coral that will allow features to be matched for closing the loop in the SLAM algorithm. Recent work in the computer vision community has proposed a variety of methods for extracting distinctive features from images which can be used to perform reliable matching between different images of a scene. This work includes Maximally Stable Extremal regions [12] and Scale Invariant Features[9]. Both of these methods have been applied to selected images of the reefs and show good promise for data association under large changes in orientation and scale. Sensitivity to lighting conditions and turbidity remain to be investigated.

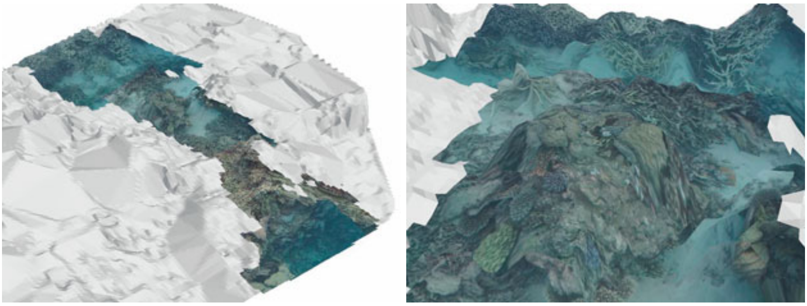
4.3 Results

This section examines results of the application of these techniques to tracking of the vehicle travelling over the reef. Figure 4 (b) shows the estimated path of the vehicle as it travels over the reef. Also shown are the principal returns identified in the sonar scans projected from the estimated vehicle poses. This path has been



(a) Oberon on the Reef (b) Vehicle Position Estimates

Fig. 4. (a) The Oberon vehicle operating on the Great Barrier Reef (b) The resulting SLAM estimates of the vehicle position have been used to position the sonar scans, resulting in this 3D terrain model.



(a) (b)

Fig. 5. The resulting terrain models built by projecting the texture of the visual images onto a surface model generated by the sonar data. a) In this view it is possible to see the path of the submersible where the textures have been projected onto the surface. The grey areas represent portions of the terrain only observed using the vehicle's sonar. b) In this view one of the coral bommy's can clearly be seen standing above the underlying sandy bottom.

generated without recourse to off-board positioning systems and relies solely on observations of the terrain structure using the sonar and visual systems as well as the vehicle gyroscope and pressure sensor readings.

During the current deployment, no positioning systems were available to validate the results of the SLAM algorithm. It is possible to examine the performance of the filter by studying the fidelity of the terrain model. Once the path of the vehicle has been estimated, the structure of the underlying reef can be reconstructed and the visual information can be projected onto the resulting surface. Figure 5 shows two views of these models. The correspondence in the data can be verified by examining the position of the image data relative to the structure of the surface. It is encouraging to note that many of the most significant bumps in the terrain structure appear to correspond quite closely with coral bommy's standing up from the sandy bottom.

5 Conclusions

This paper has shown preliminary results in the development of terrain aided navigation algorithms suitable for deployment in unstructured underwater environments. Models of the underlying reef structure have been generated based on estimates of the vehicle position generated by observing and tracking features on the sea floor. Subsequently, the image texture was projected onto these models to yield visual information regarding the structure of the reef.

Future work will concentrate on the development of techniques suitable for generating higher resolution terrain models and tying these models to *a priori*, lower resolution models. This will require the development of methods capable of tracking and identifying large numbers of points within the SLAM framework. Alternatively, higher order features or richer underlying models might be used to increase the efficiency of the tracking process. With very good visual clarity and highly textured surfaces, the Great Barrier Reef provides an ideal environment in which to develop novel navigation methods for use in unmanned underwater applications.

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. The Advanced Technology Centre of BAe Systems have also contributed generously towards the costs of deploying the vehicle in a field environment. Thanks also to David Lowe of UBC for providing access to the SIFT source to support this research work. Finally, thanks must go to the crew of the Undersea Explorer who provided an unforgettable experience aboard their vessel.

References

1. N. Bergman, L. Ljung, and F. Gustafsson. Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine*, 19(3):33–40, 1999.
2. J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardos. The SpMap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–952, 1999.
3. M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csobra. A solution to the simultaneous localization and map building (slam) problem. In *IEEE Transactions on Robotics and Automation*, volume 17(3), pages 229–241, 2001.
4. 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.
5. A.E. Johnson and M. Hebert. Seafloor map generation for autonomous underwater vehicle navigation. *Autonomous Robots*, 3(2-3):145–68, 1996.

6. D. Langer and M. Hebert. Building qualitative elveation maps from underwater sonar data for autonomous underwater navigation. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, volume 3, pages 2478–2483, 1991.
7. J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localisation for an autonomous mobile robot. In *IEEE/RSJ Intl. Workshop on Intelligent Robots and Systems*, volume 3, pages 1442–1447. IEEE/RSJ, 1991.
8. J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *Proc. Ninth International Symposium on Robotics Research*, pages 169–176. International Foundation of Robotics Research, 1999.
9. D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2003.
10. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.
11. S. Majumder, S. Scheding, 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.
12. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, pages 384–393, 2002.
13. P. Maybeck. *Stochastic Models Estimation and Control*, volume 1. Academic Press, 1982.
14. 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.
15. P. Newman. *On The Structure and Solution of the Simultaneous Localisation and Map Building Problem*. PhD thesis, University of Sydney, Australian Centre for Field Robotics, 1999.
16. V. Rigaud and L. Marc Absolute location of underwater robotic vehicels by acoustic data fusion. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, volume 2, pages 1310–1315, 1990.
17. 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.
18. 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.
19. 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 Internation Symposium on Robotics Research*, pages 346–353, 1999.
20. S.B. Williams. *Efficient Solutions to Autonomous Mapping and Navigation Problems*. PhD thesis, University of Sydney, Australian Centre for Field Robotics, 2001.
21. S.B. Williams and I. Mahon. Simultaneous localisation and mapping on the great barrier reef. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, 2004.
22. S.B. Williams and I. Mahon. A terrain-aided tracking algorithm for marine systems. In *Field and Service Robotics: Recent Advances in Research and Applications*, 2004, in press.

The SDR Experience: Experiments with a Large-Scale Heterogeneous Mobile Robot Team

Andrew Howard[1], Lynne E. Parker[2], and Gaurav S. Sukhatme[1]

[1] Robotics Research Laboratory
Department of Computer Science
University of Southern California
http://robotics.usc.edu/

[2] Distributed Intelligence Laboratory
Department of Computer Science
University of Tennessee
http://www.cs.utk.edu/∼parker/Distributed-Intelligence-Lab/index.html

Abstract. This paper reports on experiments conducted as part of the DARPA SDR (Software for Distributed Robotics) program. The core challenge for this program is to develop a system capable of carrying out 'locate-and-protect' missions: the system must be able to deploy a large number of robots into an unexplored building, map the building interior, locate a valued object, detect and track intruders, and transmit all of the above information to a remote observer/operator. To satisfy these requirements, we have developed a large heterogeneous robot team consisting of approximately 80 robots. This paper sketches the key technical elements of this system, and presents selected results from externally supervised experiments conducted in a 600 m^2 indoor environment.

1 Introduction

This paper describes a multi-robot system designed to meet a very strict set of requirements: to explore and map a single story in a large indoor environment, to detect a valued object, to deploy a sensor network and to use this network to track intruders within the building. This system must also operate autonomously, and employ as *many* robots as possible. These are the specific requirements imposed the DARPA SDR (Software for Distributed Robotics) locate-and-protect mission.

To meet this challenge, we have constructed a large robot team consisting of approximately 80 robots. Cost and power considerations dictate that the bulk of these robots should be relatively simple (i.e., with minimal sensor and computation capabilities). The robots must, however, retain the ability to make maps of the environment, detect valued objects, and navigate safely – tasks that would normally require the use of expensive ranging sensors such as scanning laser range-finders or stereo vision. To solve this apparent conundrum, we have adopted a heterogeneous approach, utilizing three distinct *classes* of robots with varying capabilities. The team therefore consists of: a small number of highly capable *mapping robots* equipped with scanning laser range-finders, cameras and unique fiducials; a small number of

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 121–130, 2006.

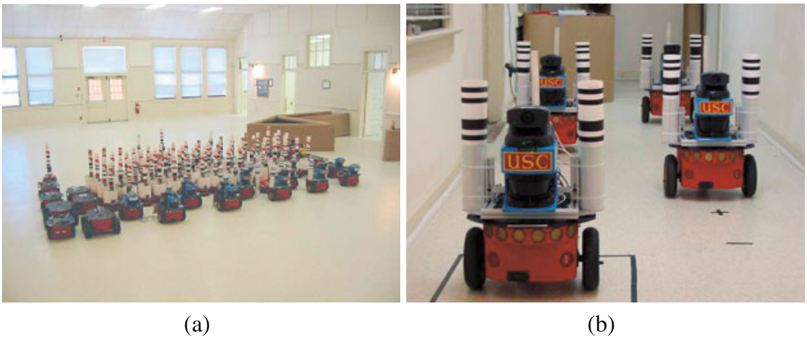


Fig. 1. (a) The heterogeneous robot team, with three classes of robots (mapper, leader and sensor); the sensor robots are based on the ActivMedia AmigoBot. (b) The mapping sub-team: note each of the four robots carries a unique laser-visual fiducial; these fiducials can be detected and identified at ranges in excess of 8m, and the relative range and bearing of the robots can be determined to within a few centimeters and a few degrees, respectively.

slightly less-capable *leader robots* equipped with scanning laser range-finders and cameras; and a large number (approximately 70) of simple *sensor robots* equipped with a microphone and a crude camera. All of the robots are have 802.11b WiFi, and a modified ad-hoc routing package (AODV) is used to ensure network connectivity.

With this heterogeneous team, the locate-and-protect mission is divided into two distinct phases: exploration and mapping, and deployment and detection. For the first phase, the mapping sub-team explores the environment, acquires an occupancy grid and locates the valued object. Exploration is coordinated, and mutual observations are used to solve difficult correspondence problems (the mapping robots observe one other, and thereby identify loops in the environment). In the second phase, the acquired map is used to compute a set of deployment locations, and the simple sensor robots are deployed to these locations using an *assistive navigation* technique. That is, since the sensor robots are not capable of navigating by themselves, they are guided into position by the more capable leader robots. Once deployed, the sensor robots collaborate to form a distributed acoustic sensor network that tracks audible targets moving through the environment.

In the sections that follow, we consider these two phases in greater detail. We also describe selected experiments used to validate the system; these experiments are distinguished by the fact that they were carried out under *external supervision*. Specifically, an independent government team selected and prepared the experimental site and supervised the conduct of experiments (monitoring completion times, user interventions, code modifications and so on). At no time during the experiments were operators allowed to enter the site, and prior access was limited to certain prescribed areas. Thus, at the outset, most of the environment was genuinely unknown to the human operators.

2 Exploration and Mapping

For exploration, we employ a fully decentralized frontier-based approach [1,10], using local occupancy grids and minimal communication between the robots. This exploration algorithm is fairly elementary, and will not be treated in detail here; we note only that the algorithm scales linearly with team size, ensures minimal interference between robots, and achieves asymptotic coverage of the environment.

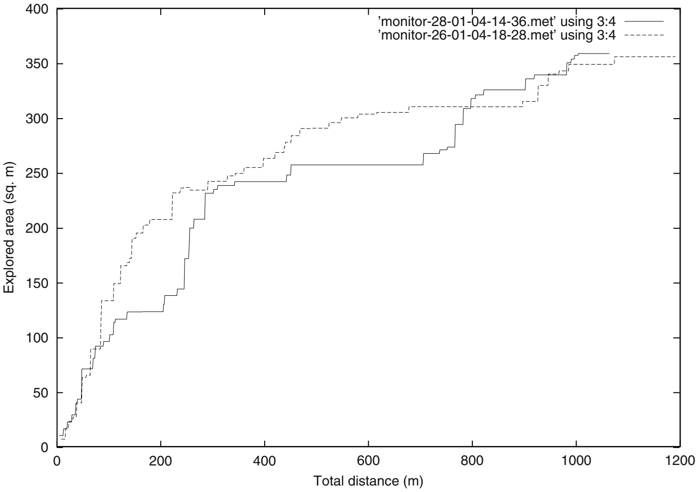
Some results from the final experiments are plotted in Figure 2(a). This plot shows the net area explored by a team of four robots as a function of total distance traveled (i.e., the sum of the distances traveled by all four robots). Note the rapid initial exploration, followed by slow convergence to full coverage; the latter is product of the strictly local nature of the algorithm: robots are not using global knowledge, and are not sharing information about previously explored areas. Faster convergence may be possible with the addition of global knowledge and/or coordination, albeit at the cost of increased system complexity.

In contrast with exploration, the mapping algorithm employs both centralized and decentralized components. Each robot in the team uses an incremental SLAM algorithm to maintain an independent local pose estimate; laser scan data is used to correct most, but not all, of the odometric drift. These independent local pose estimates (along with the raw laser scan data) are transmitted to a remote console, where they are combined and corrected to form single map of the environment. We will consider each of these two components in turn.

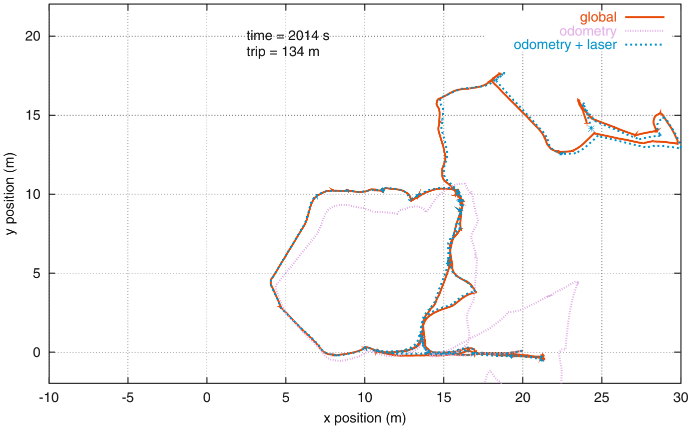
For estimating local pose, each robot employs an incremental maximum likelihood filter [9] that is similar in spirit – if not in detail – to that described in [3]. The state vector for this filter consists of two components: a local pose estimate and a local map. Roughly speaking, the local map is simply a finite-length queue containing recent laser scan data; each incoming scan is fitted against this local map (using a dense scan-matcher), thereby inducing an updated pose estimate for the robot. This estimate will, of course, drift over time (an unavoidable feature of every incremental method); the drift *rate*, however, is one or two orders of magnitude slower than that seen with odometry alone. See, for example, the results plotted in Figure 2(b): while the odometry estimates quickly diverge, the error in the local pose estimate is less than less 0.1 m after 100 m of travel.

There is no attempt by individual robots to close loops or merge data from other robots; this is the role of the remote console, which aggregates local pose estimates and range scans from each of the mapping robots into a central repository. The overall map is assembled from this data using a global SLAM algorithm that is built around three key technologies: maximum likelihood estimation, manifold representations and loop closure using mutual observation.

Maximum likelihood estimation (MLE) is used to generate globally consistent maps; put simply, MLE determines the set robot trajectories that minimizes the inconsistency between overlapping laser scans [5]. In practice, this is a high-dimensional optimization problem that must be solved using sparse graph-based representations and numerical optimization algorithms (modern algorithms allow simultaneous optimization over hundreds of variables and thousands of constraints). In this context,



(a)



(b)

Fig. 2. (a) Coverage plot for four robots: the plot shows the area explored by the robots as a function of the total distance traveled; coverage is determined by a post-hoc analysis of the data, once a complete map is available. (b) Comparison of robot pose estimates using odometry and incremental SLAM; the incremental pose error is less that 0.1 m after 100 m of travel. Note that the ground-truth trajectory is determined post-hoc using a global localization algorithm and a complete map of the environment.

it should be noted that the use of local pose estimates in the place raw odometry greatly simplifies the optimization problem, as the local and global estimates diverge comparatively slowly.

For this project, we have made one important extension to the basic MLE formalism: instead of treating the map as a two-dimensional planar structure, we represent

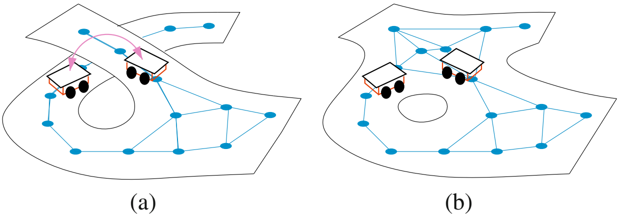


Fig. 3. A manifold map representation, before and after loop closure. The manifold is a self-consistent – but redundant – representation, in which the same point in the world may appear more than once on the manifold. (a) Robots at remote locations in the manifold observe that they are proximal in the world (i.e., the robots detect one another using their sensors). (b) This information is propagated through the manifold, allowing the loop to be closed.

it using a two-dimensional *manifold* [4]. Unlike planar representations, the manifold representation is always *self-consistent* irrespective of whether or not loops have been closed. To achieve this self-consistency, the manifold representation must sacrifice *uniqueness*; i.e., a single location in the world may be represented more than once in the manifold (see Figure 3). Thus, in this representation, loop closure is reduced to the problem of identifying and bringing together those points on the manifold that represent the same point in the world.

To recognize such points, we make use of mutual observations: if two robots are far apart on the manifold, but are proximal in the world (i.e., they can see one another), we can infer a new set of correspondence points and close the loop. It should be noted that this active, multi-robot approach to loop closure entirely side-steps the hard correspondence problems that bedevil global SLAM algorithms.

Exploration and Mapping Results

Five exploration and mapping trials were performed under government supervision, all of which resulted in successful map generation. The first three trials were performed with identical software and initial conditions; the final two trials were performed under 'challenge' conditions designed to test the system envelope.

The most difficult of the challenge conditions involved a team of four mapping robots deployed from two different locations (i.e., two robots from each of two entry points). Since the relative pose of the two entry points was unknown, each pair of robots was required to explore and map independently, giving rise to two unconnected maps or 'islands'. After approximately 10 minutes, however, the two teams encountered one another, and as a result of this mutual observation, the two maps were merged into one. The combined robot team then proceeded to complete the exploration task, yielding the final occupancy grid map shown in Figure 4. It should be emphasized that this exploration and mapping process was entirely autonomous, with the exception of a single user intervention to direct a robot into the otherwise unexplored room at the bottom left of the map (this intervention was

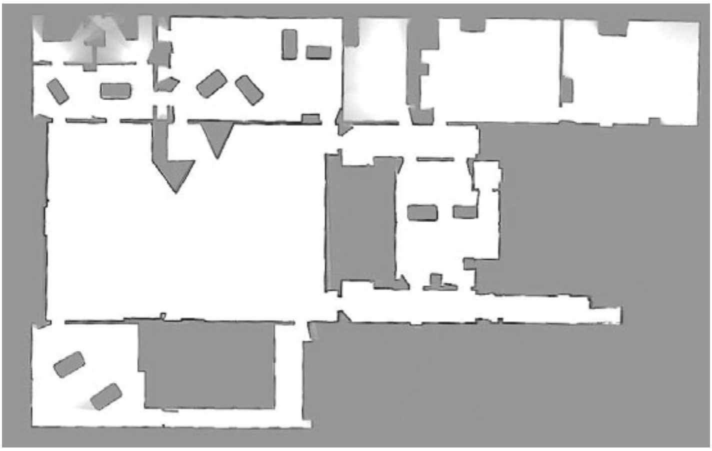


Fig. 4. Occupancy grid map produced during a multiple-robot, multiple-entry trial: two robots entered from the door at the right top, another two robots entered from the door at the right bottom (the relative pose of the two doorways was unknown). The environment is approximately 45 by 25 meters in size, with an internal area of 600 m^2.

not strictly necessary, was performed to achieve faster map completion). Maps were generated in real time, for an environment approximately 600 m^2 in size. Map accuracy is comparable to or better than that achieved by a human survey team using a tape-measure, pencil and paper.

3 Deployment and Detection

The intruder detection and tracking task requires the deployment of a large number of simple sensor robots. These robots have low-fidelity microphones to serve as a distributed acoustic sensor network, but no sensors for localization or obstacle avoidance. Because they cannot navigate safely on their own, our approach provides cooperative assistive navigation to the sensor robots through the use of more capable leader robots. Using the map generated in the first phase, these leader robots are able localize themselves and guide the sensor robots to their deployment positions. The basic method is as follows: sensor robots are assembled into chains behind a leader using simple color blob tracking (multi-robot follow-the-leader); once a deployment destination is reached, a single robot in the chain is autonomously 'tele-operated' by the leader to the correct position, using the camera mounted on the leader; the leader and the remaining chain then proceed to the next deployment position. Thus, the leader visits a series of locations in turn, and deploys a single sensor robot at each. Figure 5 shows a series of snapshots of our navigational assistance system in operation. Details of this deployment strategy are reported in [7].

With this heterogeneous cooperative approach, it is important to develop a thoughtful deployment strategy to ensure that the sensor robots will be positioned in the desired distribution. An especially challenging problem is for the leader robots to follow paths that the simple robots can follow, to ensure that the sensor robots reach their desired positions while moving in a follow-the-leader formation. Obviously, the navigational challenges grow if the leader robots move to random sensor node deployment positions without taking into account the formation of robots that is following behind. Our approach begins with four planning steps. Step 1 generates the planned sensor deployment positions to meet several criteria, including minimizing pathway obstruction, achieving a minimum distance between sensor robots, and maximizing visibility coverage. Step 2 then generates the waypoints of the path that the leader robot must follow to guide the sensor robots to the vicinity of the sensor deployment positions. Figure 6 shows a set of autonomously planned sensor and leader waypoint positions for the earlier map. Since each leader robot can only deploy a few sensor robots at a time, and since several leader robots are available to operate in parallel, Step 3 divides the sensor deployment positions into groups to facilitate the deployment operation. Each group of positions is assigned to a team (consisting of one leader robot and several sensor robots) for deployment. The team assignments are generated to achieve the objectives of: (1) minimizing travel by the deployment teams, (2) minimizing interference between teams, and (3) minimizing the amount of turning a team must perform as it travels to all of its assigned deployment positions. The traffic management of Step 4 incorporates deployment delays to coordinate multiple deployment teams operating in parallel, similar to [2]. Details of Steps 1-3 of the planning process are reported in [8].

Once the acoustic sensor robots have been deployed, they activate their distributed acoustic sensor net to detect targets that are moving through the environment. Figure 7 shows these robots deployed in the planned sensor net positions and acting as this sensor network. Since our sensor robots are equipped with only a very simple microphone, we assume that the target moving through the environment is making some detectable noise, and that the target is the only source of sound. Our approach involves each robot filtering its acoustic data and then communicating its volume heard to its local neighbors. The robot that detects the highest volume in its neighborhood (above a threshold) reports its own position as the current target position estimate to the operator control unit. Details of this distributed acoustic sensor network process are reported in [6].

Deployment and Detection Results

In three separate trials supervised by the government team, our distributed acoustic sensor network achieved 100% detection of targets in the environment (all targets we localized to the correct room or corridor) with no false positives; these trials involved 3 leaders and up to 35 sensor robots. Clearly, these results show that the planned sensor positions and the distributed acoustic sensor processing worked well for the objectives of these experiments.

Another metric of evaluation is the percentage of sensor robots successfully deployed (i.e., the ratio of successful deployments to attempted deployments). Our experimental data shows an overall deployment success rate of 60% - 90%, depending upon the environmental characteristics. In other words, for each attempt at deploying a simple robot, 60% - 90% of those robots successfully reached their planned deployment position. The reason for the low end of this success rate is the complexity of our heterogeneous robot system. Our system for simple robot deployment is composed of several non-trivial modules, including localization, path planning, navigation, leader following, visual marker detection, and inter-robot communication. The successful completion of the entire deployment process depends upon the successful completion of all of the system modules while the robots are operating in cluttered environments along complex paths. Additionally, the independent experimentation reported here was especially challenging because we forced the robot team to deal with the consequences of prior deployment failures. Thus, subsequent robot team deployments had to deal with situations such as partially blocked doorways if a prior deployment resulted in a simple robot being caught on the doorway. If all the test runs had been independent, the overall system success rate would certainly have been higher.

Clearly, there are many potential failure modes in such a complex heterogeneous system involving such a large number of robots. The most common failure modes of the system were caused by variable lighting conditions (which could cause the sensor node robots to lose the color blobs, or the leader robots to lose the visual marker for autonomous tele-operation), cluttered environments (which could cause the follower sensor node robots to lose the leader robot amidst many navigational twists and turns), and communications failures (due to delays in multi-hops in the wireless ad-hoc network). To account for these potential subsystem failures, we built extensive fault tolerance into the behavior of the leader robot. Table 1 shows the set of base failure states identified for this system and the implemented recovery action. Using these methods of behavior fault tolerance, the success rate of the leader robots making it back home autonomously in these rigorous experiments was 91% (over 45 trials).

Table 1. Failure states detected by the leader robot and implemented recovery actions.

Failure Type	Fault Recovery Action
Can’t reach waypoint	Re-plan path.
Lost simple robot	Leave lost robot in wait state and move on to next robot in chain.
Leader robot camera failure	Leave simple robot(s) in wait state, send camera failure feedback to human operator and return home.
Simple robot motor failure	Check if simple robot is close to goal; if so, change simple robot state to sensor detection and proceed; else, leave simple robot in wait state and proceed.
Localization drift	Check if simple robot is close enough to goal; if so, change simple robot state to sensor detection and proceed; else, leave simple robot in wait state and proceed;
Lost marker	Leave simple robot in wait state and move on to next robot in chain.
Communication failure	Return back home.

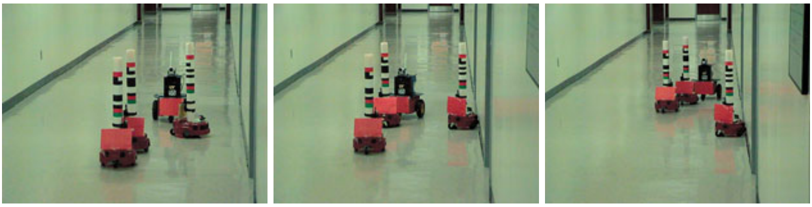


Fig. 5. Deployment of a sensor robot using assistive navigation: the lead robot first guides and then directs the sensor robot into position.

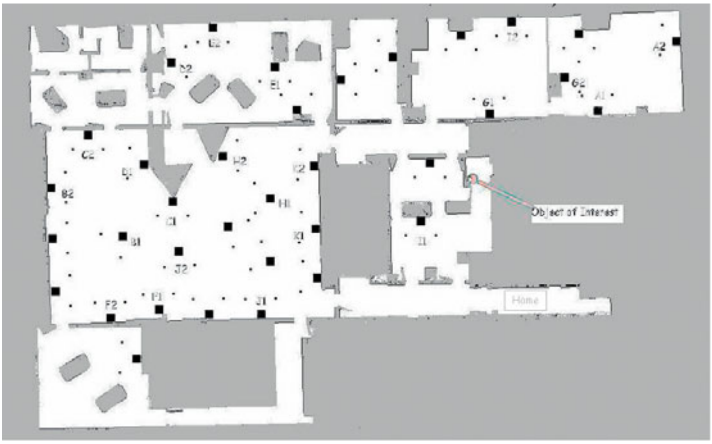


Fig. 6. Autonomously planned sensor net positions (black squares) and planned leader waypoints associated with each sensor position (small dots).

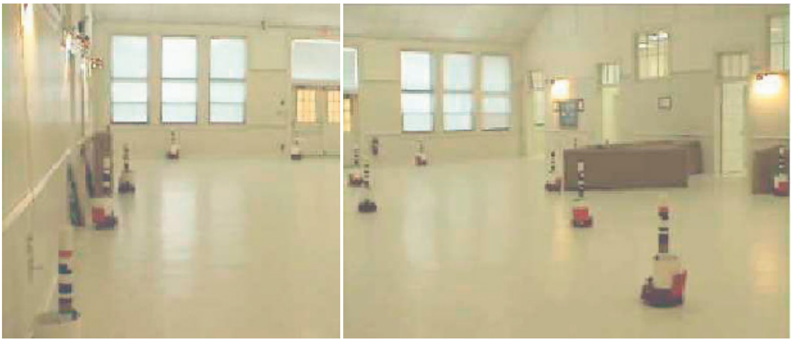


Fig. 7. Physical robots deployed according to the results of the autonomous planning process for the environment shown in Figure 6.

4 Conclusion

In our view, the SDR project has established two new benchmarks in the field of multi-robot systems: this is by far the most complex task ever demonstrated by a large heterogeneous robot team, and by far the most rigorous set of experimental conditions. It should also be noted that while much of our work on this project was necessarily focused on robust engineering, we are confident that the products of this engineering (such as algorithms for multi-robot mapping) will quickly find broader application within the robotics community.

Acknowledgments

This work is sponsored in part by DARPA grant 4400057784/SDR (Software for Distributed Robotics) in conjunction with SAIC, Telcordia, the University of Southern California and the University of Tennessee Knoxville.

Particular thanks go to John Spofford for his patience and understanding, and to Eric Krotkov and Doug Hackett for their extraordinary efforts in the preparation and supervision of the Fort AP Hill experiments.

References

1. W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proc. of IEEE International Conferenceon Robotics and Automation (ICRA)*, volume 1, pages 476–81, 2000.
2. Y. Guo and L. E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2002.
3. J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedingsof the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2000.
4. A. Howard. Multi-robot mapping using manifold representations. In *IEEE International Conference on Robotics and Automation*, pages 4198–4203, New Orleans, Louisiana, Apr 2004.
5. F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.
6. L. E. Parker, B. Birch, and C. Reardon. Indoor target intercept using an acoustic sensor network and dual wavefront path planning. In *Proceedings of IEEE International Symposium on Intelligent Robots and Systems (IROS '03)*, October 2003.
7. L. E. Parker, B. Kannan, F. Tang, and M. Bailey. Tightly-coupled navigation assistance in heterogeneous multi-robot teams. In *submitted to IROS '04*, 2004.
8. Y. Tang, B. Birch, and L. E. Parker. Mobile sensor net deployment planning using ray sweeping and wavefront path planning. In *IEEE International Conference on Robotics and Automation (ICRA '04)*, 2004.
9. S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
10. B. Yamauchi. Frontier-based approach for autonomous exploration. In *Proceedings of the IEEE International Symposium on Computational Intelligence, Robotics and Automation*, pages 146–151, 1997.

Centibots: Very Large Scale Distributed Robotic Teams

Kurt Konolige, Dieter Fox[1], Charlie Ortiz, Andrew Agno, Michael Eriksen, Benson Limketkai[1], Jonathan Ko[1], Benoit Morisset, Dirk Schulz[1], Benjamin Stewart[1], Regis Vincent

SRI International, Artificial Intelligence Center
[1]University of Washington, Department of Computer Science & Engineering

1 100 Robots

We describe the development of Centibots, a framework for very large teams of robots that are able to perceive, explore, plan and collaborate in unknown environments. The Centibots team currently consist of approximately 100 robots (Figure 1). The Centibots team can be deployed in unexplored areas, and can efficiently distribute tasks among themselves; the system also makes use of a mixed initiative mode of interaction in which a user can influence missions as necessary. In contrast to simulation-based systems which abstract away aspects of the environment for the purposes of exploring component technologies, the Centibots design reflects an integrated end-to-end system.

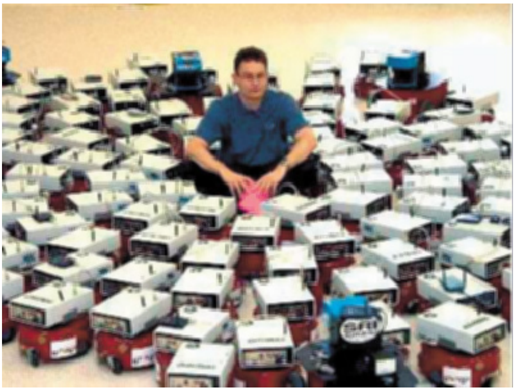


Fig. 1. 100 robots. Four of the robots are Pioneer IIs with SICK laser range-finders. The rest are Amigobots with sonars, a camera and a small PC on top. The OOI is in the hand of one of the authors.

As part of DARPA's Software for Distributed Robotics (SDR) project, the Centibots were tested on a mapping and search mission in a new, unknown environment. This experiment involved deployment of Centibots in three successive stages: (1) a *mapping stage* for the coordinated exploration of the environment while simultaneously constructing a very high accuracy occupancy map using a laser range finder; (2) a *search stage* in which the environment is exhaustively searched for a predefined object of interest (OOI), chosen so that it could be easily distinguished within the environment by its shape and its color; and (3) an *intruder detection stage* in which robots are distributed throughout the environment to "guard" the OOI by continuously searching the environment for human intruders. This stage included recharging a portion of the robots to prove the system could continue indefinitely.

Previous work has largely focused on isolated aspects of our system, including multi-robot exploration [1], architecture [3], task allocation [12], coordination [9],

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 131–140, 2006.

and human interaction [11]. Here we describe the integration of various technologies to achieve an operational robot team, which was tested under rigorous conditions by SDR's outside evaluation team during a final demonstration. The main criteria of the evaluation focussed on the effectiveness of the robot team in performing the mapping and surveillance task (Sections 2.2 and 3.3).[1]

Our approach to multirobot coordination is significantly different between the mapping phase and the subsequent search and surveillance. Mapping is performed with a small number (1-5) of robots working completely autonomously, often out of contact with the base station. Their interactions are tightly focussed on solving a single task, exhaustively mapping an area in the shortest time. We developed specialized algorithms based on utility theory to coordinate the mapping robots, under the condition of an unknown environment, intermittent communication and no centralized planner. In search and surveillance, a much larger number of robots ($\approx$100) must be coordinated, and the tasking is more flexible, e.g., robots can be commanded to watch over a given area. Here, issues of spatial reasoning, task distribution, resource allocation, and user interaction become much more important.

In the following sections, we describe the coordination strategy for mapping and exploration, and give the results of the evaluation for this phase. We then give an account of search and surveillance, along with their results.

2 Distributed Mapping and Exploration

In the mapping phase, multiple robots explore the environment in order to build a map that can be used in the subsequent search and surveillance phases. We developed a decentralized system that goes beyond the state of the art in multi-robot mapping in that it does not depend on reliable communication between robots and makes no assumptions about the robots' relative start locations.

2.1 Overview of Exploration System

Multi-robot Mapping Our technique for multi-robot mapping is based on a representation of *local probabilistic constraints* among robot poses. These constraints arise from robot motion (odometry) and matching laser rangefinder scans. Figure 2 shows a laser map along with the trajectory of a robot (gray) and the constraint links (black). The trajectory also represents robot motion links. The optimal position of the poses is the one that maximizes the posterior probability of all the constraints. Although the constraints are nonlinear, there are efficient approximations that work well in practice [4, 7, 8]. Note the fine detail of the laser scan map resulting from this optimization, showing even the thickness of the walls.

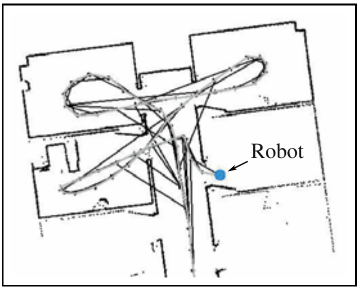


Fig. 2. Links among poses. Gray lines show 1m grid for size.

[1] Two other teams, one led by SAIC and one by MIT, also underwent the same evaluation process, but as of this writing we do not have access to their results for comparison.

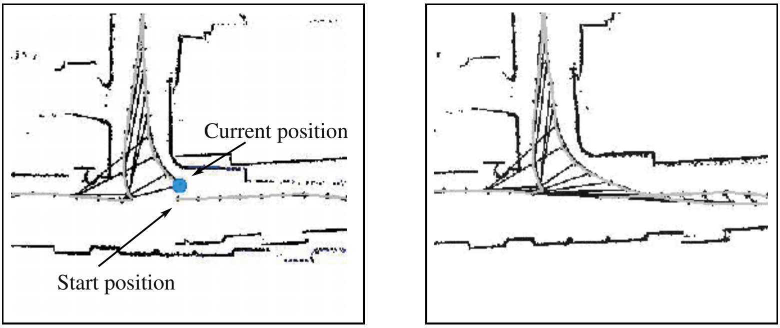


Fig. 3. Pose constraints before (left) and after (right) linking the start and end of a loop.

The constraint network is ideal for integrating map information with uncertain alignment. Consider first the case of closing a loop: a robot returns to a position it has previously visited, but accumulated error causes it to be misaligned (Figure 3, left). Here the robot has traversed an interrupted loop, going out of the top of the figure before coming back. In the right side figure, scan matching has established links with poses at the beginning of the loop, and by optimization the loop can be closed correctly.

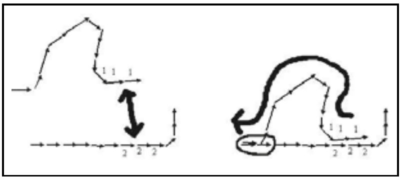


Fig. 4. Zippering two unregistered partial maps. Poses "1" are found to be related to poses "2". After joining, the new poses are added to the first map, starting from the join point; note the additional loop closure (circled region).

The constraint representation naturally facilitates the merging of partial maps built by different robots. For example, the left side of Figure 4 shows the robot poses of two partial maps built independently by two robots, without any notion of where they are with respect to each other. Suppose we can link poses in one map (labeled "1") to poses in the other map ("2"), by some good decision process. Then, we can move the two maps together to register them in the same metric space. Finally, we go through one of the partial maps and add all of its scans to the other map, just as if all scan were collected by a single robot. In this process, the two maps are *zippered* together, adding connections resulting in a globally consistent map.

Abstractly, the zippering process lets us take any partial maps produced by any robots and put them together, once a common location (*colocation*) between their trajectories has been identified (note that colocation is transitive). In order to determine these common locations, we developed an efficient algorithm that sequentially estimates the relative locations between robot pairs as they explore an environment [6]. The approach considers only pairs of robots since the complexity of estimating map matches is exponential in the number of robots considered jointly. For each robot pair, the technique uses an adapted particle filter to estimate the position of one robot in the other robot's partial map. By estimating the posterior over robot positions both inside and outside the partial map, the approach is also able to estimate whether or not there is an overlap between the robots' maps. To accurately determine the overlap probability, we developed a hierarchical Bayesian technique that learns a prior over the *structure* of indoor environments and uses the structural model to estimate the certainty of map matches [2, 10].

Active Multi-robot Colocation and Exploration Virtually any map matching technique can generate false-positive matches, especially in large, highly symmetric environments. A wrong map match between two robots can generate subsequent, wrong map matches with other robots. Thus, undoing a wrong match requires considering all other map matches as well. To avoid this complexity, we developed a technique that coordinates robots to *actively* verify whether or not a map match hypothesis is correct. The approach is integrated into a decision-theoretic, frontier based multi-robot exploration strategy (see [6] for details).

Figure 5(b) shows an example run using our coordination technique. The two robots, A and B, start from different, unknown locations. Initially, the robots explore on their own. As they explore, each robot estimates the other robot's location in its own map, using the modified particle filter mentioned above. When deciding where to move next, both A and B consider whether it is better to move to an unexplored area (frontier), or to verify a hypothesis for the other robot's location. At one point, B decides to verify a hypothesis for A's location. It sends A the message to stop and moves to A's hypothesized location. Upon reaching this location, both robots check the presence of the other robot using their laser range-finders (robots are tagged with highly reflective tape). When they detect each other, their maps are merged using the zippering process described above. From then on, they explore the environment in a coordinated way. If a hypothesis verification fails, on the other hand, then the hypothesis is simply deleted, and all robots keep on exploring.

Our coordination technique works for more than two robots. Multiple robots can share a common map and coordinate to explore and verify hypotheses for the locations of other robots. Since each map merge operation increases the number of robots sharing a common map, team coordination improves over time. Howard et al. [5] also use robot detections to merge maps (and close loops). In contrast to our active colocation technique, their approach is purely passive in that robots have to detect each other coincidentally. Passive map merging can result in significant delays, for example, when one robot follows the path of the other robot and never actually detects it.

Exploration with Limited Communication Robots form so-called exploration clusters, which are groups of robots that share a common map. A team leader robot uses this map to coordinate the other robots. New robots can be added to a cluster, once their relative location w.r.t. the cluster map is determined. Maps are represented compactly as sets of laser range-scans annotated with robot poses and probabilistic links (scans are recorded only every 50cm). Each robot integrates its observations into its own map, and broadcasts the information to the other robots. While most of the other robots only store this data, the team leader integrates all the sensor information it receives. Thus the team leader has a complete and consistent map representing the data collected by all robots in the cluster. Frequently, this map is broadcast to the other robots, in order to guarantee consistency. The data can be sent

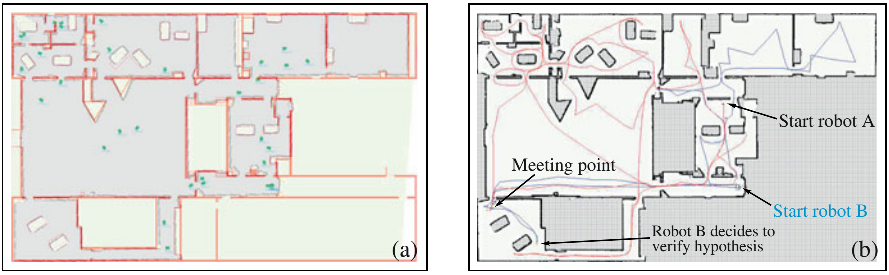


Fig. 5. (a) Map overlayed with the ground truth CAD model of the building. The CAD model was generated by manually measuring the locations and extensions of rooms and objects. (b) Map and paths of an exploration run. The two robots start exploring from different, unknown locations. After finding a good hypothesis for their relative locations, they meet at the meeting point, merge their maps, and continue coordinated exploration.

very compactly, since only updated robot poses and links have to be transmitted (scans are already stored by the other robots). The most complex broadcast follows whenever a robot closes a loop, since the optimization operation modifies all robot poses in a map.

Our exploration system achieves robustness to communication loss by enabling every robot to explore the environment on its own. Whenever a robot in an exploration cluster reaches an assigned goal point, it keeps on exploring based on its own map until it receives a new goal point. Thus, if a robot moves outside the communication range of its cluster, it automatically keeps on building its own map until it gets back into communication range. After getting back into communication, robots exchange all the relevant data that was lost. Such a ’sync’ operation only involves the communication of rather small data sets. The approach is also robust to loss of the team leader, since any other robot in the cluster can explore on its own or take over the team leader role. In the extreme, if none of the robots can communicate with each other, each robot will explore the environment independently of the other robots. The result will still be a complete map; only built less efficiently.

2.2 Experimental Evaluation of Mapping

The SDR project is unique in having an experimental validation conducted by an outside group. For a week in January 2004, the Centibots were tested at a $650m^2$ building in Ft. A.P. Hill, Virginia. We were tested under controlled conditions, with a single operator in charge of the robot teams via the mixed initiative interface described in the next section. The evaluation criteria for mapping included time to create a map, topological accuracy, and percent of area mapped. Ground truth for mapping was given by a manually constructed map (Figure 5(a)); in fact, the robot’s maps were more accurate. Extensive software tuning was circumvented by limiting access to only half of the experimental area during test runs.

The results for four official mapping runs are summarized in Table 1. In all runs, the robots were able to autonomously generate a highly accurate map of the whole environment. The average mapping time for single robot exploration was 24 minutes; this time was reduced to 18 minutes when using two robots. We also

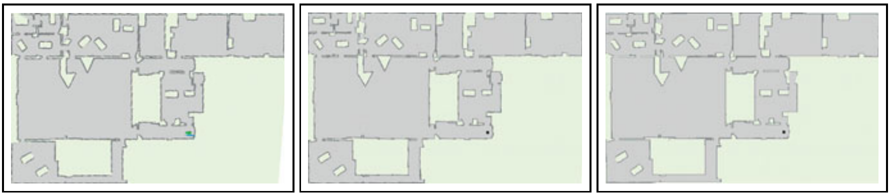


Fig. 6. Maps built during three autonomous exploration runs. The maps look almost identical, even though they were built under very different circumstances; (left to right) by one robot, by two robots starting from the same location, and by two robots starting from different, unknown locations. The similarity between the maps illustrates the robustness of the system and supports our belief that these maps are more accurate than the hand-built map.

performed some more experimental runs. In one setup, three robots were able to map the area in 15 minutes. Two robots starting from different, unknown locations generated a complete map within 26 minutes (this run is shown in Figure 5(b)). All generated maps looked virtually identically, as shown in Figure 6.

3 Search and Surveillance

In this phase of the mission, the challenge of coordinating the robots becomes more difficult because of the following factors.

- The number of robots is large (≈100).
- The mission goals are not predetermined, and can change during the mission.
- The goals should be specified at a high level, e.g., “search the building.”
- The robots must establish and maintain a communications network.
- All robots must be controlled and coordinated by a single operator.

In Centibots there were only two high-level missions: the search for OOI and the protection of the OOI. In both cases the first step was to determine *where* to send the robots to achieve the mission: how the robots can cover the most free space while maintaining other objectives such as a communication backbone. In the Centibots system, a spatial reasoner determines an optimal or near-optimal assignment of robots to positions in the space, while a separate coordination module (the *dispatcher*) implements an assignment that follows the plan, and also monitors and adapts the plan as robots fail.

Many interesting low-level behaviors are built into the search robots, allowing the coordination module to abstract away some of the difficulties of the problem. For example, robots have the ability to stay localized within a map, to navigate to particular positions within the map, and even to perform simple traffic-control behavior such as staying to the right in a corridor. In searching for the OOI, there are behaviors to scan in a circle, to detect the object, and to transmit information about its location. Finally, each robot has a single camera, and algorithms for detecting people as moving objects and reporting their position.

3.1 SPARE

The SPARE component (for SPAtial-REasoning) is a general system dedicated to the spatial allocations in Centibots. SPARE is structured in two parts: spatial representation generation and spatial reasoning.

Occupancy Map → Skeleton Map → Spatial Reasoning

The output of the mapping phase of the mission is an occupancy map, specifying occupied, unoccupied, and unknown regions. From this, the representation generator constructs an abstract skeleton map of nodes and connections between the nodes. The skeleton map is used by spatial reasoning processes to search for robot locations that maximize their utility for a given task.

Spatial Representation The spatial representation creates an abstraction of the map, called the *topological graph* (*TG*), that has the following characteristics:

- It is topologically correct.
- It is compact.
- All points are reachable by the robots.
- It can be used for a quick but correct path planning computation.
- It contains enough points to find good solutions, but not too many to make the search computationally unfeasable.

To create the *TG*, we first compute a Voronoi diagram (*VD*) from the input occupancy map (a 2D grid). *TG* is then generated from *VD* by filtering, feature identification, and expansion. Figure 7 shows the some steps in the generation of *TG*. In general, *TG* consists of a set of nodes and links that cover the area in a manner that facilitates searching and visibility.

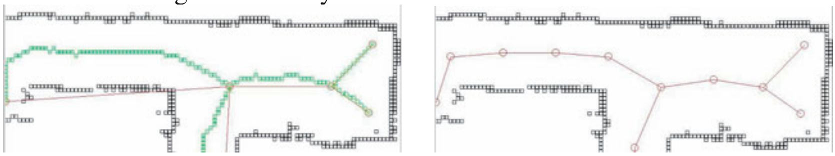


Fig. 7. *VD* components associated with vertices and edges of *TG*, and additional vertices added by the expansion step.

Spatial Reasoning For any given mission, we abstract the problem of placing robots in the environment to the task of assigning robots to nodes of *TG*. We assume all the robots are homogenous, so any robot can fulfill any task requirement. An *assignment* α is a mapping from the nodes of *TG* to 1 or 0 (robot or no robot). It would be a simple modification to change the boolean value to a set of values to account for *classes* of robots.

The *cost* $C(\alpha)$ is a scalar function of the assignment. Since the number of assignments is exponential in the number of nodes, we decompose the cost into subcosts that can be easily calculated, and use an approximate method to determine a good assignment (Section 3.1). In general, we want costs to be local to a single node, or at least to a small neighborhood of nodes, so that incremental optimization algorithms will work well. To this end, we determine the global cost by a summation of smaller cost functions:

$$C(\alpha) = \sum_{i=0}^{p} w_i \sum_{j=0}^{n} c_i(v_j; \alpha) \tag{1}$$

where n is the number of nodes in *TG*, and p is the number of cost functions. The w_i are *weights* that can be changed to reflect the type of mission under consideration. The weights were chosen empirically, to reflect the different priorities in the two missions of searching for the OOI and protecting the OOI.

Note that, potentially, each "local" cost function c_i could involve the whole assignment. In practice, we have developed 10 local cost functions, of which 8 relate to a single node, 1 to a local neighborhood, and one which tries to minimize the number of robots, and so uses the whole assignment.

Finding Good Assignments The problem is to identify the n-tuple $(v_1, v_2, ...v_n)$ that optimizes C with a weight distribution w_i specific to the mission. The search space size is 2^n where n is the number of vertices in *TG* (typically, a few hundred). In such a huge search space, and in the context of our application, our goal was not to find an optimal solution but a satisficing one in a reasonable time (within a few minutes). The quality of a solution is determined by a human expert: a solution is considered good if no misallocation is detected by a human analysing the result. We have found that a standard simulated annealing algorithm works well in finding reasonable solutions.

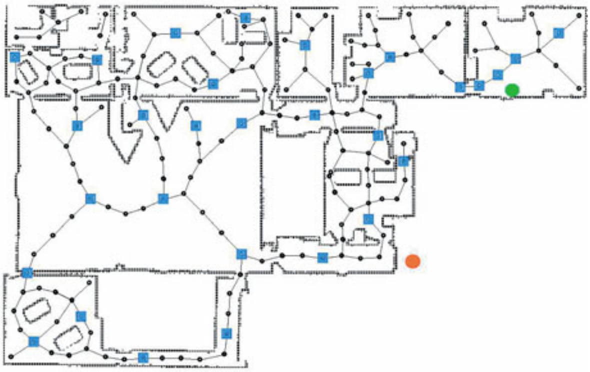


Fig. 8. *TG* shown with superimposed assignment (squares) for a guarding operation. The OOI is in the upper right room; the operator is at the lower right. All rooms are covered for intruder detection, and the comms backbone has a max distance of 10m per robot.

The SPARE system has been tried on over a dozen different maps for several types of tasks. For each map, the computation time does not exceed a few hundred milliseconds. For a given task, the difficulty lies with manual tuning to obtain suitable values for the weight distribution w_i. This tuning is done by trial and error. In practice, once a weight distribution is found for a given task, the weight distribution stays efficient on other maps. For instance, for the searching and protecting tasks performed at the demonstration site in a completely unknown environment, no new tuning of the weight distributions were necessary. Even if the optimality of the solution returned by SPARE is impossible to evaluate formally, the cost diminution between the initial random solution and the final one is considerable (by a factor of 100 on average). Figure 8 shows the *TG* and a guarding assignment for one of the evaluation runs.

3.2 Hierarchical Dispatching

Once we have all the goals generated by SPARE, the operator assigns a mission to a number of robots. The operator can refine the assignment by creating sub-teams to achieve part of mission independently, or let the system coordinate the entire pool of robots. For example, one team could be assigned to create a communications backbone for the search task. Within a team, a manager or *dispatcher* is chosen to assign tasks to individual robots. Robots register with one or more dispatchers and receive an assignment from the set of goals available to the team. A robot can register to several dispatchers, with one preferred. If there are no more goals to be done for the preferred dispatcher, the robot will ask other dispatchers for work, which is a way of load-balancing. When the goal is finished, the robot calls back in to report the completion, says that it is available, and requests a new goal from the dispatcher. The dispatcher can also allocate a set of goals at once for a robot to do, and the robot will only re-contact the dispatcher when it has finished all of them. The dispatcher is designed as a network service that resides physically anywhere on the network. It can run on any team member and only requires local communication within the team. In every experiment with Centibots, the operator only formed one team, therefore our hierarchical dispatching was equivalent to a centralized approach. This approach was quite effective because the team was limited in size (less than 50).

3.3 Experimental Evaluation of Search and Surveillance

For searching, the evaluation criteria were time to locate OOI(s), positional accuracy, and false detections. For the protection stage, the criteria were detection of intruders, and time to first detection. There were four evaluation runs, and the results are shown in the Table 1. They show that the team was highly effective in finding the object and setting up a guard perimeter. Note that we used very simple visual detection hardware and algorithms, since we had limited computational resources on the robots – false and missed detections were a failure of these algorithms, rather than the spatial reasoning and dispatching processes.

Table 1. Results of the 4 evaluation runs.

Run	Mapping Time	Map Area	Search Robots	Search Time False Pos	Position Error / Topo Error	Intruder Detect	Time to Detect
1	22 min	96%	66	34 min / 0	11 cm / none	75%	8 sec
2	26 min	97%	55	76 min / 1	24 cm / none	50%	8 sec
3	17 min (2 robots)	95%	43	16 min / 0	20 cm / none	25%[2]	8 sec
4	19 min (2 robots)	96%	42	Missed / 2	NA	100%	48 sec
Avg.	21 min	96%	51	30 min / 0.75	14 cm / none	62%	18 sec

[2] Caused by a misconfigured tracking filter, fixed before the next run.

4 Conclusions

It is a measure of the state of maturity of mobile robotics that it is possible to field and evaluate a larg teams of robots in the short 18 months of this project. Our Centibots team uses inexpensive COTS mobile robots, cameras, and processors; the algorithms we have developed for mapping, planning and mixed-initiative deployment will work in real-time, in a distributed fashion, with unreliable communications, in an unknown environment. These are some of the toughest real-world conditions yet imposed on a robotics project. We believe that evaluations like this one are an important step to moving AI robotics into the real world.

Acknowledgements

We thank Doug Gage and DARPA for support under the Software for Distributed Robotics Program (Contract #NBCHC020073). We would also like to thank the evaluation team, Erik Krotkov and Douglas Hackett, for their extraordinary efforts in designing and running the demonstration.

References

1. W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. 2000.
2. D. Fox, J. Ko, K. Konolige, and B. Stewart. A hierarchical Bayesian approach to mobile robot map structure learning. In *International Symposium of Robotics Research*, 2003.
3. Brian P. Gerkey and Maja J Matarić. Multi-Robot Task Allocation: Analyzing the Complexity and Optimality of Key Architectures. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, Taipei, Taiwan, May 2003.
4. J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1999.
5. A. Howard, L.E. Parker, and G.S. Sukhatme. The SDR experience: Experiments with a large-scale heterogenous mobile robot team. In *Proc. of the International Symposium on Experimental Robotics*, 2004.
6. J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical, decision-theoretic approach to multi-robot mapping and exploration. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.
7. K. Konolige. Large-scale map-making. In *Proc. AAAI*, 2004.
8. F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.
9. P. Modi, W. Shen, M. Tambe, and M. Yokoo. An asynchronous complete method for distributed constraint optimization. In *Proceedings of Autonomous Agents and Multi-Agent Systems, 2003.*, 2003.
10. B. Stewart, J. Ko, D. Fox, and K. Konolige. The revisiting problem in mobile robot map building: A hierarchical Bayesian approach. In *Proc. of the Conference on Uncertainty in Artificial Intelligence*, 2003.
11. A. Tews, M. Mataric, and G. Sukhatme. A scalable approach to human-robot interaction. In *Proc. of the IEEE International Conference on Robotics and Automation*, 2003.
12. Osher Yadgar, Sarit Kraus, and Charles Ortiz. Hierarchical organizations for realtime large-scale task and team environments. In *AAMAS*, 2002.

Large-Scale Robotic 3-D Mapping of Urban Structures

Michael Montemerlo and Sebastian Thrun

Stanford AI Lab
Stanford University
{mmde,thrun}@stanford.edu

Abstract. This article presents results for building accurate 3-D maps of urban environments with a mobile robot based on the Segway scooter. The goal of this project is to use robotic systems to rapidly acquire accurate 3-D maps which seamlessly integrate indoor and outdoor structures. Our approach uses an efficient implementation of the global scan alignment algorithm of Lu and Milios in order to integrate GPS, IMU, and laser data into globally consistent maps. The 3-D models acquired by the robot are analyzed for navigability using a multi-resolution evidence grid approach, and visualized using a meshing algorithm adapted from the computer graphics literature. Results are presented for a number of environments which combine indoor and outdoor terrain.

1 Introduction

In recent years, there has been a number of projects seeking to map physical environments. Classical work includes mapping from the air [12], the ground [7], indoors [6,11], outdoors [24], under water [28] and in the subterranean world [3]. The development of techniques for the acquisition of such maps has been driven by a number of desires. They include photo-realistic rendering [1,2], surveillance [27], measurement [3], and robot guidance [28]. Not surprisingly, the best work in this area has emerged from a number of different scientific fields, such as photogrammetry, computer vision, computer graphics [13,22], and robotics [25].

This paper describes a robotic system designed to acquire large 3-D maps. Urban terrain possesses a number of characteristic features: It combines large open places, such as plazas and roadways, with narrowly confined spaces, such as building interiors. GPS is often inaccurate in outdoor urban areas due to multi-path effects, and unavailable indoors. From a SLAM (simultaneous localization and mapping) perspective, maps of the size targeted by our research involve 10^6 or more features; gathered over 10^5 poses. Urban terrain is non-flat, hence the robot has to be localized in 6-D. The overall SLAM problem, thus is orders of magnitude more complex than prior work. While applying equally well in to 6-D data in theory, the vast majority of deployed SLAM algorithms have only been applied in planar environments with 3-D poses, to keep the data sets manageably small. Even those that perform 3-D mapping, often do so via 2-D SLAM [9], with the exception of [21] which offers no provision for closing cycles [8,4]. Further, past work has not provided effective means to incorporate occasional GPS measurements.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 141–150, 2006.

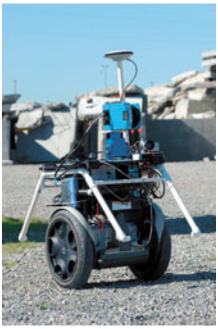


Fig. 1. The Segbot, a robot based on the Segway RMP platform and developed through the DARPA MARS program.

2 SLAM in Urban Environments

A key problem in building large-scale urban maps pertains to the ability to integrate information from multiple information sources, specifically GPS (global positioning system), IMU (the inertial measurement unit), odometry, and the LIDAR sensor(a laser-light detection and ranging sensor). This mapping problem is a version of the SLAM problem, short for simultaneous localization and mapping. The SLAM problem is characterized by a necessity to estimate the map of an environment while simultaneously localizing the sensor relative to the map. Outdoors, GPS provides absolute position labels; indoors, it presently does not.

Our approach builds on prior work on SLAM by Lu and Milios, who proposed Kalman filter-based approach that represents SLAM posteriors through collections of local constraints between nearby poses [15] (see also [8]). Also related is recent work in [4,20,26], who propose variants of the information filter for solving the SLAM problem. These algorithms are approximate, and they fail to integrate occasional global position data when available. However, both families of approaches are related in that they represent SLAM posteriors through local constraints—which is in stark contrast to the classical SLAM solution, the EKF [23], which maintains a full covariance between any two features.

Our approach represents the SLAM posterior as an undirected Markov network, where nodes correspond to poses, GPS measurements, and range measurements. The network possesses three types of pairwise node potentials: There are potentials between range measurements and the corresponding pose at which the measurement was required; there are potentials between subsequent poses, governed by the IMU measurements. And finally, there are absolute location potentials for poses at which GPS data was received. All of these potentials are nonlinear-quadratic; they are composed of a deterministic non-linear projective function (e.g., the robot motion model; the measurement model) with a quadratic penalty function that measures deviations from this non-linear projection. This representation generalizes past work on SLAM, most notably [15], in that the resulting sum of potentials can be thought of as a non-normalized log-likelihood function. However, representing them as potentials avoids numerical instabilities of the covariance representation in [15].

The map can be retreived by finding the most probable set of robot poses given the set of non-linear constraints. This is done by repeatedly linearizing the constraints

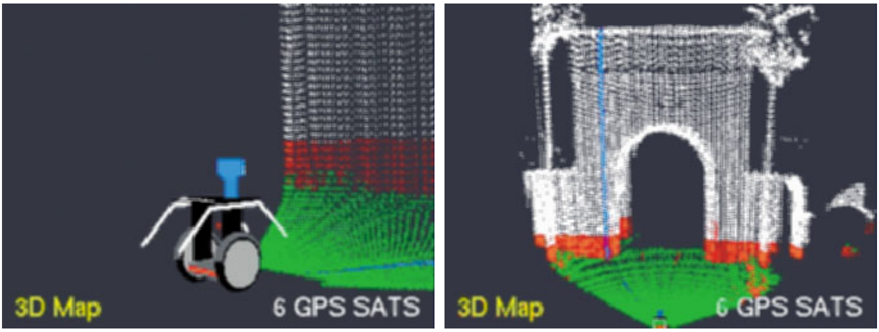


Fig. 2. Data acquisition through a two-directional scanning laser (the blue stripe indicates a vertical scan). The coloring indicates the result of terrain analysis: The ground surface is colored in green, obstacles are red, and structure above the robot's reach are shown in white.

and solving the resulting linear least squares problem. This least squares problem is extremely sparse because each pose is only linked to other nearby poses and measurements. As a result, the system can be solved efficiently using iterative algorithms such as conjugate gradient descent. The number of linearization steps required depends on the quality of the initial solution.

The advantage of this approach is threefold: It is free of numerical instabilities; it can represent extremely high-dimensional Gaussian SLAM posteriors; and the resulting optimization is efficient: Generating an actual map takes in the order of seconds on a low-end workstation. The process performs the mapping in 6-D [21]. The curvature of seemingly urban flat terrain tends to be sufficiently non-flat that SLAM approaches that assume the robot operates on a plane are simply inapplicable. The 6-D optimization also accommodates the fact that the Segway's pitch is continuously adjusted so as to not lose balance.

3 Navigation

A key component of our approach pertains to the "understanding" of the terrain and surrounding urban structures, to the extent necessary for safe robot navigation. Since our robot sometimes navigates autonomously, our approach also analyzes terrain for possible obstacles, using an algorithm that generalizes related work in [10,16].

Our basic approach analyzes scans for three type of obstacles: (1) terrain that is too steep or too rugged for safe traversal, such as curbs; (2) obstacles protruding into the workspace of the robot, such as overhangs, and (3) locations that lacks the necessary physical support for the robot, such as holes in the ground. The first two obstacle types are "positive obstacles," meaning that they can be detected by analyzing scan measurement points. The third type is commonly called a "negative obstacle," to indicate that such obstacles are only detectable by the absence of sensor measurements.

Our approach identifies positive obstacles by analyzing individual ground scan lines. Each scan line contains a sequence of measurement points. By calculating the derivative of these points in workspace coordinates, our robot can assess the

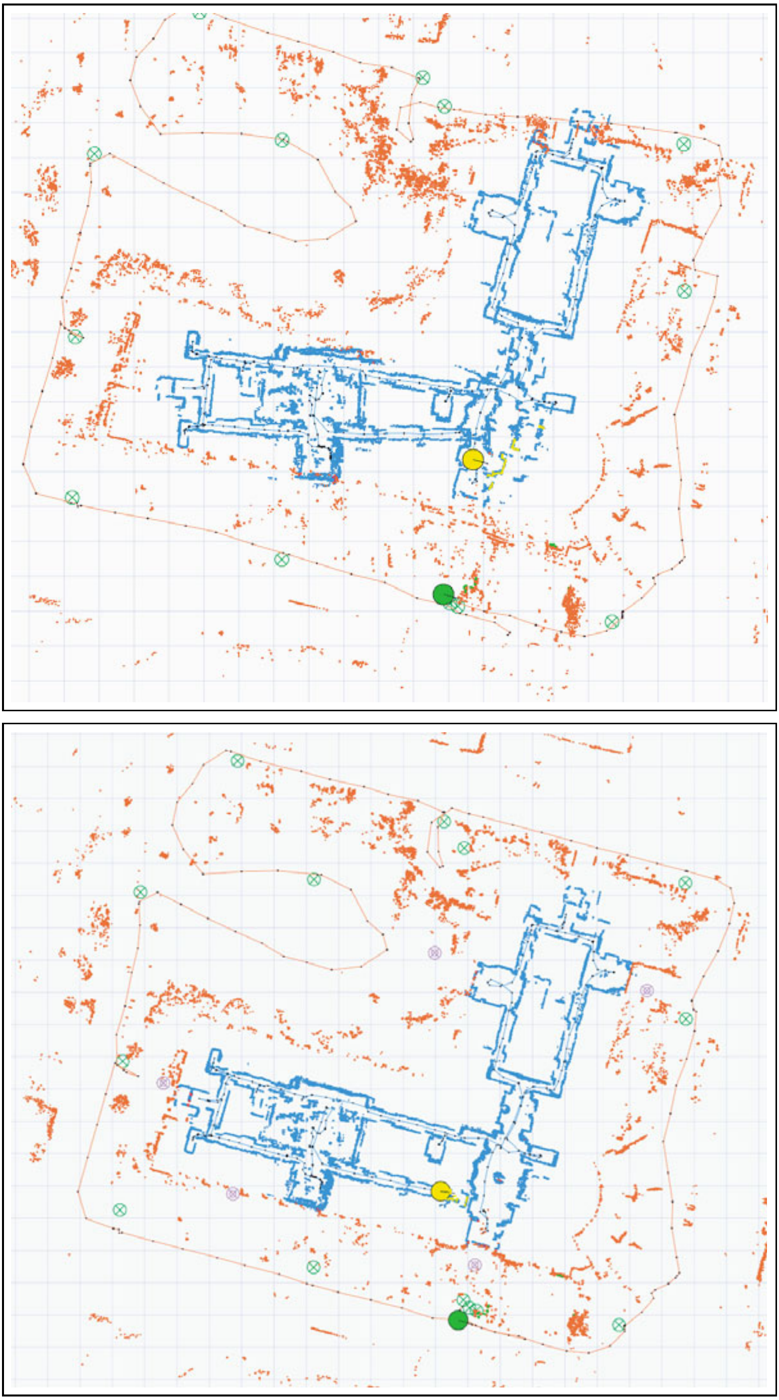


Fig. 3. Indoor mapping. Top: Cross section of map data without any processing. Bottom: Same model incorporating the outdoor GPS information. This experiment highlights the utility of global information in interior spaces.

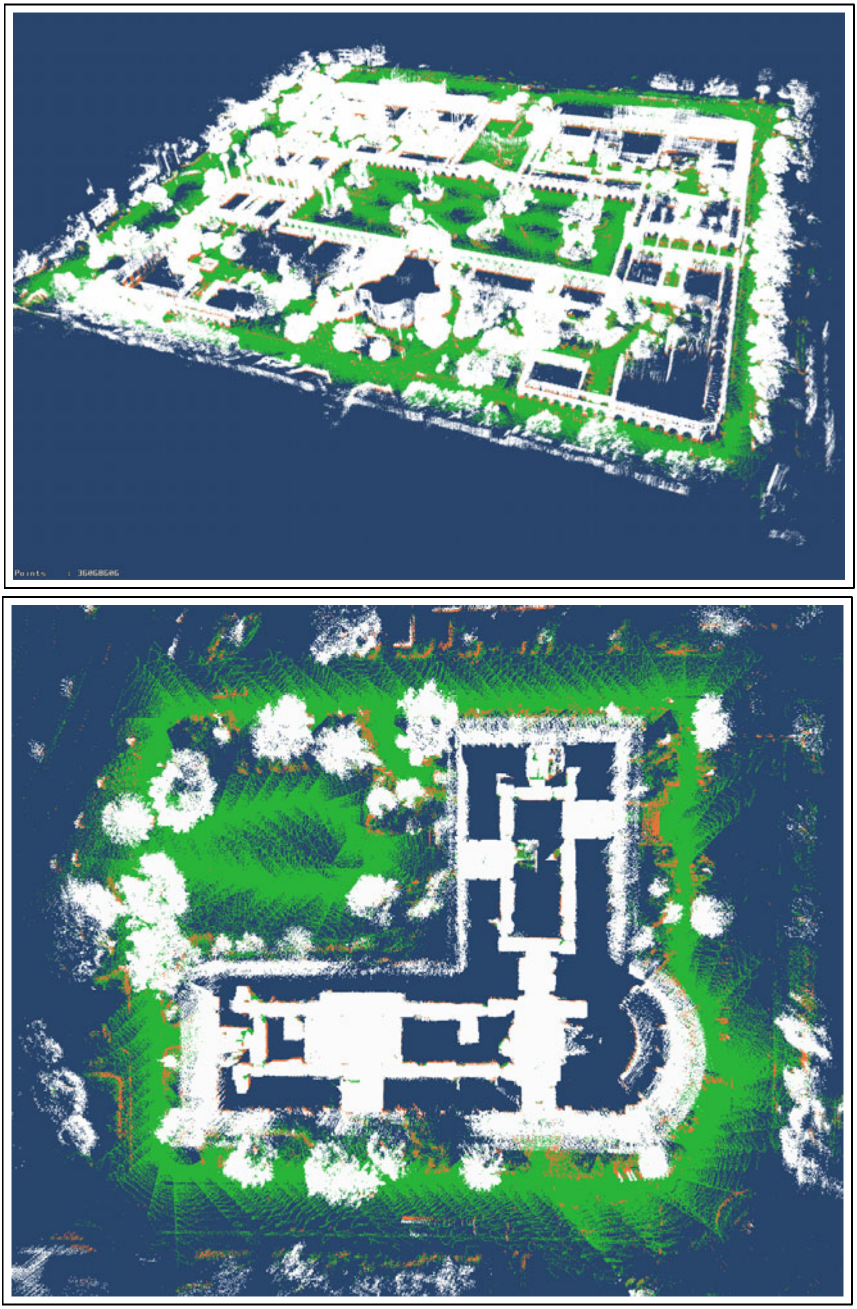


Fig. 4. Top: A map of Stanford University's main campus, whose diameter is approximately 600 meters. Bottom: Combined 3-D map of the interior of Gates Computer Science building and the surrounding terrain.

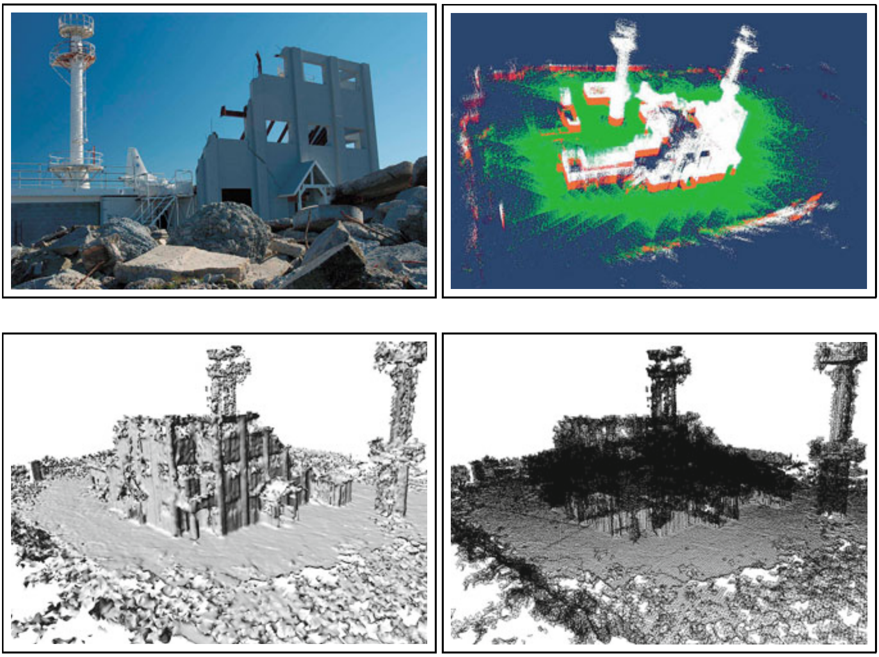


Fig. 5. Visualization of the NASA Ames ARC Disaster Assistance and Rescue Team training site in Moffett Field, CA. This site consist of a partially collapsed building with two large observation platforms. Top: Model. Bottom: Edges.

steepness of individual ground patches. In this way, it can avoid obstacles such as steep ramps and upwards staircases. Objects protruding into the robot's workspace are identified by searching up to a limited height for measurement points relative to the ground area. By limiting the search height, our vehicle can navigate indoor environments and underneath overhangs.

Negative obstacles are detected by lack of supporting ground plane. However, this analysis is somewhat non-trivial, due to the sparseness of measurement points at longer ranges. To perform this analysis, most existing techniques partition the workspace into a grid, similar to the well-known occupancy grid map algorithm [19]. For each grid cell, sensor measurements are integrated using Bayes rule to gradually increase coverage, while at the same time diminishing the effect of sensor noise.

Real-world terrain sensors have limited measurement resolution. For example, our laser range finders can only measure ranges with $0.5°$ accuracy; similar limitations exist for stereo camera systems and sonar sensors. Limited resolution causes two problems with standard evidence grid algorithms: First, the limited resolution may make it impossible to detect small obstacles at a distance. Obstacles like curbs or low-hanging wires are typically only detectable at close ranges. Second, limited sensor resolution makes it difficult to systematically find navigable terrain at a distance. As a result, a motion planner is either forced to make optimistic assumptions about the nature of terrain at a distance (with the obvious expense of having to

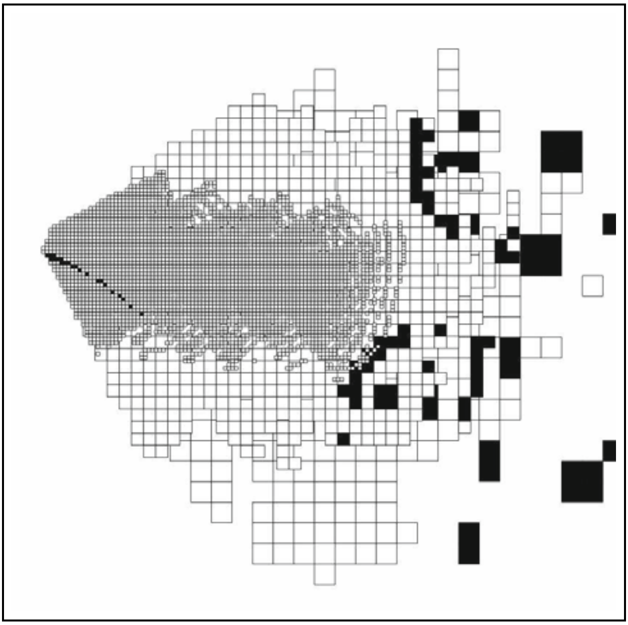


Fig. 6. Multi-resolution pyramid model of the robot's surroundings. The majority of holes in the map are filled in. The terrain map close to the robot is very high resolution, while the area far from the robot is very coarse.

replan when negative obstacles are encountered), or must remain confined to nearby regions that have been completely imaged.

Our approach relies on a set of N evidence grids of differing resolutions, each sensitive to different intervals of sensor readings. Grids sensitive to longer ranges are coarser than grids closer to the robot. The advantage of using such a multi-resolution grid is two-fold. First, the coverage in areas further away is increased, without compromising the spatial distribution of the overall map. This leads to improved paths of the robot during autonomous motion. Second, and possibly more importantly, evidence of non-traversibility acquired at short range cannot be overridden by evidence of traversibility acquired at longer range. This common failure mode of evidence grids is the result of using gradients for traversibility analysis: such gradients are necessarily less accurate at a distance, where the density of measurements is reduced. As a result, small obstacles such as curbs are usually not detected at far range. A standard occupancy grid technique would consequently fail to update a cell as non-traversable while the obstacle is still far away. Results in [18] suggest that the time at which small obstacles are detectable at short range may be too short for overriding the evidence acquired at longer ranges; as a result, the robot may run into such obstacles.

Our multi-resolution approach overcomes this by maintaining range-specific maps. Small obstacles such as curbs do not show up in the coarse, long-range map, but they do show up in the fine-grained short-range map. When combining maps for assessing the navigability of terrain, preference is given to shorter range maps; however, all maps participate in motion planning.

4 Visualization

The result of the mapping process is a point cloud, in which each point is a noisy measurement of a boundary between free and occupied space. To visualize this data as a solid model, we employ a well-known technique from computer graphics called Space Carving [5]. This algorithm builds up a signed distance function that represents the distance from any point in space to the surface. This function is sampled over a uniform 3-D grid. Each range reading is used to update a small section of the grid, similarly to evidence grid mapping algorithms in robotics. A mesh can be constructed by extracting the zero-surface of the signed distance function using any of a variety of isosurface algorithms, such as Marching Cubes [14].

5 Results

We conducted a number of experiments, all with the vehicle shown in Figure . In particular, we have mapped a number of urban sites, including NASA's DART Search and Rescue Facility, and a large fraction of Stanford's main campus; snapshots of these experiments will be discussed below.

Our experiments either involve the collection of a single large dataset, or a number of smaller datasets. The latter has become necessary since for the environments of the size studied here, the robot possesses insufficient battery capacity to collect all data within a single run. In most experiments, the robot is controlled manually. This is necessary because the urban environments are usually populated with moving objects, such as cars, which would otherwise run danger of colliding with our robot. We have, on several occasions, used our navigation package Carmen [17] to drive the robot autonomously, validating the terrain analysis techniques discussed above.

Our research has led to a number of results. First and foremost, a primary finding is that with our representation, maps with more than 10^5 variables can be computed quickly, even under multiple loop-closure constraints. We find that scan matching can be achieved in real-time, as the robot moves, using a portable laptop computer. This is a long-known result for horizontally mounted laser range finders, but it is reassuring that the same applies to the more difficult scan matching problem involving a vertically panning scanner. More importantly, the relaxation of the pose potentials takes in the order of 30 seconds even for the largest data set used in our research, of an area 600m by 800m in size, and with a dozen cycles. This suggests the appropriateness of our representation as an algorithms for large-scale urban mapping.

The second and possibly more important result pertains to the utility of GPS data for indoor mapping. In general, GPS signals can not be received indoors. However, when a robot moves from the outdoors into an indoor map, robot poses inside the map become indirectly linked to the GPS measurements. The resulting indoor maps are significantly more accurate than indoor maps without access to global information.

This also has the added benefit of generating accurately geo-referenced models of the interiors of buildings. Further below, we will discuss an experimental snapshot of combined indoor-outdoor mapping.

Finally, when in autonomous mode, we find that our terrain analysis techniques provide effective navigation at speeds of up to one meter per second. The vehicle navigates collision-free and successfully avoids negative obstacles, but sometimes fails to detect fast moving obstacles fast enough to avoid a collision. The latter is because of the panning motion of the sensor, which requires approximately 2 seconds for a full sweep. We also find that curbs are reliably identified through our multi-resolution approach, where the fixed-resolution evidence grid approach fails.

Experimental snapshots can be found in Figures 3 through 5. Figures 4 and 5 show some of the maps acquired by our system. All maps are substantially larger than previously software could handle; all are constructed with some GPS information. The map shown on the left in Figure 4 corresponds to Stanford's main campus; the one on the right is an indoor-outdoor map of the building that houses the computer science department.

The key result of improved indoor maps through combining indoor and outdoor mapping is illustrated in Figure 3. Here we show a 2-D slice of the 3-D map using SLAM under two different conditions: In the map on the left, the indoor map is constructed independently of the outdoor map, whereas the right map is constructed jointly. As explained, the joint construction lets GPS information affect the building interior through the sequence of potentials liking the outdoor to the indoor. As this figure suggests, the joint indoor-outdoor map is significantly more accurate; in fact, the building possesses a right angle at its center, which is well approximated.

Figure 6 shows a snapshot of our multi-resolution grid map for finding negative obstacles. The resolution depends on the distance to the robot. This specific snapshot shows several curbs, some larger ones far away, and one near the robot that a flat approach would have failed to identify.

Acknowledgement
The research has been sponsored by DARPA's MARS Program (contracts N66001-01-C-6018 and NBCH1020014), which is gratefully acknowledged.

References

1. P.K. Allen and I. Stamos. Integration of range and image sensing for photorealistic 3D modeling. ICRA-2000.
2. R. Bajcsy, G. Kamberova, and L. Nocera. 3D reconstruction of environments for virtual reconstruction. WACV-2000.
3. C. Baker, A. Morris, D. Ferguson, S. Thayer, C. Whittaker, Z. Omohundro, C. Reverte, W. Whittaker, D. Hähnel, and S. Thrun. A campaign in autonomous mine mapping. ICRA-2004.
4. M. Bosse, P. Newman, M. Soika, W. Feiten, J. Leonard, and S. Teller. An atlas framework for scalable mapping. ICRA-2003.

5. B. Curless and M. Levoy. A volumetric method for building complex models from range images. SIGGRAPH-1996.
6. S. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin. Sensor based creation of indoor virtual environment models. VSMM-1997.
7. A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3), 1987.
8. J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. CIRA-2000.
9. D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun. Towards lazy data association in SLAM. ISRR-2003.
10. K. Hashimoto and S. Yuta. Autonomous detection of untraversability of the path on rough terrain for the remote controlled mobile robots. FSR-2003.
11. L. Iocchi, K. Konolige, and M. Bajracharya. Visually realistic mapping of a planar environment with stereo. ISER-2000.
12. G. Konecny. *Geoinformation: Remote Sensing, Photogrammetry and Geographical Information Systems*. Taylor & Francis, 2002.
13. M. Levoy. The digital michelangelo project. 3DIM-1999.
14. W. Lorensen and H. Cline. A high resolution 3d surface construction algorithm. SIGGRAPH-1987.
15. F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4, 1997.
16. L. Matthies, T. Litwin, K. Owens, A. Rankin, K. Murphy, D. Coorobs, J. Gilsinn, T. Hong, S. Legowik, M. Nashman, and B. Yoshimi. Performance evaluation of UGV obstacle detection with ccd/flir stereo vision and ladar. JCSTIS-1998.
17. M. Montemerlo, N. Roy, and S. Thrun. Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (CARMEN) toolkit. IROS-2003.
18. M. Montemerlo and S. Thrun. A multi-resolution pyramid for outdoor robot terrain perception. AAAI-2004.
19. H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2), 1988.
20. P. Newman. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem.* PhD thesis, Sydney, 2000.
21. A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with application in autonomous mine mapping. ICRA-2004.
22. S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. 3DIM-2001.
23. R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4), 1986.
24. S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master. Calibrated, registered images of an extended urban area. CVPR-2001.
25. S. Thrun. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
26. S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 2004.
27. C.-C. Wang, C. Thorpe, and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. ICRA-2003.
28. S. Williams, G. Dissanayake, and H.F. Durrant-Whyte. Towards terrain-aided navigation for underwater robotics. *Advanced Robotics*, 15(5), 2001.

V. Medical Robotics

As robot applications in human environments increase, we see many interesting application in medicine. These include robotic technologies used in robot-doctor interfaces for minimally invasive surgery and novel robotic devices that can navigate inside human bodies. This chapter presents 3 interesting articles representing the various facets of medical robotics. Papers in other chapters, however, also describe medical applications of robotics.

The first article by Zemeti and co-workers presents the design and analysis of a Minimally Invasive Surgery robot. The trocar is designed with force measurement capability, where the force sensor is placed outside the patient – "to reduce cost and sterilizibility requirements". This paper presents the results of the feasibility experiments.

In the next article, Dario and colleagues take medical robotics a step further into the future by reporting on the concept and the preliminary modeling of legged micro robot locomoting in a tubular, slippery and compliant environment. The intended application is for the microrobots to navigate inside the gastrointestinal tract for diagnosis and therapy. The microcapsules are designed to be ingestible and then to make its way to the gastrointestinal tract. The new contribution being studied here is the micro robot's capability of effective locomotion while inside the human body.

The third article by Casals, *et al*. presents a multimodal approach to human-machine interface, applied to medical robotics. This carries a similar idea to the first article in Chapter XIII by Iba *et al*, under the heading of Haptics and Augmented Reality. The idea is to provide a range of media of communication between the surgeon and the robots, thus creating a quasi-hands-free control of the equipments. This would allow the surgeon to better control the equipments in the operating room without losing too much focus on the task at hand. The mode of interaction being studied is gesture recognition, with other modes available such as: tactile, speech, pedals, etc.

A Force Controlled Laparoscopic Surgical Robot without Distal Force Sensing

Nabil Zemiti, Tobias Ortmaier, Marie-Aude Vitrani, and Guillaume Morel

Laboratoire de Robotique de Paris (LRP), BP 61,
Route du Panorama, 92 265 Fontenay aux Roses Cedex, Paris, France

Abstract. Minimally invasive surgery (MIS) challenges the surgeon's skills due to his separation from the operation area which can be reached with long instruments only. Therefore, the surgeon loses access to the manipulation forces inside the patient. This reduces his dexterity when performing the operation. A new compact and lightweight robot for MIS is presented which allows for the measurement of manipulation forces. The main advantage of this concept is that no miniaturized force sensor has to be integrated into surgical instruments and inserted into the patient. Rather, a standard sensor is attached to a modified trocar outside the patient, which allows for the measurement of manipulation forces. This approach reduces costs and sterilizability demands. Results of first force control experiments are presented to show the feasibility of the concepts.

1 Introduction

The use of robots for surgical interventions is a an approach that is now proven to increase the quality of operations and to establish new types of surgical procedures (see [1] for an up-to-date overview of this research field). Especially, minimally invasive surgery (MIS) in which long instruments are used to gain access to the area of interest seems to be a promising field for robotic surgery. Here, robots help the surgeon to regain virtually direct access to the operation field he is seperated from: actuated instruments provide him with full dexterity inside the patient as in open surgery. In order to enhance the overall system performance, force control capabilities are desirable [2,3]. These forces can be used in two ways: it becomes possible to run the robot in a force controlled mode, helping to prevent unintentional damage of tissue or to compensate for organ motion in case of contact between instrument and organ [4]. Furthermore, manipulation forces can be displayed back to the surgeon (with appropriate kinesthetic input devices), providing him with direct sensation of the remote forces applied. Unfortunately, the therefore necessary measurement of manipulation forces in minimally invasive robotic surgery (MIRS) is rarely realized as appropriate small and sterilizable force sensors which can be inserted into the patient are still missing [5]. In this paper the new robot MC^2E (french abbreviation for compact manipulator for endoscopic surgery) is presented, offering a new possibility of force measurement in MIRS. The robot is shown in Fig. 1 during in vivo experiments realized at the Surgical School of Paris.

The reminder of this article is organized as follows: In Sect. 2 the robot's kinematics is presented. Section 3 describes the novel principle of force measurement

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 153–163, 2006.

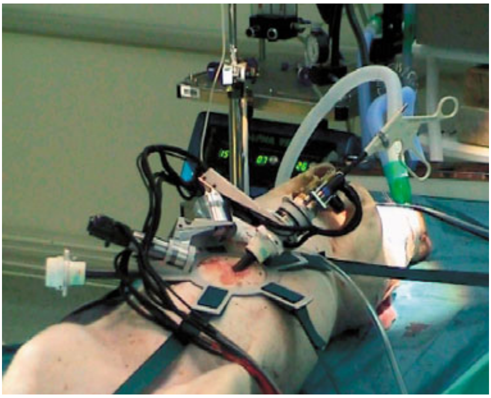
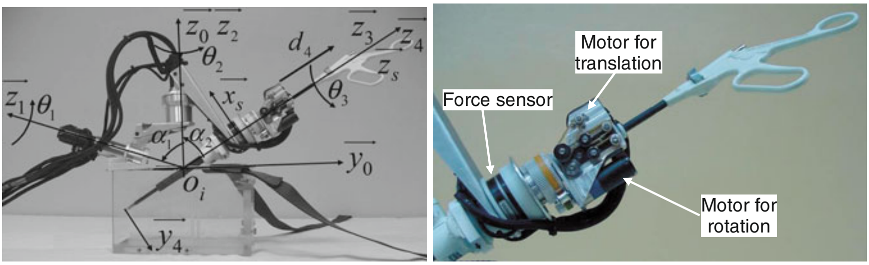


Fig. 1. MC^2E during in vivo experiments on a pig.

for MIRS in detail. Force control and experimental results are given in Sect. 4. A discussion of the results and further directions for research are given in Sect. 5.

2 Robot and Kinematics

A robot used in the operating room (OR) has to be lightweight and compact, as only a small amount of space for additional equippment is available. Furthermore, a lightweight robot can be easily mounted and removed by one nurse which helps to reduce preoperative setup time and is also a safety feature in emergency situations.



(a) Assembled robot (b) Upper part of the MC^2E

Fig. 2. MC^2E robot together with coordinate frames and Denavit-Hartenberg parameters.

Unlike most other MIS robots MC^2E moves not only the instrument but also the trocar in which the instrument is inserted. As shown in Fig. 2 the robot is comprised of two parts: the lower part moves the trocar and is a compact spherical 2 DoFs mechanism (Θ_1 and Θ_2) providing an invariant center at the fulcrum point. The base of this lower subsystem is easily installed on the patient’s body and clipped to the trocar. The upper part is mounted on the trocar and provides 2 DoFs: rotation about the instrument axis (Θ_3) and translation along the instrument axis (d_4). The upper part is depicted in detail in the right side of Fig. 2. The design of the robot is rather compact. Furthermore, it allows for the use of standard disposable instruments, and enables co-manipulation by the surgeon and the robot. Note that similar compact designs are proposed in [6,7].

Two different types of motors from *Faulhaber* were chosen: Two powerful motors (ref. 2342S024CR, 12 Watt) for the spherical part of the robot and two smaller motors (ref. 1724T003SR, 4 Watt) for the upper part of the robot. The encoder resolution for all DoFs is 512 per revolution of the motor, providing in combination with the gear ratios sufficient resolution at the link side for high accuracy motion. The robot is equipped with a *Nano43* 6 axis force/torque sensor from *ATI Industrial Automation*. The particular mounting is described in detail in Sect. 3. A sample rate of 670 Hz is used to realize the force control law presented in Sect. 4.

Table 1. Denavit-Hartenberg parameters.

	α_i	a_i	Θ_i	d_i		α_i	a_i	Θ_i	d_i
${}^0\boldsymbol{T}_1$	70°	0	Θ_1	0	${}^2\boldsymbol{T}_3$	−60°	0	Θ_3	0
${}^1\boldsymbol{T}_2$	−70°	0	Θ_2	0	${}^3\boldsymbol{T}_4$	0	0	0	-d_4

The modified Denavit-Hartenberg (DH) parameters as shown in Fig. 2 (together with the relevant frames) are summarized in Table 1. The overall transformation from the instrument tip frame F_4 to the robot base frame F_0 is:

$$
{}^0\boldsymbol{T}_4 = {}^0\boldsymbol{T}_1 {}^1\boldsymbol{T}_2 {}^2\boldsymbol{T}_3 {}^3\boldsymbol{T}_4 = \begin{bmatrix} {}^0\boldsymbol{R}_4^{3\times 3} & \boldsymbol{v} \\ \boldsymbol{0}^T & 1 \end{bmatrix} . \tag{1}
$$

Noticing that the angle Θ_3 does not affect the instument tip position $\boldsymbol{v}$, one selects the following vector in order to parameterize the 4 instrument’s DoFs:

$$
\boldsymbol{x} = [\boldsymbol{v}^T(\Theta_1, \Theta_2, d_4), \;\; \Theta_3]^T . \tag{2}
$$

In the rest of the paper, the angle Θ_3 is considered to be fixed, and only the position $\boldsymbol{v}$ of the instrument tip is controlled. The Jacobian $\boldsymbol{J}^{3\times 3}$ for the translational DoFs can thus be written as:

$$
\boldsymbol{J} = \frac{\partial \boldsymbol{v}}{\partial \left[\Theta_1, \; \Theta_2, \; d_4\right]^T} . \tag{3}
$$

The singularities of the translational workspace are given by the solutions of:

$$
\det(\boldsymbol{J}) = 0 = -\mathrm{d}_4^2 \sin(\alpha_1) \sin(\alpha_2) \sin(\Theta_2) . \tag{4}
$$

Thus, singularities are encountered when:

$$
d_4 = 0\,, \quad \text{or} \quad \Theta_2 = \pm \mathrm{k}\pi \quad \text{with} \quad \mathrm{k} = 0, 1, 2, ... \tag{5}
$$

The first singularity, which is of second order, is reached when the instrument tip is at the fulcrum point. It has no consequence during experiments. Indeed, it is crossed only when inserting or removing the instrument, which can be done under joint position control. For the second singularity, $\Theta_2 = \pi$ cannot be reached due to joint limits. Thus, the only physically feasible singularity is $\Theta_2 = 0$, which separates the workspace into two parts. During in vivo experiments, one chooses initially for $\Theta_2 > 0$ or $\Theta_2 < 0$ and the singularity is never crossed.

3 Force Measurement

In manual MIS manipulation forces cannot be sensed by the surgeon anymore, due to the friction in the trocar. It is expected that force measurement and force feedback in MIRS increase the immersion of the surgeon into the remote side. Furthermore, measurement of forces is a prerequisite for force control. This, again, helps to avoid damage of tissue and suturing material and might also lead to new operation techniques as manipulation with predefined forces become possible [4,2,3].

Force measurement can be realized by placing miniaturized force/torque sensors near the instrument tip inside the patient [5]. Here, questions of sterilizability and electromagnetic compatibility still need to be answered. Alternatively, if the force sensor was integrated in the instrument shaft and placed outside the patient, disruption of the force measurement would occur due to friction in the trocar and torques necessary to rotate the trocar around the fulcrum point. This would be especially the case in heart surgery where the trocar is placed in the narrow space between the ribs.

3.1 Measurement Principle

The solution proposed here is a new trocar in which the sensor is integrated, but placed outside the patient, avoiding the before mentioned problems. This is possible, as the trocar is moved (and so is the force sensor) to realize motion inside the patient (see Sect. 2). The trocar is depicted in Fig. 3: the instrument is placed inside a passive guidance, which increases the rigidity of the system. The passive guidance is attached to the upper part of the force/torque sensor. The lower part of the sensor is placed on a conventional trocar. This set up allows for the measurement of the interaction forces between instrument tip and tissue, without having to cope with friction inside the trocar. The dynamic equation of the instrument (1) writes:

$$0 = \boldsymbol{w}_{2\to 1} + \boldsymbol{w}_{3\to 1} + \boldsymbol{w}_{7\to 1} + \boldsymbol{w}_{g\to 1} - \boldsymbol{w}_{d_1} , \tag{6}$$

where $\boldsymbol{w}_{i\to j}$ denotes the wrench applied by part i to part j, $\boldsymbol{w}_{g\to i}$ is the wrench applied to part i due to gravity, and $\boldsymbol{w}_{d_i}$ is the dynamic wrench of part i, accounting for the inertial effects due to acceleration. Similarily, the dynamic equations of parts (2) and (3) are:

$$0 = \boldsymbol{w}_{1\to 2} + \boldsymbol{w}_{3\to 2} + \boldsymbol{w}_{g\to 2} - \boldsymbol{w}_{d_2} , \tag{7}$$

$$0 = \boldsymbol{w}_{2\to 3} + \boldsymbol{w}_{1\to 3} + \boldsymbol{w}_{4\to 3} + \boldsymbol{w}_{g\to 3} - \boldsymbol{w}_{d_3} . \tag{8}$$

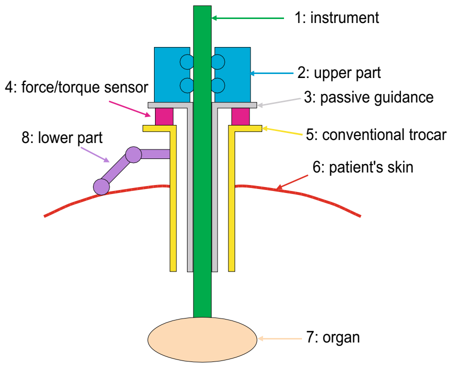


Fig. 3. Modified trocar with integrated force.

The wrench $\boldsymbol{w}_{1\to7} = -\boldsymbol{w}_{7\to1}$ is the interaction of interest (between instrument and organ) whereas $\boldsymbol{w}_{4\to3}$ is the wrench measured by the sensor. Summing Eq. 6, Eq. 7, and Eq. 8 yields:

$$\boldsymbol{w}_{1\to7} = \boldsymbol{w}_{4\to3} + \boldsymbol{w}_g - \boldsymbol{w}_d \ , \tag{9}$$

where $\boldsymbol{w}_g$ and $\boldsymbol{w}_d$ are the total gravity wrench and dynamic wrench, respectively:

$$\boldsymbol{w}_g = \boldsymbol{w}_{g\to1} + \boldsymbol{w}_{g\to2} + \boldsymbol{w}_{g\to3} \quad \text{and} \quad \boldsymbol{w}_{\mathrm{d}} = \boldsymbol{w}_{\mathrm{d}_1} + \boldsymbol{w}_{\mathrm{d}_2} + \boldsymbol{w}_{\mathrm{d}_3} \ . \tag{10}$$

Remarkably, neither the friction between the instrument and the passive guidance, $\boldsymbol{w}_{1\to3}$, nor the wrench between the trocar and the patient's skin, $\boldsymbol{w}_{5\to6}$, influence the measurement. Therefore, there is no need for any model of these disturbances. Rather, in order to calculate the interaction wrench $\boldsymbol{w}_{1\to7}$ one has to know the gravitation wrench $\boldsymbol{w}_g$ and the dynamic wrench $\boldsymbol{w}_d$. Usually, $\boldsymbol{w}_d \approx 0$ holds, as velocities and accelerations in MIS are rather small. Therefore, in practice, estimating the distal interaction $\boldsymbol{w}_{1\to7}$ from the measured wrench $\boldsymbol{w}_{4\to3}$ is reduced to a gravity compensation algorithm, which is detailed in the next section.

3.2 Gravity Compensation

The influence of gravity is calculated using a model of the robot whith several unknown parameters, which need to be identified. These parameters can be divided into two groups: fixed parameters which do not change between experiments and variable parameters which vary between experiments. The fixed parameters are calculated offline, only once, whereas the others need to be calculated online at the beginning of each experiment.

The fixed parameters are the angle ψ describing the rotation about the z_4-axis between the sensor frame $\mathcal{F}_s$ and frame $\mathcal{F}_3$, as well as the center of gravity ${}^sG$ (expressed in $\mathcal{F}_s$) of the parts 1, 2, and 3 of the robot, as depicted in Fig. 2 and Fig. 3.

One of the variable parameters is the unknown weight ${}^0\boldsymbol{p} = m\,{}^0\boldsymbol{g}$ expressed in the robot base frame $\mathcal{F}_0$. This parameter is variable since the robot is mounted on the patient in an arbitrary orientation. Furthermore, the sensor measurement offsets (${}^{s}\boldsymbol{f}_{o}$ for forces and ${}^{s}\boldsymbol{t}_{o}$ for torques), and the mass m of the parts 1, 2, and 3 of the robot are not known. Note, that the mass m does not change during the experiments, but m can be easily estimated online, too:

$$m = \frac{\|{}^{0}\boldsymbol{p}\|_{2}}{9.81\ \mathrm{m/s^{2}}}\ . \tag{11}$$

Therefore, m can be used to verify the online calibration results. With these parameters the following model for forces due to gravity ${}^{s}\boldsymbol{f}_{m}$ can be defined (in sensor frame $\mathcal{F}_{s}$):

$${}^{s}\boldsymbol{f}_{m} = {}^{s}\boldsymbol{R}_{3}{}^{3}\boldsymbol{R}_{0}{}^{0}\boldsymbol{p} + {}^{s}\boldsymbol{f}_{o}\ . \tag{12}$$

For the torques the following equation holds (again in sensor frame $\mathcal{F}_{s}$):

$${}^{s}\boldsymbol{t}_{m} = \left[{}^{s}\boldsymbol{R}_{3}{}^{3}\boldsymbol{R}_{0}{}^{0}\boldsymbol{p}\right] \times {}^{s}\boldsymbol{d}_{GS} + {}^{s}\boldsymbol{t}_{o} = -[{}^{s}\boldsymbol{d}_{GS}]_{\times}\left[{}^{s}\boldsymbol{R}_{3}{}^{3}\boldsymbol{R}_{0}{}^{0}\boldsymbol{p}\right] + {}^{s}\boldsymbol{t}_{o}\ , \tag{13}$$

where ${}^{s}\boldsymbol{d}_{GS}$ is the vector from G to the center S of $\mathcal{F}_{s}$, expressed in $\mathcal{F}_{s}$, and $[\boldsymbol{a}]_{\times}$ denotes the skew symmetric matrix associated with a vector $\boldsymbol{a}$ such that, for any vector $\boldsymbol{b}$, $[\boldsymbol{a}]_{\times}\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$. The vector ${}^{0}\boldsymbol{p}/\|{}^{0}\boldsymbol{p}\|_{2}$ represents the unknown orientation of the gravity vector in the robot base frame $\mathcal{F}_{0}$. For matrix ${}^{3}\boldsymbol{R}_{0}$ the following relation holds:

$${}^{3}\boldsymbol{R}_{0}(\Theta_{1}, \Theta_{2}) = ({}^{0}\boldsymbol{R}_{3})^{-1} = ({}^{0}\boldsymbol{R}_{1}{}^{1}\boldsymbol{R}_{2}{}^{2}\boldsymbol{R}_{3})^{-1}\ , \tag{14}$$

that can be computed from the geometric model of the robot. The matrix ${}^{s}\boldsymbol{R}_{3}(\psi) = \mathrm{rot}(\psi, \boldsymbol{z}_{3})$ describes the rotation ψ about the z-axis between the sensor frame $\mathcal{F}_{s}$ and frame $\mathcal{F}_{3}$.

For the offline calibration $N = 100$ points covering the workspace were chosen and the joint values

$$\Theta_{1,i} \quad \text{and} \quad \Theta_{2,i} \quad \text{with} \quad i = 1, .., N\ . \tag{15}$$

are recorded. The sensor force values ${}^{s}\boldsymbol{f}_{s,i}$ and torque values ${}^{s}\boldsymbol{t}_{s,i}$ are simultaneously recorded. During this motion the instrument does not touch the environment, and the wrench is measured when the robot stays still ($i.e.$ $\boldsymbol{w}_{d} = 0$), so that the transmission model (Eq. 9) reduces to:

$$0 = \boldsymbol{w}_{4\to 3} + \boldsymbol{w}_{g}\ , \tag{16}$$

which means that the measure matches the gravity wrench.

For each position, the force ${}^{s}\boldsymbol{f}_{m,i}(\Theta_{1,i}, \Theta_{2,i}, {}^{0}\boldsymbol{p}, {}^{s}\boldsymbol{f}_{o}, \psi)$ is calculated according to Eq. 12. The optimal values for the unknown parameters are assumed to be at the minimum of the error function

$${}^{s}e_{f} = \frac{1}{N}\sum_{i=1}^{N}({}^{s}\boldsymbol{f}_{m,i} - {}^{s}\boldsymbol{f}_{s,i})^{T}({}^{s}\boldsymbol{f}_{m,i} - {}^{s}\boldsymbol{f}_{s,i})\ . \tag{17}$$

Table 2. Optimization results for the angle ψ, the components of the gravitation forces ${}^0\boldsymbol{g}$ forces, and the force offset ${}^s\boldsymbol{f}_o$ of the sensor.

$\psi[rad]$	${}^0g_x$ [N/kg]	${}^0g_y$ [N/kg]	${}^0g_z$ [N/kg]	m [kg]	${}^sf_{o,x}$[N]	${}^sf_{o,y}$[N]	${}^sf_{o,z}$ [N]
$\approx 3\pi/2$	0.0	-0.66	-9.79	0.289	0.64	2.8	1.28

To calculate the unknown parameters a gradient based approach is used, as ${}^se_f$ is a nonlinear function of ψ. The optimization results are summarized in Table 2. In this experiment the robot was placed on an approximately horizontal table, thus the estimated gravity field vector ${}^0\boldsymbol{g} = {}^0\boldsymbol{p}/m$ is expected to be close to ${}^0\boldsymbol{g}_{\text{th}} = [0, 0, -9.81\text{N/kg}]^T$. The angle β between the calculated ${}^0\boldsymbol{g}$ and ${}^0\boldsymbol{g}_{\text{th}}$ is $\beta = 3.87$ deg.

These results are now used to calculate the unknown offset ${}^s\boldsymbol{t}_o$ and the unknown center of gravity ${}^sG$. The following error function is used:

$$
{}^se_t = \frac{1}{N}\sum_{i=1}^{N}({}^s\boldsymbol{t}_{m,i} - {}^s\boldsymbol{t}_{s,i})^T({}^s\boldsymbol{t}_{m,i} - {}^s\boldsymbol{t}_{s,i}) \, . \tag{18}
$$

The results are summarized in Table 3 (note, the point ${}^sG$ is expressed in frame F_s).

Table 3. Optimization results for the center of gravity ${}^s\boldsymbol{G}$ and the torque offset ${}^s\boldsymbol{t}_o$ of the sensor.

${}^sG_x$ [mm]	${}^sG_y$ [mm]	${}^sG_z$ [mm]	${}^st_{o,x}$ [Nmm]	${}^st_{o,y}$ [Nmm]	${}^st_{o,z}$ [Nmm]
1.44	-4.42	42.2	-111	21.1	4.1

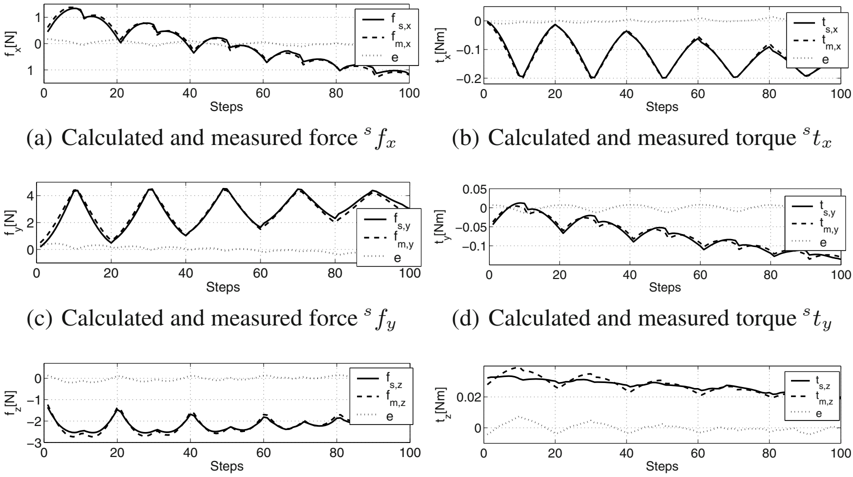
The high accuracy of this approach can be seen in Fig. 4 where the measured data and the model based (calculated) values are given.

4 Force Control

This section describes the chosen force control structure in detail and gives first experimental results.

At the lowest level of the controller, a joint position loop encapsulates the joint velocity loop realized in hardware by the power amplifiers. This joint position loop uses a simple proportional gain, avoiding the use of an integral term. As a consequence, the joint position servo loop is not extremely precise, but keeps enough bandwidth available for the outer force feedback loop.

The force feedback loop computes a velocity, fed to the position controller through an integrator. Although, a desired force ${}^4\boldsymbol{f}_d = [{}^4f_{d,x}, {}^4f_{d,y}, {}^4f_{d,z}]^T$ is to be exerted by the instrument tip on the organ, the vector ${}^4\boldsymbol{h} = [{}^4t_y, {}^4t_x, {}^4f_z]^T$ was chosen in order to be servoed. The vector ${}^4\boldsymbol{h}$ is constituted by the two components of the torque at the fulcrum point O that belong the plane $(O, \boldsymbol{x}_4, \boldsymbol{y}_4)$ and the force component along the instrument penetration axis $(O, \boldsymbol{z}_4)$.



(a) Calculated and measured force ${}^s f_x$ (b) Calculated and measured torque ${}^s t_x$

(c) Calculated and measured force ${}^s f_y$ (d) Calculated and measured torque ${}^s t_y$

(e) Calculated and measured force ${}^s f_z$ (f) Calculated and measured torque ${}^s t_z$

Fig. 4. Calculated and measured forces and torques in F_s during the offline calibration procedure. Furthermore, the error e is given.

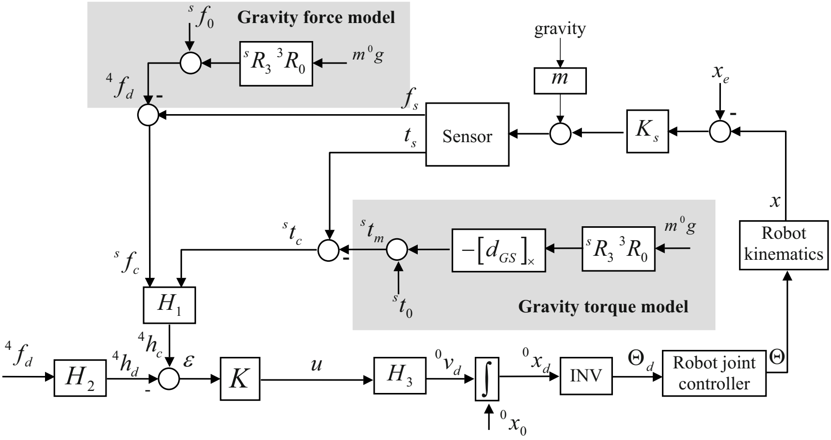


Fig. 5. Control structure used to servo the contact forces.

This vector was chosen in order to provide a stable behavior when operating in a comanipulation mode: in this case, the system runs a transparency mode, *i.e.* a force controlled mode with a zero desired value. Therefore, when a force is applied to the instrument, the robot must produce a motion that attempts to cancel the force. Moreover, the system should be transparent for forces exterted either by an organ inside a patient, or by the surgeon outside the patient. This situation is sketched in Fig. 6, for a simplified planar case of an instrument constrained by a fulcrum point.

In order to cancel out the external forces (red arrows in Fig. 6) the velocity which has to be provided by the controller has two components (blue arrows in Fig. 6):

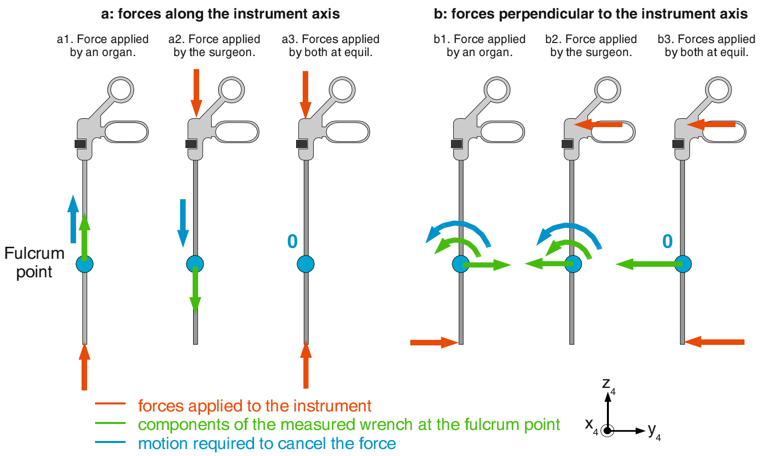

a: forces along the instrument axis
a1. Force applied by an organ.
a2. Force applied by the surgeon.
a3. Forces applied by both at equil.
b: forces perpendicular to the instrument axis
b1. Force applied by an organ.
b2. Force applied by the surgeon.
b3. Forces applied by both at equil.
Fulcrum point
0
0
forces applied to the instrument
components of the measured wrench at the fulcrum point
motion required to cancel the force
z4
x4
y4

Fig. 6. Forces, torques, and motions involved during co-manipulation.

the linear velocity ${}^4v_z$ along $\boldsymbol{z}_4$ (case shown in Fig. 6 a) and rotational velocity ${}^4\omega_x$ along $\boldsymbol{x}_4$ (case shown in Fig. 6 b). Additionally, the measured forces (two planar components ${}^4f_y$ and ${}^4f_z$) and torque at the fulcrum point (one component ${}^4t_x$) corresponding to the externally applied force are represented in green. Note that, in this example the wrench is measured at the fulcrum point, whereas for MC^2E this is not the case. Furthermore, the influence of the instrument weight on the measurements is neglected here (as they are compensated for in the experiments).

One can see in Fig. 6 a1 and Fig. 6 a2 that the velocity ${}^4v_z$ is always in the direction of the measured force ${}^4f_z$. Therefore, a simple proportional control law:

$$ {}^4v_z = k\,{}^4f_z \tag{19} $$

is appropriate for both cases. On the contrary, in Fig. 6 b1, the measured force ${}^4f_y$ and the cancelling velocity ${}^4\omega_x$ are both positive, while in Fig. 6 b2, they have an opposite sign. Therefore, selecting the force ${}^4f_y$ as a component to be servoed is not appropriate for the control of ${}^4\omega_x$: a positive (resp. negative) gain would lead to unstable behavior with respect to outer (resp. inner) forces. That is why the following control was selected:

$$ {}^4\omega_x = k\,{}^4t_x\ . \tag{20} $$

Indeed, ${}^4\omega_x$ and ${}^4t_x$ have the same sign (positive) in both Fig. 6 b1 and Fig. 6 b2. Finally, note that in comanipulation, when there is an equilibrium between the surgeon force and the organ force, the system stays still (see Fig. 6 a3 and Fig. 6 b3). This allows for the surgeon to feel the distal forces from the proximal grasp. Generalizing this reasonning to the 3 DoFs problem, the final control law for a zero desired force consists of computing the velocity command vector $\boldsymbol{u}$:

$$ \boldsymbol{u} = \begin{bmatrix} {}^4\omega_x \\ {}^4\omega_y \\ {}^4v_z \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} {}^4t_x \\ {}^4t_y \\ {}^4f_z \end{bmatrix}\ , \tag{21} $$

where $\boldsymbol{K}$ is a diagonal matrix of positive gains.

In order to apply this control law to MC^2E the wrench measured by the sensor $({}^s\boldsymbol{f}_s, {}^s\boldsymbol{t}_s)$ given in frame $\mathcal{F}_s$ at point S has to be gravity compensated, which gives the contact situation (*i.e.* the external wrench)

$$\begin{bmatrix} {}^s\boldsymbol{f}_c \\ {}^s\boldsymbol{t}_c \end{bmatrix} = \begin{bmatrix} {}^s\boldsymbol{f}_s \\ {}^s\boldsymbol{t}_s \end{bmatrix} - \begin{bmatrix} {}^s\boldsymbol{f}_m \\ {}^s\boldsymbol{t}_m \end{bmatrix} , \tag{22}$$

and then transformed in the frame $(O, \boldsymbol{x}_4, \boldsymbol{y}_4, \boldsymbol{z}_4)$ located at the fulcrum point O:

$${}^4\boldsymbol{h}_c = \begin{bmatrix} {}^4t_x \\ {}^4t_y \\ {}^4f_z \end{bmatrix} = \boldsymbol{H}_1 \begin{bmatrix} {}^s\boldsymbol{f}_c \\ {}^s\boldsymbol{t}_c \end{bmatrix} , \tag{23}$$

$({}^s\boldsymbol{f}_m, {}^s\boldsymbol{t}_m)$ being the model based gravity compensation and the transformation matrix

$$\boldsymbol{H}_1 = \begin{bmatrix} 000100 \\ 000010 \\ 001000 \end{bmatrix} \begin{bmatrix} {}^4\boldsymbol{R}_s & \boldsymbol{0}^{3\times 3} \\ {}^4\boldsymbol{R}_s[\boldsymbol{d}_{SO}]_\times & {}^4\boldsymbol{R}_s \end{bmatrix} . \tag{24}$$

In order to be able to specify a desired force ${}^4\boldsymbol{f}_d = [{}^4f_{d,x}, {}^4f_{d,y}, {}^4f_{d,z}]^T$ at the instrument tip P, one needs to compute the corresponding desired components for the servoed vector ${}^4\boldsymbol{h}_d$:

$${}^4\boldsymbol{h}_d = \begin{bmatrix} {}^4t_{d,x} \\ {}^4t_{d,y} \\ {}^4f_{d,z} \end{bmatrix} = \begin{bmatrix} 0 & -d_4 & 0 \\ d_4 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^4f_{d,x} \\ {}^4f_{d,y} \\ {}^4f_{d,z} \end{bmatrix} = \boldsymbol{H}_2 \begin{bmatrix} {}^4f_{d,x} \\ {}^4f_{d,y} \\ {}^4f_{d,z} \end{bmatrix} . \tag{25}$$

The final control law, accounting for the desired force vector is:

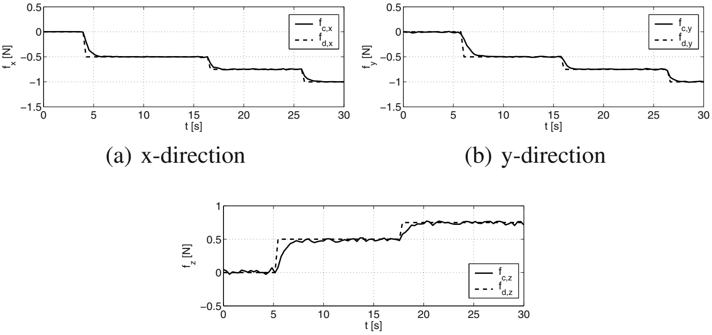
$$\boldsymbol{u} = \boldsymbol{K} \left(\boldsymbol{H}_1 \begin{bmatrix} {}^s\boldsymbol{f}_c \\ {}^s\boldsymbol{t}_c \end{bmatrix} - \boldsymbol{H}_2 \begin{bmatrix} {}^4f_{d,x} \\ {}^4f_{d,y} \\ {}^4f_{d,z} \end{bmatrix} \right) . \tag{26}$$

Thereafter, the command vector $\boldsymbol{u}$ is converted into the desired instrument tip velocity ${}^0\boldsymbol{v}_d$, expressed in the robot base frame $\mathcal{F}_0$ by

$${}^0\boldsymbol{v}_d = {}^0\boldsymbol{R}_4\boldsymbol{H}_2^T\boldsymbol{u} = \boldsymbol{H}_3\boldsymbol{u} \tag{27}$$

and then integrated to the desired position ${}^0\boldsymbol{x}_d$. An iterative inverse model is used to compute the desired joints $\boldsymbol{\Theta}_d$ of the robot which are passed to the robot joint controller. As the robot moves, the instrument touches the environment (here modeled as spring $\boldsymbol{K}_s$) located at $\boldsymbol{x}_e$.

Experimental results are given in Fig. 7 for ${}^4f_{c,x}$, ${}^4f_{c,y}$, and ${}^4f_{c,z}$. It can be seen that the desired contact forces are achieved, whithout a remaining offset. Nevertheless, a significant rise time occurs, mainly due to the soft environment used as a subsitute for real organs.



(a) x-direction

(b) y-direction

(c) z-direction

Fig. 7. Trajectories of the desired (${}^4f_d$) and measured (${}^4f_c$) contact force in tool tip frame $\mathcal{F}_4$.

5 Conclusions and Outlook

In this paper a compact and lightweight robot for force control in MIS is presented. This robot posesses an invariant point due to its kinematics and is mounted on the patient. A new trocar with an integrated force sensor allowing for the measurement of contact forces is described. Although this sensor is placed outside the patient friction inside the trocar does not deteriorate the measurements. Experimental force control results are given, validating the chosen concepts. Future work includes the setup of a force reflecting telemanipulation system for MIS, which will provide a realistic impression of the remote forces to the surgeon.

References

1. Russell H. Taylor and Dan Stoianovici. Medical robotics in compter-integrated surgery. *IEEE Transactions on Robotics and Automation*, 19(5):765–781, 2003.
2. C. Wagner, N. Stylopoulos, and R. Howe. The role of force feedback in surgery: Analysis of blunt dissection. In *Proc. of the 10th Annual Haptics Symposium*, March 2002.
3. B. Deml, T. Ortmaier, and H. Weiss. Minimally invasive surgery: Empirical comparison of manual and robot assisted force feedback surgery. In *Proceedings of the 4th International Conference EuroHaptics 2004*, Munich, Germany, June 2004.
4. T. Ortmaier. *Motion Compensation in Minimally Invasive Robotic Surgery*. VDI Verlag, 2003. PhD Thesis.
5. U. Seibold and G. Hirzinger. A 6-axis force/torque sensor design for haptic feedback in minimally invasive robotic surgery. In *Proceedings of the 2nd VDE World Microtechnologies Congress*, Munich, Germany, October 2003.
6. P. Berkelman, E. Boidard, P. Cinquin, and J. Troccaz. LER: The light endoscope robot. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2003*, Las Vegas, USA, October 2003.
7. Yuki Kobayashi, Shingo Chiyoda, Kouichi Watabe, Masafumi Okada, and Yoshihiko Nakamura. Small occupancy robotic mechanisms for endoscopic surgery. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002: 5th International Conference, Proceedings, Part 1*, Tokyo, Japan, September 2002.

Modeling and Experiments on a Legged Microrobot Locomoting in a Tubular, Compliant and Slippery Environment

Paolo Dario, Cesare Stefanini and Arianna Menciassi

CRIM Lab
Polo Sant'Anna Valdera
Scuola Superiore Sant'Anna
Viale Rinaldo Piaggio 34 - 56025 - Pontedera (Pisa) - Italy
{dario, c.stefanini, arianna}@sssup.it

Abstract. This paper presents the concept and preliminary modeling of a legged microrobot locomoting in a tubular, compliant and slippery environment. The application field is the one of endoscopic microcapsules intended to navigate inside the gastrointestinal tract for diagnosis and therapy. After introducing and discussing the issue of autonomous locomotion of endoscopic devices, with reference to worldwide ongoing research in the field, the legged solution is proposed and peculiarities of this approach are described. The importance of simulation means for such a device is discussed and a modeling for this particular kind of locomotion is presented. Experimental results are also described for the definition of biomechanical parameters necessary in the model.

1 Introduction

Minimally invasive therapy and surgery (MIS and MIT) are more and more considered by medical doctors and especially by surgeons the true ongoing revolution in medicine. The typical goal for this discipline is to develop new microinstruments and endoscopes for direct access to areas never reached, or inadequately negotiated, by hand-operated tools. In particular the development of autonomous machines able to accurately act in regions whose typical dimensions are smaller than few millimetres would lead to dramatic improvements in the treatment of pathologies occurring in a large number of fields like, for example, the neurological, cardio-vascular, gastrointestinal, gynaecological, urological or articular sectors [1].

In addition to medical doctors, also scientists, engineers and even companies are investing resources in new technologies aimed at scaling down autonomous machines for the exploration of the human body, in a sort of futuristic – but more and more realistic – voyage inside our organism [2].

This paper addresses an application that the authors consider to be very promising for new diagnostic and therapeutic autonomous tools: active gastrointestinal capsular endoscopy. Several reasons make this area a good candidate for medical, micro-sized machines: first of all, from a clinical point of

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 165–174, 2006.

view, a large number of pathologies occur in the gastrointestinal tract, as in the case of gut cancers which, according to the National Cancer Institute [3], are the fourth most commonly diagnosed cancers and rank second among cancer deaths in the United States. Secondly, from an engineering viewpoint, the gastrointestinal tract is a relatively large endoluminal environment, where miniaturization of autonomous machines seems feasible. Finally, the gut does not require sterilization of ingestible parts, thus simplifying the design, fabrication, and packaging of artificial devices as in the case of robotic capsules.

On the other hand, endoscopy of the entire gastrointestinal tract (GIT) by using teleoperated microcapsules with active locomotion and with the ability to make on board diagnosis, therapy and biopsy is a very challenging goal, whose achievement depends on the technological development in many engineering fields (micromachining, microactuation, sensorization, localization, teleoperation, etc.). In particular active and controllable locomotion is the main feature which can in principle transform a high-tech miniaturized camera into a really useful device for endoscopic diagnosis and therapy. However, no practical solutions for locomotion in the gastrointestinal tract have been completely developed and a locomotion model in such an environment is still unexplored.

For these reasons authors have addressed the issue of understanding the peculiar locomotion of a microrobot inside a tubular, deformable and slippery substrate.

The paper is organized as follows. In paragraph 2 state of the art of autonomous and semi-autonomous endoscopy is presented. Paragraph 3 reports illustrates the motivations and characteristics of a legged capsule solution whose dimensions, in the order of one-two centimeters, are suitable for gastrointestinal application. In paragraph 4 theoretical modeling of the legged locomotion is presented and in paragraph 5 experiments done in order to tune model parameters are described.

2 Related work

The development of self-locomoting, semiautonomous microrobots able to move in tubular geometries has been addressed worldwide by many researchers: applications range from pipe inspection [4,5] to endoscopy [6-9].

Regarding completely autonomous devices available for medical inspection, they are still without locomotion capabilities. First examples of wireless monitoring systems and capsules appeared in 1957: endoradiosondes, also called “radio-pills”, were developed by Jacobsen et al. and they were used in many medical and biological studies [10]. In 1962, Nagumo et al. developed a passive echo capsule for temperature and pH monitoring [11].

In recent years the interest in endoscopic capsules increased significantly, with efforts from various research centers and commercial institutions. An autonomous image vision system with wireless data transmission, integrated in a small pill, received recently the approval for clinical evaluation in the US [12,13].

The system is equipped with a CMOS sensor, a transmitter, LEDs for illumination and watch-like batteries. The main engineering limitation of this device is related to the lack of an active control of the locomotion: the pill proceeds by normal peristalsis and it cannot be stopped during its journey.

This problem should be partially overcome by a similar device, under development by RF Norika (Nagano, Japan), which uses a wireless power transmission and possesses active rotation capabilities thanks to the interaction of electromagnetic fields generated by a wearable jacket [14]. On the other hand, no detailed information about the pill performance are provided at the time of this paper and working prototypes are not generally available.

3 A legged microrobot for capsular endoscopy

An effective locomotion in a slippery and deformable substrate – such as the human gut – requires the harmonic integration of two equally critical systems respectively dedicated to:

- achieve contact with the tissue so that locomotive forces can be transmitted (by adhesion, mechanical clamping or other means);
- displace the contact points so that locomotion can be produced.

The dimensioning of both subsystems must take into account biomechanics of the tissue, which is an extremely compliant non-linear viscoelastic material typically covered with a thick (up to 2 mm) layer of lubricant mucus, with a friction coefficient as low as 10^{-3} [15]. Large deformations produced onto the tissue may dramatically degrade the performance of the locomoting system.

In the semiautonomous colonoscope previously developed by P. Dario et al. [16,17] the two subsystems were based, respectively, on suction + mechanical

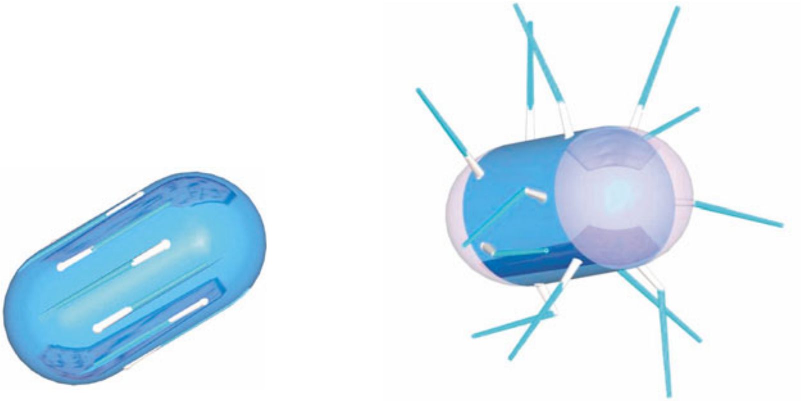


Fig. 1. Legged capsule before (left) and after (right) leg deployment. The first configuration is suitable for ingestion, thanks also to an external thin coating which dissolves in the stomach, exploiting pH sensitivity to gastric juices.

clamping and on inchworm movements. In particular, vacuum was used in order to attract the tissue between the jaws of the mechanical clamp. Although an inchworm system may appear a suitable solution also for the locomotion of a 3 cubic centimeters capsule, the authors carefully analyzed how some limitations of the inchworm concept could affect locomotion in the microscale. By taking into account these considerations, the devised solution consists of reversing the point of view: instead of forcing the tissue onto the adhesion/clamping area, this area is pushed against the tissue, by mounting it at the end of deployable micro-legs.

The legged capsule solution (Fig. 1) is based on a series of micro legs [18], whose primarily goal is to achieve clamping/locomotion/turning/rotation while giving also the opportunity to perform other tasks, such as biopsy, liquid sampling, etc.

A legged locomotion is for sure more technically demanding in terms of mechanical hardware and control strategy than inchworm. Nevertheless, there are undoubtedly peculiar advantages over the strategies followed so far that justify the selection of this solution, including:

- Increased safety and control: in the inchworm locomotion the body of the capsule slides over the tissue, with none or small ability to avoid injuries or pathologic areas. By exploiting legs, a better control on the trajectory is allowed and the capsule may pass over critical areas without touching them.
- Better adaptability to the environment: thanks to legs, the capsule is suitable to operate in anatomically and biomechanically different areas (stomach, small and large intestine, featured by different average diameters).
- Higher velocity: no cinematic amplification of stroke occurs in inchworm locomotion. If a linear actuator provides a stroke ΔL, the maximum displacement of the capsule will be less than ΔL (locomotion efficiency is always less than 1 because of the biomechanics of tissue). Higher velocities can be achieved only by using adequate stroke amplification mechanisms, as in the case of legs, acting as levers.
- Simplified adhesion: by localizing the contact points in small areas (tip of each leg) higher contact pressure can be reached, thus producing significant local deformation. Since the main exploitable friction mechanism in the GI tract is based on hysteresis [15], high friction coefficient can be reached in the contact points.

4 Theoretical modeling of the legged capsule locomotion

The behavior of a legged capsule in the intestine is a complex phenomenon and a theoretical modeling can provide important guidance rules for design. Literature demonstrates that this research field is not well investigated from the scientific point of view: several implementations of self-locomoting endoscopes exist but no theoretical studies are available. In particular there are no indications or evaluations on legged devices locomoting on highly compliant and slippery substrates, as in the case of the gastrointestinal tract. Research on legged microsystems exist, but sizes are much smaller than in the endoscopic capsule application and degrees of freedom of each leg are reduced in a way that those systems are more addressing "cilia-like" motion than legged robots [19].

The goal of this preliminary simulation is to obtain a tool able to visualize output motion and to compute the energy and force requirements of the legged device. The simulation of the legged capsule (and of the single leg in particular) has to strongly consider the environment of locomotion. Qualitative studies on the morphology and conditions of the intestine and quantitative measurements of the elasticity and frictional properties of the gut wall are required to model the interaction between legs and tissue and to "adjust" the motion laws of the legs.

Fig. 2 illustrates a robotic schematization of the legged capsule, which just as an example is 8-legged (in general it has **N** legs). The relative position of each leg tip with respect to the capsule can be computed using traditional robot manipulator theory, once joints coordinates are given. In this study authors considered the more general situation, in which an arbitrary, 3-DOF motion law is provided to the individual tip through active actuation. Only after obtaining results from simulation this assumption can be modified, restricting the degrees of freedom of leg tips to specific motions, depending on effectiveness of the simulated gait. Independent variables in the simulation are the relative coordinates of the tips, expressed by vectors $\mathbf{r}_i$.

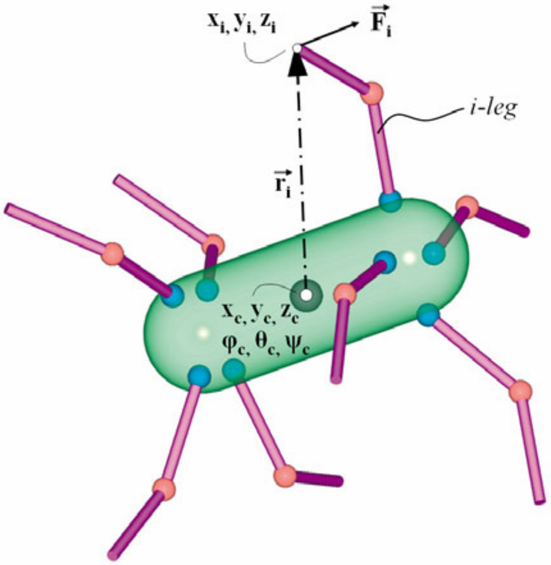


Fig. 2. Scheme of an 8-legged capsule with multiple degrees of freedom per leg. Spatial coordinates of each leg tip can be derived using direct kinematics once capsule coordinates (orientation, position) are known.

In modeling the legged capsule inside the GIT, the basic assumption which has been made is to consider the output motion of the capsule as a sequence of configurations characterized by static equilibrium (resultant force and moment equal to zero). This approach is well suited to miniaturized systems having low speed, as in the case of the legged microcapsule, where scaling effects make inertial forces negligible with respect to contact interaction.

Referring to Fig. 2, equations modeling the system are:

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = f_1(\vec{r}_i, x_c, y_c, z_c, \varphi_c, \theta_c, \psi_c) \qquad i=1,.., N \qquad (1)$$

$$\vec{F}_i = f_2(x_1,...,x_N, y_1,...,y_N, z_1,...,z_N) \qquad i=1,.., N \qquad (2)$$

$$\sum_{i=1}^{N} \vec{F}_i = 0 \qquad (3)$$

$$\sum_{i=1}^{N} \vec{F}_i \wedge \vec{r}_i = 0 \qquad (4)$$

where $\boldsymbol{f_1}$ is the known relation (composition of rigid motions) for obtaining the absolute position of each leg tip starting from its relative coordinates with respect to the capsule, while $\boldsymbol{f_2}$ is the experimental relation for determining the force due to the contact, which is investigated in paragraph 5 through the characterization of two main phenomena: the bio-elasticity and bio-tribology of the gut tissue. Equations (3) and (4) describe static equilibrium to translation and rotation. Unknown variables are the positions of leg tips ($\boldsymbol{x_i}$, $\boldsymbol{y_i}$, $\boldsymbol{z_i}$), the coordinates of the capsule ($\boldsymbol{x_c}$, $\boldsymbol{y_c}$, $\boldsymbol{z_c}$, $\boldsymbol{\varphi_c}$, $\boldsymbol{\theta_c}$, $\boldsymbol{\psi_c}$) and the forces $\boldsymbol{F_i}$ arising at the tips. Therefore the number of scalar unknown is equal to 6+6N, the same than scalar equations. Leg tips configuration, described by vectors $\boldsymbol{r_i}$, are input data, whose dependence on time makes the gait pattern.

Regarding gut modeling, the selected approach is based on the schematization of an intestine segment as a tubular viscoelastic element with pure membrane strains and stresses. A geometrical description of the intestine tract has been done using Matlab^{TM} software. The in-built function "convhulln", able to find the convex hull of a point set, has been adopted. The resulting tool (partially still under-construction) will be used for the design synthesis, for defining the optimal number of legs and the best number of degrees of freedom for each leg. It will be also used for controlling the device, by identifying the gait pattern to be adopted to obtain different capsule movements. Fig. 3 shows the modeling of an intestine tract deformed by an eight-legged capsule. Each leg is identified in the simulation with its tip (a point, external to the capsule, capable of moving along the three spatial directions). Frames have been extracted from a simulation generated for a

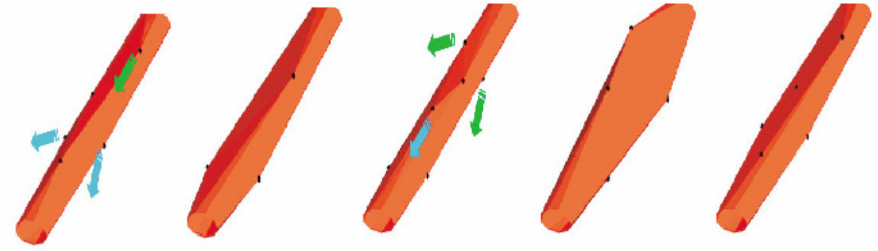


Fig. 3. Matlab^{TM} modeling of the gut tissue being deformed by an 8-legged capsule moving inside the GI tract with a forward gait pattern.

gait pattern where all the legs move with an elliptic motion in forward direction. Small spheres represent leg tips, while arrows indicate those legs (front and back legs) generating motive force.

Inverse kinematics for the computation of the joint coordinates in the leg can be done for the optimal capsule design. Performances, such as power required for locomotion, energetical efficiency associated with gait patterns and capsule displacement, can be derived thanks to the knowledge of forces arising at each leg tip.

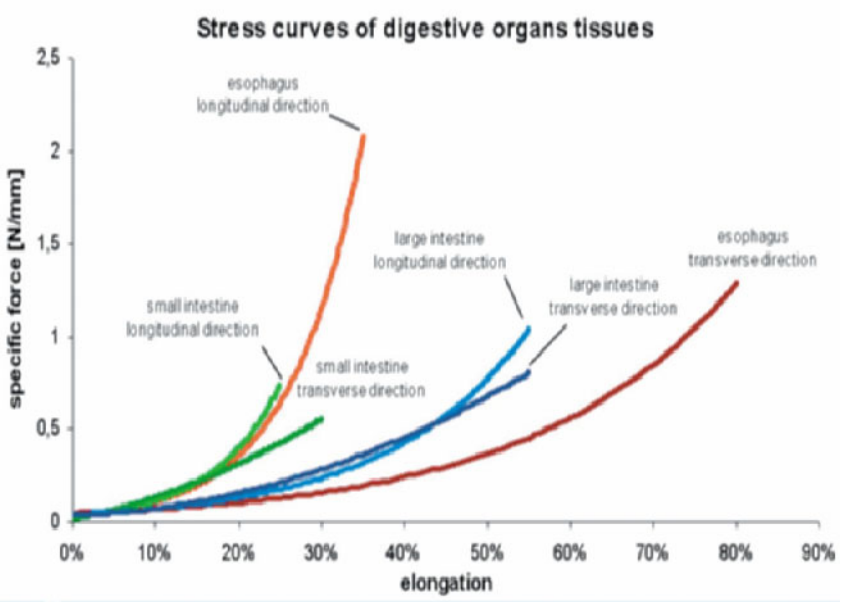


Fig. 4. Force / strain relation for various gastrointestinal tissues in the two directions (longitudinal and transversal), for bio-elasticity characterization.

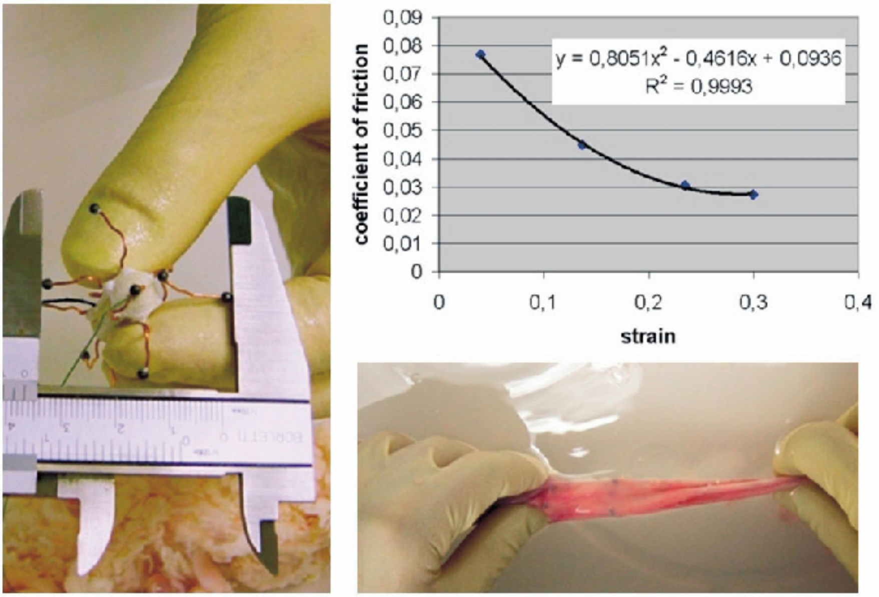


Fig. 5. Ex-vivo tests on bio-tribological interaction between capsule legs and gastrointestinal tissue.

5 Experimental tests for model tuning

As pointed out in the previous paragraph, the critical aspect to be considered in the model is the interaction between the leg tip and the gut tissue. Forces exchanged by the two bodies arise because of the stretching of the tissue (normal component) and because of friction (tangential component).

Regarding the first aspect force is caused by the bio-elastic response of the tissue being deformed. In particular both circumferential and longitudinal enlargement due to leg extension have been considered in the model, and tests have been carried on for the determination of stress/strain relation. Fig. 4 shows the results of this characterization. In the experimental apparatus, strips of tissue have been put in traction and strain has been detected using optical sensors, while force has been measured thanks to a load cell. The elastic response is expressed in N/mm because it refers to the force given by the tissue per unit of width.

Concerning friction, a capsule mock-up has been built as shown in Fig. 5. This device has been used in an ex-vivo gastrointestinal tract using different legs apertures, that is for various tissue strains. The capsule has been pulled in the tubular structure and also in this case the pulling force has been measured using a load cell. With this configuration, tangential forces have been directly measured, while normal forces acting on the legs have been estimated thanks to the results of the previous tests and to the known strain. The resulting curve, also shown in Fig. 5, allows to estimate the friction coefficient for different strain conditions.

6 Results and Conclusion

In this paper authors presented a new concept of robotic locomotion for biomedical application, together with the description of a tool to be used for simulation. Theoretical and computational methods have been illustrated, and an experimental characterization of the interaction between the device and the tissue has been done in order to provide the simulation tool with realistic parameters.

The main result is the modeling of the interaction between a locomoting legged robot and a tubular, compliant and slippery substrate, where coefficients of friction in the order of 10^{-3} are involved. Future work will address the completion of the tool and extensive simulations for the identification of optimal gait patterns maximizing performances and minimizing power consumption.

Acknowledgment

Authors would like to thank the Korean government and the Intelligent Microsystem Center (IMC) for promoting the Intelligent Microsystem Program (IMP) and supporting the participation of the Scuola Superiore Sant'Anna in the Endoscopic Microcapsule Locomotion Project (EMILOC). They would also like to thank Dr. M.O. Schurr for his precious advice in ex-vivo experiments.

References

[1] Samii, M., in a talk given at the 2nd Congress of the Italian Society for Computer Aided Surgery, Naples, November, 17th, 2000.

[2] Bowersox, J.C., "Bits and Bytes of 21st Century Medical Care". The Pfizer Journal, vol. 5, n. 2, 2001, pp. 20-25.

[3] Miller, B.A., L.N. Kolonel, L. Bernstein, J.L. Young, G.M. Swanson, D. West, C.R. Key, J.M. Liff, C.S. Glover and G.A. Alexander, "Racial/Ethnic patterns of cancer in the United States 1988-1992". National Cancer Institute, NIH Pub. No. 96-4104: Bethesda, MD, 1996.

[4] Wakimoto, S., J. Nakajima, M. Takata, T. Kanda and K. Suzumori, "A micro snake-like robot for small pipe inspection". Proceedings of the International Symposium on Micromechatronics and Human Science (MHS), 2003, 19-22 Oct. 2003, pp. 303-308.

[5] Suzumori, K., T. Miyagawa, M. Kimura and Y. Hasegawa, "Micro inspection robot for 1-in pipes". IEEE/ASME Transactions on Mechatronics, Vol. 4, no. 3, Sept. 1999, pp. 286-292.

[6] Ikuta, K., M. Tsukamoto and S. Hirose, “Shape memory alloy servo actuator system with electric resistence feedback and application for active endoscope”. Proc. of the 1988 IEEE Int. Conf. Robot. Automat., Philadelphia, Pennsylvania, April 1988, pp. 427-430.

[7] Phee, S.J., W.S. Ng, I.M. Chen, F. Seow-Choen and B.L. Davies, “Locomotion and steering aspects in automation of colonoscopy”. IEEE Engineering in Medicine and Biology Magazine, vol. 16, no. 6, 1997, pp. 85-96.

[8] Menciassi, A., and P. Dario, “Bio-inspired solutions for locomotion in the gastrointestinal tract: background and perspectives”. Philosophical Trans. Royal Soc. A Vol. 361, No. 1811, October, 2003, pp. 2287-2298.

[9] Dario, P., M.C. Carrozza and A. Pietrabissa, “Development and in vitro tests of a miniature robotic system for computer assisted colonoscopy”. Journal of Computer Aided Surgery, vol. 4, 1999, pp. 4-14.

[10] Nebeker, F., “Golden accomplishments in biomedical engineering”. IEEE Engineering in Medicine and Biology Magazine, vol. 21, May-June 2002, pp. 17-47.

[11] Nagumo, J. et al., “Echo capsule for medical use (a batteryless endoradiosonde)”. IRE Trans. Bio-Med. Electronics, BME-9:195, 1962.

[12] Meron, G., “The Development of the Swallowable Video Capsule (M2A)”. Gastrointestinal Endoscopy, vol. 6, 2000, pp. 817-819.

[13] Iddan, G., G. Meron, A. Glukhovsky and P. Swain, “Wireless capsule endoscopy”. Nature, vol. 405, 2000, p. 417.

[14] Uehara A. and K. Hoshina, “Capsule endoscope NORIKA system”. Minimally Invasive Therapy & Allied Technologies, Vol. 12, No. 5, Sept. 2003, pp. 227-234.

[15] Accoto, D., C. Stefanini, L. Phee, A. Arena, G. Pernorio, A. Menciassi, M.C. Carrozza and P. Dario, “Measurements of the frictional properties of the gastrointestinal tract”. World Tribology Congress 2001, Vienna, Austria, 3-7 September 2001, p. 728.

[16] Phee, L., D. Accoto, A. Menciassi, C. Stefanini, M. C. Carrozza and P. Dario, “Analysis and Development of Locomotion Devices for the Gastrointestinal Tract”. IEEE Transactions on Biomedical Engineering, Volume: 49 Issue: 6 , Jun. 2002, pp. 613-616.

[17] Dario, P., P. Ciarletta, A. Menciassi and B. Kim, “Modeling and Experimental Validation of the Locomotion of Endoscopic Robots in the Colon”. International Journal of Robotic Research, Vol. 23, No. 4-5, Apr.-May 2004, pp. 549-556.

[18] Dario, P., D. Accoto, S. Gorini, A. Menciassi, G. Pernorio, C. Stefanini, “Capsula Robotica per Applicazioni Biomediche Intracorporee”. Italian Patent pending (deposited on 16-02-2004).

[19] Tabata, O., H. Kojima, T. Kasatani, Y. Isono, and R. Yoshida, “Chemo-mechanical actuator using self-oscillating gel for artificial cilia”. The IEEE Sixteenth Annual International Conference on Micro Electro Mechanical Systems, 2003 (MEMS 2003), Kyoto, 19-23 Jan. 2003, pp. 12-15.

Quasi Hands Free Interaction with a Robot for Online Task Correction

A. Casals[1], M. Frigola[1], J. Amat[1] and E. Laporte[2]

[1] Technical University of Catalonia (UPC), Center of Research in
Biomedical Engineering (CREB), Barcelona, Spain
alicia.casals@upc.es
[2] Consortium Parc Taulí, Sabadell, Spain
elaporte@cspt.es

Abstract. New human machine interfaces are becoming more and more necessary in robotics applications where the robot operates in close cooperation with a human, and one near to each other. Many of these applications require intuitive interaction systems since they are used by non robotic experts, thus the use of natural language based interfaces (either voice or gestures) are highly convenient. This research tackles the problem of interpreting the user arm and hand gestures as orders to the system, while the user is working close to the robot. The proposed human-machine interface looks for a compromise between classical interfaces and the hands free operation requirements.

1 Introduction

The continuous progress towards service and personal robots, in applications that must operate in slightly structured environments and in close contact with humans, demands for new solutions for human robot interaction. Teleoperation and cooperative human-robot systems are evolving at a speed similar to that suffered by classical preprogrammed robots some decades ago, that is, as the industrial robots operating in production systems.

The need to interact with robots requires the development of new programming techniques. The capability of perceiving human intention and to respond to “natural” commands constitutes the means to spread such robotic systems for its use by non robotic specialists [1].

Medical robotics, and more specifically surgical robots constitute a target sector for these robotics improvements. In robot assisted surgery some kind of interventions can be performed, in part, autonomously after a preplanning phase. This is especially true in orthopedic surgery, where it is possible to model the patient’s, or the part to be intervened, from its 3D reconstruction from CT images. Unfortunately, many other tasks are less susceptible to be performed autonomously due to the deformability of the organs to be intervened, or their movement when it is not possible or desirable to fix the patient. In any case, a friendly and efficient human-robot interface is essential.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 175–184, 2006.

With the aim to improve the human robot interaction in surgical environments, we propose the use of a 3D vision system that behaves as a virtual exoskeleton [2], that is, a new interface for the control of robots and other equipment in an Operating Room (OR). This interface tries to overcome the limitations of the most commonly used human-robot interaction means for their use in an O.R. A qualitative evaluation of the performances of different human - machine interfaces in the O.R. is shown in fig.1, where the two of them marked with an asterisk show underlined some desired performances in medical environments.

In this context, we define a quasi hands free interface, a system based on the interpretation of the hand gestures, which allows commanding the robot without the need of touching a screen, to modify the robot position, the planned trajectory or to manipulate other devices that force the surgeon taking-off the surgical gloves.

Operation with deformable parts or organs implies the need of taking some references to navigate adequately. Some aids can be provided visually, based on augmented reality systems. For instance, in [3], synthetic structures of the heart are overlaid over the images so as to aid the surgeon to know the instruments positions, with respect to relevant elements, and move them accordingly. Working with non fixed non deformable parts implies the need to register the previously reconstructed 3D model of the anatomical structure with the patient current position and orientation [4], as well as the need to define the constrains in the preplanning phase [5] .

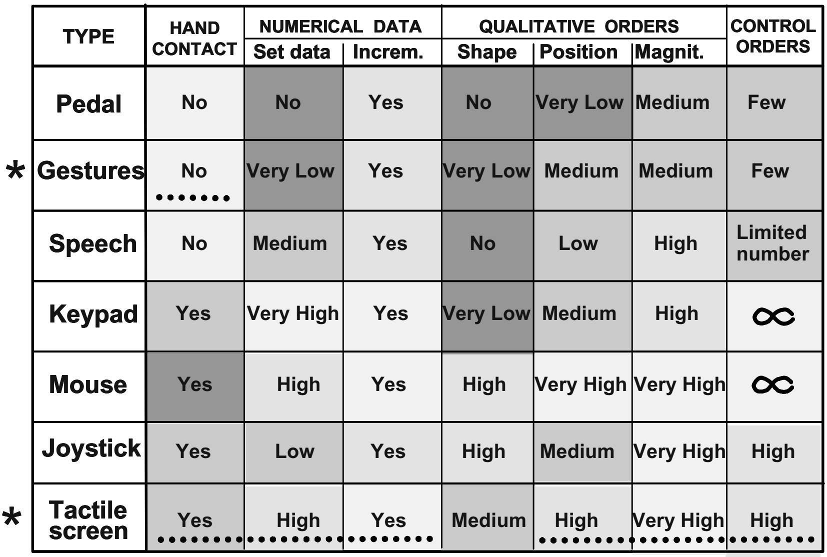
In another different surgical field, in laparoscopic surgery, continuous efforts have been devoted to provide robotics aids in positioning and holding the camera, the laparoscope [6, 7, 8] and in reducing the operating time, either when operating *in situ* or at distance [9]. This kind of surgery has also motivated the research in sensing, so as to provide the surgeon with perception feedback, to better perceive what happens in the operating area [10].

Such surgical robots require not only the capability to be programmed intuitively, in a natural way, but also to enable the surgeon to interactively change the previously planned task or actions to perform. With this aim, in [11] a strategy to deviate the robot from its trajectory or changing the operation strategy is described. It is based on the user's physical interaction with the robot to adapt the programmed task to the new and changing working conditions.

The work developed is part of the research program going on in the experimental Operating Room (O.R) of the hospital of the Parc Taulí. In this program, some efforts are devoted to provide an improved interface aimed to assist the surgeon to operate with the developed robotic systems.

The aim of this research is to provide the surgeon with a quasi hands free interaction device to modify the robot position, the predefined cutting plane or the movement constraints defined by planes.

For instance, in laparoscopic surgery, the surgeon can control the position of the camera directly, by him/herself, and quasi hands free, avoiding the use of voice with the boring and difficulties that such interfaces produce in an O. R. In orthopedic surgery, the surgeon can change the intervention strategy defined in the planning phase, as for instance movement constraints, planned path and so.



TYPE	HAND CONTACT	NUMERICAL DATA		QUALITATIVE ORDERS			CONTROL ORDERS
		Set data	Increm.	Shape	Position	Magnit.	
Pedal	No	No	Yes	No	Very Low	Medium	Few
* Gestures	No	Very Low	Yes	Very Low	Medium	Medium	Few
Speech	No	Medium	Yes	No	Low	High	Limited number
Keypad	Yes	Very High	Yes	Very Low	Medium	High	∞
Mouse	Yes	High	Yes	High	Very High	Very High	∞
Joystick	Yes	Low	Yes	High	Medium	Very High	High
* Tactile screen	Yes	High	Yes	Medium	High	Very High	High

Fig. 1. Performances of Human-Machine interfaces in the Operating Room.

2 Interface limitations and requirements

Since one of the biggest requirements in surgery is the need to keep the attention towards the intervention, it is very important to have available interfaces that enable the surgeon to interact with the robotic system with the minimum effort. The convenience to operate hands free is another constraint in the design of such interfaces.

For these reasons, many robotic systems in the surgical environment rely on voice commands, pedals, or tactile screens (Fig. 1). The use of multimodal systems contributes to alleviate the limitations of each of them [12].

Some current robotic systems use oral commands in the OR, as a hands free communication means. Such interface is adequate for interactions requiring imperative orders, such as standby, run, stop, left, right, or decrement, increment, but it is very difficult to define a 3D position, a task that is relatively common in many graphical interfaces.

On the other side, pedals are very useful, but only when dealing with few and concrete orders (on-off type).

Tactile screens enable the surgeon to interact with a menu, and thus provide good flexibility. Currently, tactile screens constitute the more powerful interfaces in the operating room, since they provide the means to access to all the equipment and devices in this environment. Operating over a digital screen, the surgeon can select and activate or control all the equipment, with the same efficiency resulting from acting directly over push buttons, switches etc. Nevertheless, touching a screen is not desirable for cleanness and sterilization reasons.

The proposed gesture based interface tries to find a good compromise among the existing systems, avoiding the above mentioned limitations, and requiring only the hands movements, without disturbing much the sequence of actions performed by the surgeon, those which are involved in the surgical procedure.

3 Gesture based interface

The purpose of developing this interface is to achieve the capability of controlling the elements of the environment through the perception, by vision, of the user's "natural" gestures. The user interact with a menu, through a screen, giving orders such as: press a click, dragging the pointer over the screen or increase or decrease a magnitude.

The vision system developed has been designed so as to achieve the maximum security and efficiency. For this reason, the system combines two stereo vision modules with different configurations. Firstly, it is expected that the system detects and interprets the hands movements performed in front of the screen. Secondly, a panoramic view visualizes the whole person interpreting its gestures.

The first module consists of two stereo cameras placed close to the computer screen, side by side, so as to focalize the hands when they approach the screen, trying at the same time to minimize the number of additional elements to be detected on the images. The second stereovision module, panoramic view, has also a long baseline to achieve enough resolution in the 3D positioning of the body. A long baseline provides high resolution in depth computation, nevertheless, at the same time it decreases the number of matches between the homologue points in the two images. The causes of missing these matches are the occlusions and the perspective deformation that affects differently the two images. Consequently, the identification of points or features on each stereo image does not always result in a right matching between the two images.

A compromise to deal with these opposite performances relies on the use of the above mentioned two vision systems that complement one to each other in a hierarchical way. The structure of the system is shown in fig. 2.

The first module detects the hands, when they approach the screen but without producing a physical contact. The system detects the user's movements starting at the finger tip and advancing to the hand and upwards the arm. On the other side, the panoramic vision module detects the head and body, and then follows to the arms, and advances downwards up to the hands. The first module detects with good resolution the finger tip and hand, but the resolution decreases as the visualization moves upwards to the arm. The same happens with the vision of the body focalizing the head and trunk, operating with a decreasing resolution towards the hand. Therefore, the composition of the two vision systems provides a more robust detection, location and tracking of the hand and finger movements.

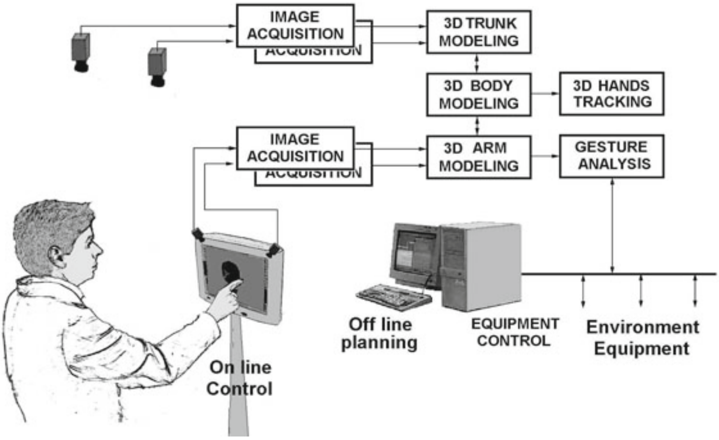


Fig. 2. Structure of the gesture based interface with its two complementary stereo vision systems.

3. 1 Panoramic vision system

In order to acquire in real time the user's posture (3D position of the head, trunk and arms) a procedure composed of simplified computer vision processes has been designed. The computer vision algorithm begins extracting some basic features from the movement analysis in a sequence of images. These image features are basically significant local curvature points of the edges, together with their associate 2D normal. Subsequently, the 3D positions of these features are estimated by triangulation using a coherence test to refuse false correspondences between the two stereo-images. The availability of fully calibrated cameras makes possible the rejection of incoherent correspondences. The quick rejection test used are the skew of the stereo lines (those defined by the singular points position and the camera's optical center) and edge normal comparison. Also, all the features far away enough from the estimated user's working area are removed together with all the 3D features located within the robot's volume. Fig. 3 shows a human holding an object and the image that has been extracted from its movement.

The head detection and the arm tip detection algorithms, as explained below, are based on a multiple hypothesis and fitting test scheme working in conjunction with a variable resolution geometrical body model. The algorithm begins generating a hypothesis of the model of the user's posture, using the 3D extracted candidate points, head or arm extreme points. A set of "adhoc" rules are used to limit the huge amount of possibilities that exist. These rules are based on a general standing posture (some head and body verticality is assumed), no arm tip consideration below the elbow and over the head, and arm tips no too close to the main body parts. Then, every posture hypothesis is compared with the other hypothesis to find the most coherent 3D posture. Here, a coherent human posture is considered better than others, when the 2D policylindrical model fitted over the human body leaves less moving edges uncovered in all camera images. Only two cylinders per arm are used in this phase.

After this coarse arm adjustment, a more detailed hand-forearm model is used to detect, one step further, the hand orientation. In this phase, the system tries to adjust two cylinders to the wrist joint with the same testing strategies as those mentioned before. However, hand posture hypothesis are restricted to those with the hand tip position located very close to the previously detected arm tip and with similar forearm orientation.

Due to the limitations imposed by the need to operate in real time and the poor image resolution of the hands thus obtained (according to the distances and field of views considered), only extended hand postures are considered. Nevertheless, this restriction does not constitute a real limitation because many significant gestures can be recognized despite these simplifications.

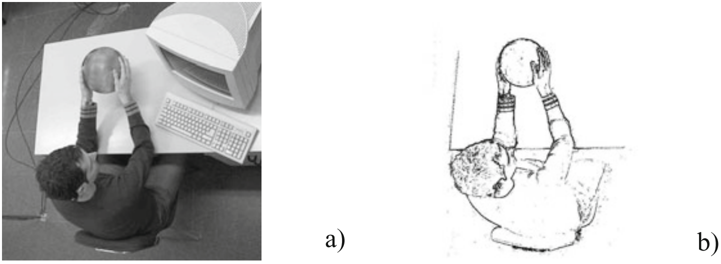


Fig. 3. Image of a human holding an object. a) Original image, b) image extracted from the user's movement.

3. 2. Finger tip acquisition

With the aim of using the own surgeon hand as a three-dimensional joystick, the proposed interface is based on the stereoscopic vision of the user's hand gestures in a working area located immediately in front of the computer screen. In this working volume, the indications or signaling performed by the user, using his or her own hand as a pointer device, are interpreted. Although the capability of the hand's gesture perception is limited, it is enough for the user to easily interact with the graphical interface. The fingers detection is very similar to the upper-body acquisition method. In a first step, the static elements in the scene, the background, is removed from the image through successive gradient images comparison. The images are taken at variable time intervals according to the dynamics of the scene [2]. The high sensibility of the movement detection process and the inability of the user to maintain the hand in a completely static position make it possible to obtain a correctly segmented hand. Afterwards, the binary image is swept so as to obtain the 2D coordinates of the silhouette detected from movement. From these ordered points list, the singular points of the image are detected.

The three-dimensional position of singular points is computed and those not belonging to a volume located in front of the monitor are rejected, keeping a security range of about 1 cm. in front of the monitor. This restriction avoids taking singular points from the monitor, those corresponding to points in the image on the screen, or possible reflections.

The computation of the position of the singular points is done by means of triangulation, since in this case the cameras are calibrated with respect to the size and position of the computer screen. To increase the robustness of the detected points, only the triangulated points with short disparity error are considered. In the same way, only those pairs of points that are of the same type and with coherent orientation are considered for triangulation. The use of stereovision techniques based on singular points, providing 3D spare images results in low cost systems achieving operating frequencies higher than 10 images per second using conventional computers.

4 Orders interpretation

The interface can behave as a pointing device (finger tip detection) or as a movement controller. For the first behavior, the finger tip is detected and located. For the second, since the shape of the hand can change significantly when it is empty or full with any part, the position and the shape of the hand is not analyzed but its movement. The vision system detects this movement and interprets the operator gestures from the analysis of a sequence of images (Fig. 4).

As a pointing device the finger act as the mouse that select any icon on the screen to give a specific order, for instance to modify the planned intervention during its execution (cutting plane, constraints…). The click function can be performed by measuring whether the fingertip position surpasses or not a given distance to the screen, as indicated by the plane in fig. 5.

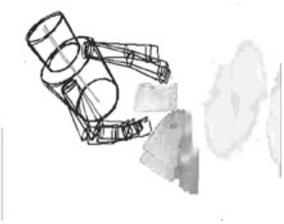


Fig. 4. Policylindrical model fitting the human body, extracted from a moving sequence.

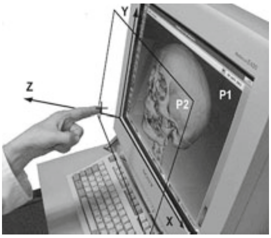


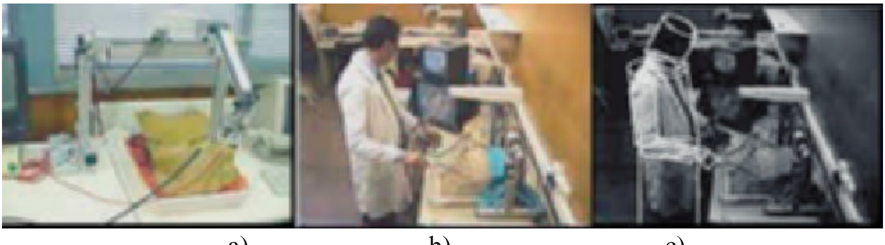
Fig. 5. The finger used as a pointing device emulating a mouse.

On the other hand, the analysis of the movement for the basic arm and hand gesture orders has given us some hints on how to extract information from the tracking of the hand. In this figure, a 3D reconstruction of a CT can be dragged or rotated from the finger indication. For instance, the order of moving rightwards or leftwards is usually done through a repetitive and alternative short movement of the hand. The trajectory of the hand is tracked allowing the perception of “natural” gestures, that is, those commonly used for warning or giving orders. Since the human movement is not quantitatively repetitive, the absolute measures of the detected displacements are not significant. Thus, the interpretation of the movement is done analyzing the signal spectrum and the number of repetitive elements. The fundamental component of the signal is a measure of the intensity of the movement while the sign of phase of the second harmonic gives the sense of the displacement of the slope of the detected displacement function. From this data a fuzzy controller enables to generate smooth movements from the operator imprecise orders (not accurate repetitive movements), or to activate some functions.

5 Results and Conclusions

The visual interface has been experimented in the laboratory in two different operating conditions. A first experimentation platform has been the quasi hands free control of the laparoscope. The technique used for giving the movement orders to the laparoscope is based on gesture recognition. This gesture based guidance is incremental; therefore, each change of the camera (laparoscope) point of view starts at the last holding position and can move with four degrees of freedom. The fuzzy controller receives the orders generated from the interpreted gestures extracted from the surgeon movements, which results in incremental movements represented in three orthogonal directions. The range of the movement is expressed by the surgeon either increasing the speed of the hand movements or with more repetitions of the orders movements. The strategy for the generation of the robotic holder control orders comes from the observation of the behaviors of multiple users. The analysis of this movement is performed over a simplified geometrical model of the thorax-arms-hands of the surgeon. The movements to be identified are the three displacements Δx, Δy and Δz and pitch rotation Δθ of the wrist.

Fig. 6 shows the surgeon working in this environment and the results of fitting a simplified policylindrical model over his body. These displacements are referred to a frame located in the camera point of view. The interpreted orders correspond to the surgeon hand movements, which can be free of objects or carrying a working tool. The detected movements can correspond to the current work, to spontaneous movements (replacement of the eyeglasses, the mask, the cap,…), or to the purposively given orders which have to be interpreted.



a) b) c)

Fig. 6. The interface for the control of a laparoscope with a robotic aid. a) the robot, b) the surgeon simulating an intervention, c) fitting of the policylindrical model over the surgeon.

Since an order is characterized by the repetitiveness of a movement, the orders are analyzed from their periodicity, from which a fuzzy magnitude is extracted from the period of the movement.

Fig 7 shows the traces obtained from the movement of the simplified model from the back and forth movements of a scrolling order, a lateral displacement; a and b correspond to the left and to the right scroll respectively. The three components, X, Y and Z are shown. The movement can be easily appreciated, first, the hand enters the cameras field of view, afterwards a back and forth repetitive movement, and finally, the hand leaves again the cameras field of view.

Fig. 8 shows the Discrete Fourier Transform of the component with the most significant movement, or amplitude; in this case X. From the spectral analysis we obtain the fundamental frequency, that corresponds to the speed of the operator's movement. The system interprets it as the magnitude of the performed action. The sense of this action, in this case rightwards or leftwards displacement, is obtained analyzing the shape of the signal. With this aim the phase of the second harmonic is obtained, that is positive or negative with respect to the fundamental (Fig. 9). The fundamental frequency and the second harmonic are indicated with a circle in both figures.

The patterns of different common movements have been extracted to distinguish some intuitive orders such as left, right, zoom, rotate, increase and decrease.

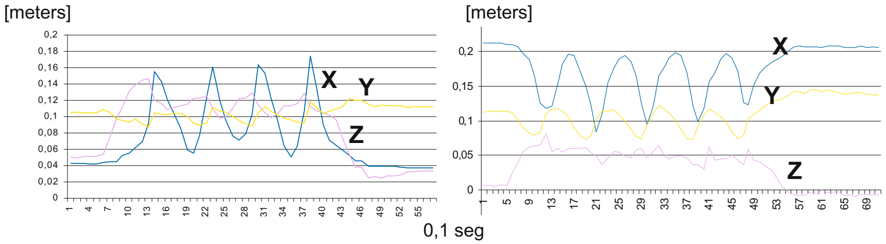


Fig. 7. The three trajectory components of the fingertip for a order scrolling. a) scrolling leftwards, b) scrolling rightwards

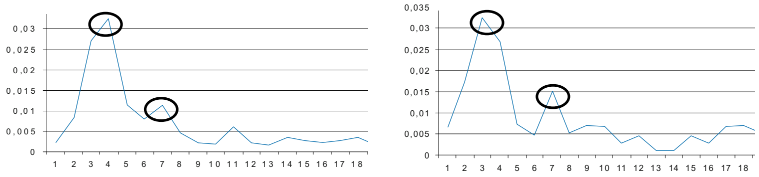


Fig. 8. Magnitude of the Discrete Fourier Transform of the X trace of fig. 5.

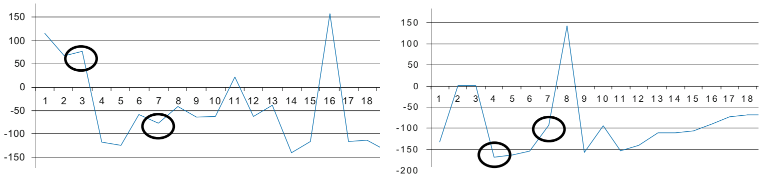


Fig. 9. Phase of the Discrete Fourier Transform of the X trace of fig. 5, that enables to determine the direction of the displacement.

References

[1] R. Dillmann, “Teaching and learning of robots tasks via observation of human performace”, *IROS Workshop on robot programming by demonstration*, Las Vegas, October 2003

[2] J. Amat, A. Casals, M. Frigola, J. Pagès, “Possibilities of man-machine interaction through the perception of human gestures”. *Contributions to Science,* Ed. Institut d’Estudis Catalans, Nº. 2 pp: 159-173

[3] F. Mourgues, T. Vieille, V. Falk and E.Coste-Manière, “Interactive guidance by image overlay in robot assisted coronary artery bypass” *Medical Image computing and Computer-Assisted Intervention- MICAII 2003*, pp: 173-181

[4] J. Ehrhardt et al, “Atlas-based recognition of anatomical structures and landmarks to support the virtual three-dimensional planning of hip operations”, *Medical Image computing and Computer-Assisted Intervention-MICAII 2003*, pp: 17-24

[5] P. Marayong, Ming Li, A.M. Okamura, G.D. Hager, “Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures,” *Proc. IEEE Int. Conf. Robotics and Automation*, Sept. 2003, pp. 1954-1959.

[6] A. Casals, J. Amat and E. Laporte, “Automatic guidance of an assistant robot in laparoscopic surgery” *IEEE Int. Conference in Robotics and Automation*, Minneapolis, April 1996, pp 859-900

[7] T. Ormaier and G. Hirzinger, “Cartesian control issues for minimally invasive robot surgery”, *IEEE Int. Conference on Intelligent Robost and Systems*, IROS 2000

[8] A. Nishikawa et al., “FAce MOUSe: A novel human-machine interface for controlling the position of a laparoscope”, *IEEE Transactions on Robotics and Automation,* Vol. 19, N. 5, pp: 825- 841

[9] S. E. Butner, M. Ghodoussi, “Transforming a robot for human telesurgery”, *IEEE Transactions on Robotics and Automation*, Vol. 19, N. 5, pp: 818- 824

[10] G. Tholey, J. P. Desai and A.E. Castellanos, “Evaluating the role of vision and force feedback in minimally invasive surgery: New automated laparoscopic grasper and a case study” *Medical Image computing and Computer-Assisted Intervention- MICAII 2003*, pp: 198-205

[11] M. Frigola, J. Poyatos, A. Casals and J. Amat, “Improving robot programming flexibility through physical human-robot interaction”, *IROS Workshop on robot programming by demonstration*, Las Vegas, October 2003

[12] S. Oviatt, “Flexible and robust multimodal interfaces for universal access”, *Int. Journal Universal Access in the Information Society*, pp: 91-95

VI. Manipulation

Manipulation is a primary function of a robot. To perform a manipulation task, the robot must be capable of interacting with the environment and with the object of interest. The tasks required are growing in complexity along with advances of robotic capabilities today. They are no longer confined in the walls of a factory as robots enter everyday human environments.

The first article by Harada and colleagues at AIST presents a mobile manipulation example by a humanoid. Manipulation by humanoids is a complex process requiring the coordination of not only the object of interest, but also the complex dynamics of the humanoid itself. The paper upped the stake by introducing manipulation of objects while the humanoid is moving.

In the second article, Pham and Chen propose a new model in the design and control of a flexure parallel mechanism based on pseudo-rigid body model. Flexure parallel mechanisms (FPM) have been rapidly developing in recent years. It possesses many advantages compared to conventional designs in ultra-precision manipulation system. In this paper, the authors successfully designed and optimized a flexure parallel mechanism for 3 DOF decoupled translational motion.

The third article by Padois and colleagues describes an experiment carried out on a mobile manipulator. On top of the usual modeling and control of the mobile manipulator, the authors introduced a high level task planning algorithm to decompose a human level command into lower level sub-tasks that can be executed by the mobile manipulator. This is an important area of growth that is necessary to make robots more interactive by bridging the difference in human high-level intelligence and the robotic lower level capabilities.

Lastly, Michelin and colleagues in LRIMM brings into medical robotics a classic algorithm of decoupling the task and posture dynamic control of a manipulator. In this case, the manipulator is the trocar for a minimally invasive surgery which needs to navigate through various bodily parts / organ. It is important that the task is satisfied while the posture of manipulator, resultant of its high degree of redundancy, is kept at the desired posture for the purpose of the surgery.

Integration of Manipulation and Locomotion by a Humanoid Robot

Kensuke Harada, Shuuji Kajita, Hajime Saito, Fumio Kanehiro,
and Hirohisa Hirukawa

Humanoid Research Group, Intelligent Systems Institute
National Institute of Advanced Industrial Science and Technology(AIST), JAPAN.
{kensuke.harada, s.kajita, f-kanehiro, hiro.hirukawa}@aist.go.jp
http://www.is.aist.go.jp/humanoid/

Abstract. This paper studies the integration of manipulation and locomotion by a humanoid robot. We will show that the walk and work capabilities of a humanoid robot can be enhanced by integrating manipulation and locomotion. Especially, we focus on three motions of a humanoid robot: pushing a heavy object placed on the floor, working while thrusting a hand on to a desk, and ascending large steps while holding a handrail for support. These motions are realized by experiment using the humanoid robot HRP-2.

1 Introduction

In recent years, several humanoid robots[1–3] have developed. They can walk on the flat floor, rough terrain with small gaps, and climb up stairs. On the other hand, a humanoid robot is expected to work instead of a human in a real environment. Since such an environment is not structured for a robot, it often includes rough terrain with large gaps and large obstacles. However, by actively using arms for propping the body, we can expect that a humanoid robot can extend walking and working capabilities.

For enhancing the walking and working capabilities, a humanoid robot can use two, three or four hands/legs as occasion requires. While we can consider many styles of integration of manipulation and locomotion, we focus on three styles of them: (i)pushing manipulation of a large and heavy object placed on the floor, (ii)handling task while supporting the body with an arm propping against a table, and (iii)climbing up a large gap while holding a handrail for support. These motions are considered to be essential when a humanoid robot works in a real environment.

From both theoretical and experimental aspects, we have been studying on the manipulation by a humanoid robot[7,9,8]. When a hand of a humanoid robot contacts an environment, one of the difficulties is how to keep balance of the robot. Conventionally, the balance control of bipedal walk in humanoid robot has been mostly ensured based on a reference index called ZMP, while the ZMP requires all the contact points lying on a plane. On the other hand, once the hands of a humanoid robot contacts an environment, it becomes difficult to judge whether or not the robot can keep balance. For such problem, we have been extended the ZMP analysis to manipulation tasks[7,9].

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 187–196, 2006.

Also, when a humanoid robot manipulates an object, the reaction force is applied at the hands. If the reaction force is large, it becomes difficult for a humanoid robot to keep balance. For such problem, we have been studied the force control based manipulation method[8].

On the other hand, this research studies the experiments on integration of manipulation and locomotion by a humanoid robot. We will focus on the practical aspects for realizing such integration. In this paper, after showing the previous works in Section 2, we show the experimental setup imitating a construction site in Section 3. The experimental results are shown in Sections 4, 5, and 6.

2 Related Works

Recently, several humanoid robots such as ASIMO[1], QRIO[2], johnnie[3], H7[4], and HRP-2[5]. As for the research on manipulation by a humanoid robot, Yokoyama et al.[10] realized by the humanoid robot HRP-2 carrying a panel with a human. Inoue et al.[11] determined the foot position maximizing the manipulability of the arms. Takenaka[12], the authors[6,8], Hwang et al.[13] also studied pushing task by a humanoid robot. The authors[7,9] also studied the ZMP applicable for the manipulation tasks.

3 Overview of the Realized Tasks

Fig.1 shows the overview of the experimental environment. This experimental environment imitates a construction site.

Pushing a Large Object:

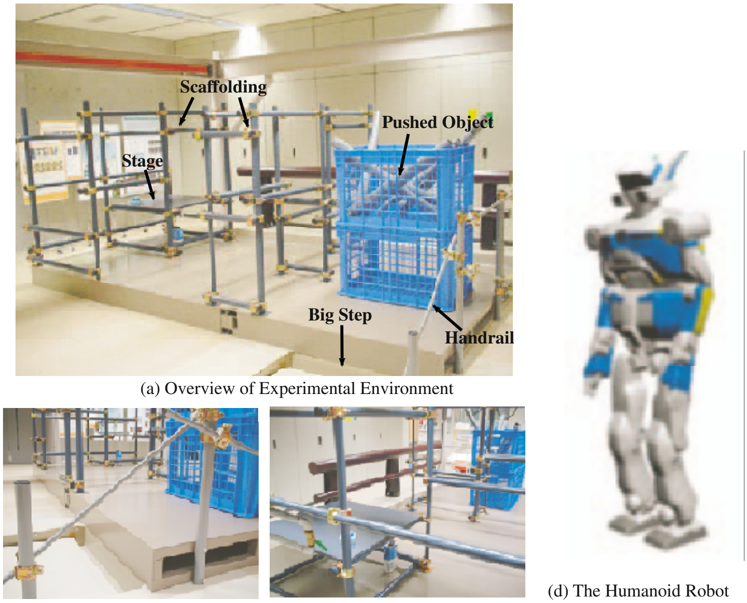
This experimental environment includes a large object. The weight of the object is 25.9[kg] which is about half of the robot's weight. The coefficient of static friction between the object and the floor is about 0.25. To manipulate the object, significant reaction force is applied at the hands. In this research, a humanoid robot will push and move the object taking the hand reaction force into account.

Handling Task:

As shown in Fig.1(a), there are structures made of steel pipes. These structures imitate the scaffolding which can be seen in a construction site. As shown in Fig.1(c), the structure includes a stage whose height is 50[cm]. Manipulation of an object placed on a stage is sometimes difficult for a humanoid robot. This is because, if the object is far from the robot, the humanoid robot cannot capture the object while standing on two legs. For such problem, we consider making a humanoid robot manipulate the object while supporting the body with an arm propping against the stage(Fig.2). In this experimental environment, the distance between the object and the robot is set to be 80[cm].

Climbing a Large Step:

As shown in Fig.1(b), the experimental environment also includes a large step whose height is 28[cm] and a handrail beside the step. For a human-sized humanoid robot, it has been difficult to climb up such a large step. There are two reasons for the



(a) Overview of Experimental Environment

(b) Step, Handrail, and Pushed Object

(c) Stage in the Scaffolding

(d) The Humanoid Robot HRP-2

Fig. 1. Experimental Setup

difficulty: (1)It becomes difficult to keep balance during the humanoid robot moves in the single support phase, and (2) It sometimes becomes difficult to generate the collision free motion within the movable range of each joint. As for the second reason, the humanoid robot H7[4] realized to climb a big step whose height is 25[cm] by utilizing the toe joint. In this research, a humanoid robot will climb up a large step whose height is 28[cm] while holding a handrail. By holding a handrail, we can expect that the stability of the robot would increase.

As a humanoid robot, we use the HRP-2[5] developed in the humanoid robotics project[17]. The total DOF of HRP-2 is 30 (2DOF Head, 2DOF Waist, two 6DOF Arms, two 1DOF Hands, and two 6DOF Legs), and the total weight of it is 58 [kg]. Since HRP-2 has 2DOF waist joint, various whole body motion can be generated. Some of the motions in this research are generated utilizing the waist joint.

From the next section, we show the experimental results of (i)pushing a large object, (ii)handling task, and (iii)climbing a large step.

4 Pushing a Large Object

Let the position of the ZMP, the COG(Center of Gravity) of the robot, and each hand be $p_{zmp} = [x_{zmp}\ y_{zmp}\ z_{zmp}]^T$, $p_G = [x_G\ y_G\ z_G]^T$, and $p_{Hj} = [x_{Hj}\ y_{Hj}\ z_{Hj}]^T$ $(j = 1, 2)$, respectively. Also, let the hand reaction force and the total mass of the robot be $\boldsymbol{f}_j = [f_{xj}\ f_{yj}\ f_{zj}]^T$ $(j = 1, 2)$ and M, respectively. The relationship between the ZMP position and the hand reaction force is expressed by[6]:

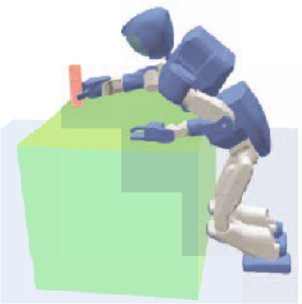


Fig. 2. Handling Task while Supporting Body with an Arm

$$x_{zmp} = \frac{-\dot{L}_{Gy} + Mx_G(\ddot{z}_G + g) - M(z_G - z_{zmp})\ddot{x}_G}{M(\ddot{z}_G + g)} - \sum_{j=1}^{2} \frac{(z_{Hj} - z_{zmp})f_{xj}}{M(\ddot{z}_G + g)}, \tag{1}$$

$$y_{zmp} = \frac{\dot{L}_{Gx} + My_G(\dot{z}_G + g) - M(z_G - z_{zmp})\ddot{y}_G}{M(\ddot{z}_G + g)} - \sum_{j=1}^{2} \frac{(z_{Hj} - z_{zmp})f_{yj}}{M(\ddot{z}_G + g)}, \tag{2}$$

where $\boldsymbol{L}_G = [L_{Gx}\ L_{Gy}\ L_{Gz}]^T$ denotes the angular momentum of the robot about the COG. In eqs.(1) and (2), the first term of the right hand side shows the ZMP position without considering the hand reaction force, while the second term shows the displacement of the ZMP position due to the hand reaction force. From the second term of the right hand side of eqs.(1) and (2), we can see that the lower the position of the COG be, the smaller the effect of hand reaction force onto the ZMP position becomes. We consider construct the motion of the humanoid robot withtout considering the hand reaction force. Then we consider modifying the motion of it according to the reaction force applied at the hands[6].

The robot can keep the dynamical balance if the ZMP is included in the foot supporting area. Here, we observed that the robot tends to be unstable if the contact force is applied at the hands in the single support phase. Therefore, we consider pushing the object during the double support phase where both of the feet contact the floor and where the convex hull of the foot supporting area becomes wide. And, the robot steps without pushing an object.

The experimental result of pushing manipulation is shown in Fig.3. First, a humanoid robot steps without pushing an object as shown in Fig.3(b). To reduce the effect of hand reaction force on the ZMP position, the humanoid robot squats down as shown in Fig.3(c). As shown in Fig.3(e) and (f), the humanoid robot pushes an object where the amount of displacement of the object is 20[cm]. Here, to compensate the error caused in the ZMP position due to the hand reaction force, the position of the feet is modified based on the second term of the right hand side of eqs.(1) and (2). After finished pushing an object, the robot steps back as shown in Fig.3(h) and (i). In this experiment, while the humanoid robot pushes the object only once, the experiment of pushing manipulation with continuously walking can be seen in the paper[6,8].

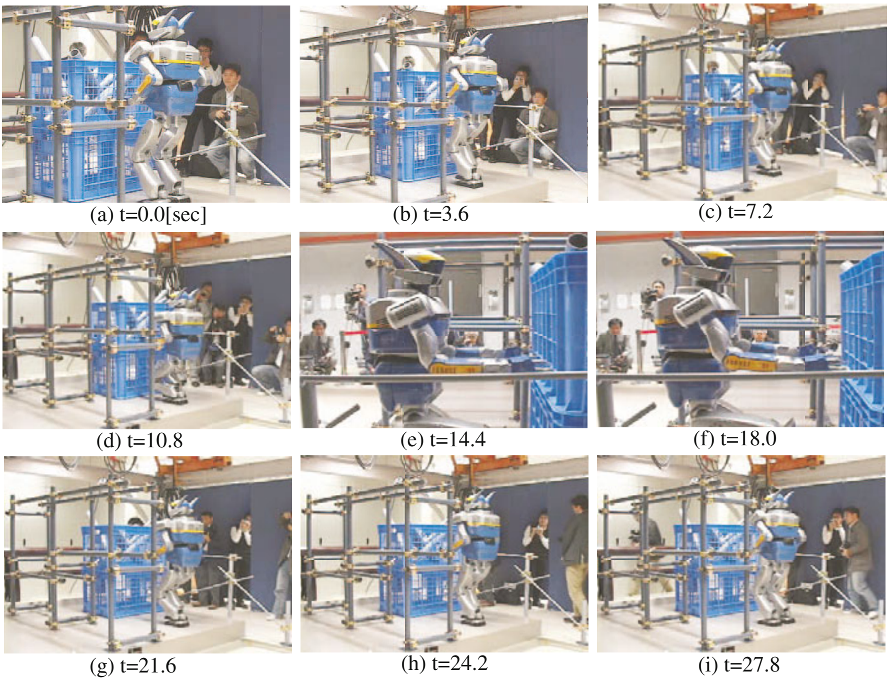

(a) t=0.0[sec]
(b) t=3.6
(c) t=7.2
(d) t=10.8
(e) t=14.4
(f) t=18.0
(g) t=21.6
(h) t=24.2
(i) t=27.8

Fig. 3. Experimental Result on Pushing Manipulation

5 Handling Task

When manipulating an object placed on the table, it sometimes become difficult for a humanoid robot to keep balance while standing on two legs. Thus, we utilize the interaction between the hand and the table to manipulate the object. The overview of the proposed method is shown in Fig.4. This method is just similar to the crawl gait of a quadruped walking robot. First, both of the hands of a humanoid robot contact the table. Then, the waist of the humanoid robot moves right making the horizontal projection of the COG be included in the convex hull of the horizontal projection of the supporting points in the right hand, right foot, and the left foot.

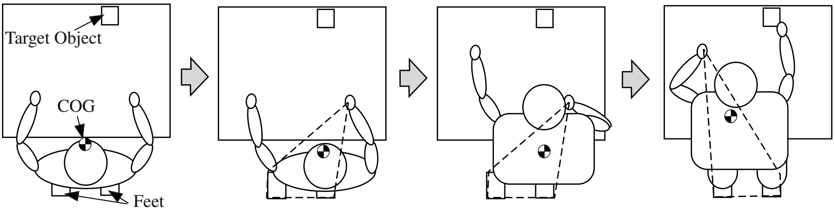


Fig. 4. Overview of the Proposed Method

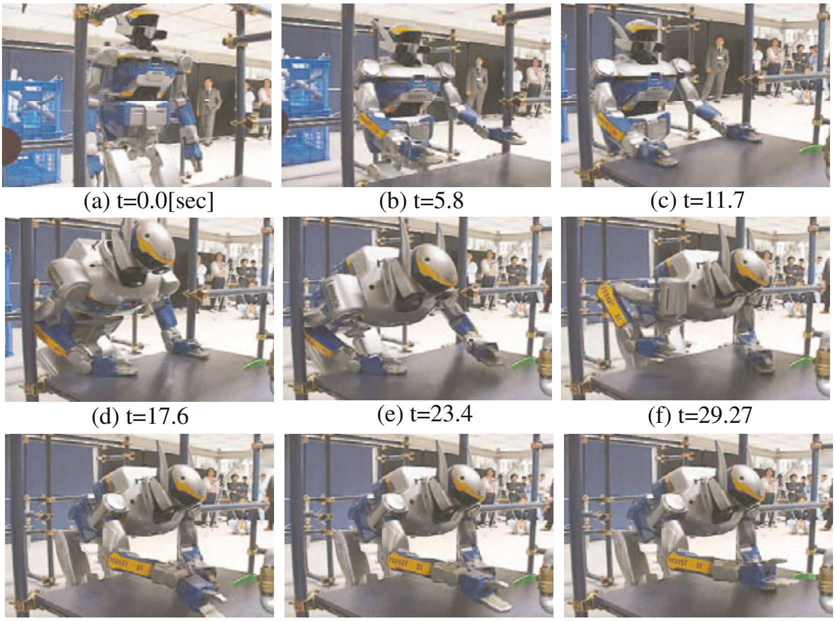

(a) t=0.0[sec]
(b) t=5.8
(c) t=11.7
(d) t=17.6
(e) t=23.4
(f) t=29.27
(g) t=35.1
(h) t=41.0
(i) t=42.0

Fig. 5. Experimental Result on Hnadling Task

Since the left hand can detach from the table, the humanoid robot moves to make the left hand be close to the object. If the humanoid robot cannot capture the object, then the humanoid robot moves to make the left hand contact the table. In such a case, the waist of the humanoid robot moves left making the horizontal projection of the COG be included in the convex hull of the horizontal projection of the supporting points in the left hand, right foot, and the left foot. Then, the humanoid robot tries to manipulate the object by the right hand. The result of experiment is shown in Fig.5. As shown in the figure, the humanoid robot succeeded in manipulating the lever while supporting the body with an arm.

We note that, while we do not consider the dynamics of the robot, we can obtain the condition for the ZMP for keeping the dynamical balance of the robot by using our previous work[7].

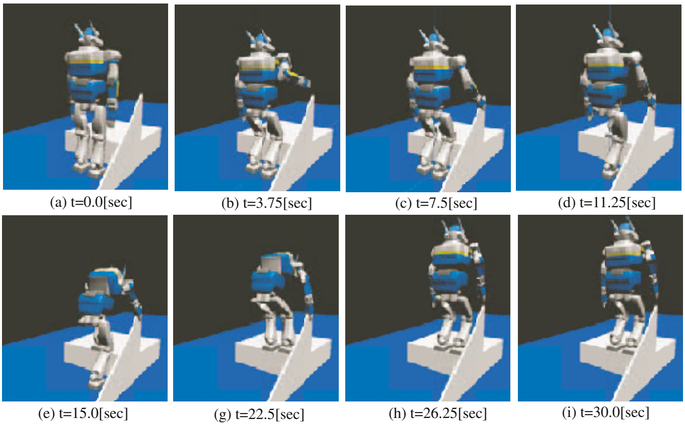

(a) t=0.0[sec] (b) t=3.75[sec] (c) t=7.5[sec] (d) t=11.25[sec]
(e) t=15.0[sec] (g) t=22.5[sec] (h) t=26.25[sec] (i) t=30.0[sec]

Fig. 6. Overview of the Planned Motion

6 Climbing a Large Step

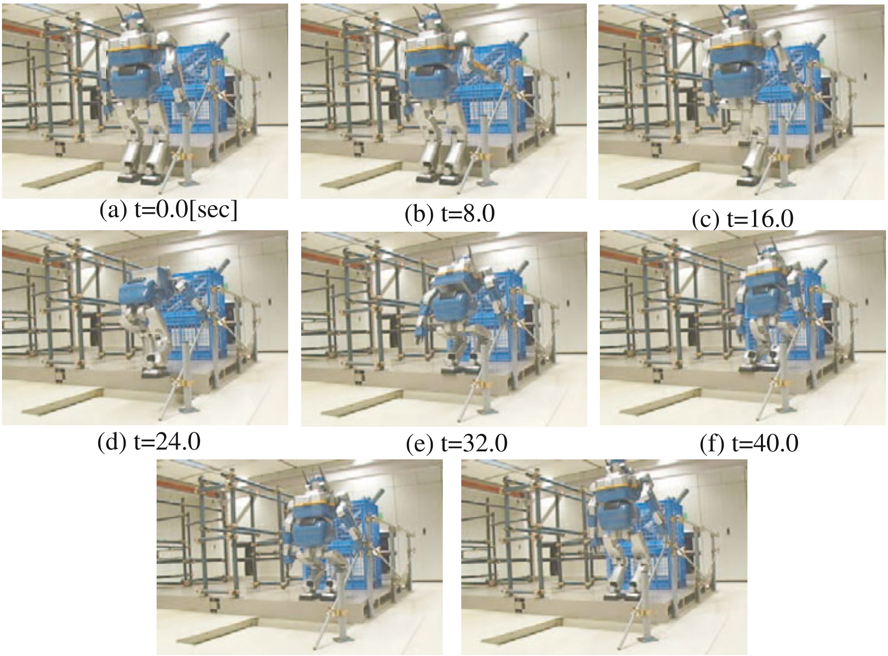
When climbing a large step, a humanoid robot can increase the stability by utilizing the interaction between the hand and the environment. We can also expect that, by grasping a handrail, it becomes easier for a robot to keep balance. The basic mechanism of ZMP for a humanoid robot grasping a handrail is shown in the appendix. We can see that, the stronger the humanoid robot grasps a handrail, the easier the humanoid robot can keep balance.

The motion of a humanoid robot climbing a large step is shown in Fig.6. First, a humanoid robot grasps the handrail. Then, a humanoid robot puts the left foot onto the step. The humanoid robot further moves to make the position of COG forward as shown in Fig.6(e). For this posture, the humanoid robot cannot keep balance unless grasping the handrail. The robot finally climbs up the large step as shown in Fig.6(i). The experimental results are shown in Fig.7.

7 Conclusion

In this paper, for the ultimate goal of developing a humanoid robot that can go anywhere an ordinary human goes, we have realized three basic motions, i.e., pushing a large object, handling task while supporting the body with an arm, and climb a large step while grasping a handrail. When performing the experiment, we assume that the geometry of the environment is known. In our future work, we will use the vision sensor when performing an experiment.

This work has been carried out under the Basic Technology Research Promotion Program "R & D of Basic Technology for Humanoid Robot Working in Real Environment" of the NEDO, FY2002-2006. We would like to express our sincere gratitude for Dr. Kenji Kaneko, Dr. Kazuhito Yokoi, Mr. Kiyoshi Fujiwara who are the humanoid robotics researchers in AIST for their helpful discussions. We would



(a) t=0.0[sec] (b) t=8.0 (c) t=16.0

(d) t=24.0 (e) t=32.0 (f) t=40.0

(g) t=48.0 (h) t=54.0

Fig. 7. Experimental Result of Climbing a Large Step

also express our sincere gratitude for Mr. Masaru Nakamura with Kawada industries. He helped us a lot when constructing the experimental environment.

References

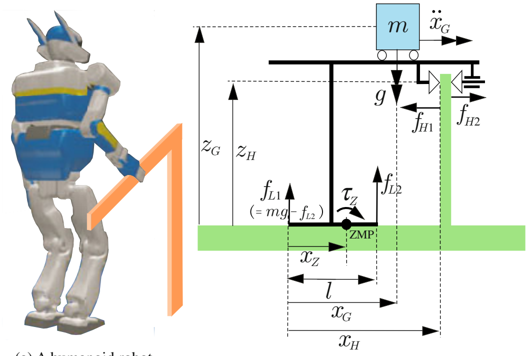
1. M. Hirose, Y. Haikawa, T. Takenaka, and K. Hirai, "*Development of Humanoid Robot ASIMO*", Workshop of IEEE/RSJ Int. Conf. on Itelligent Robots and Systems, Workshop2, 2001.
2. Y. Kuroki, M. Fujita, T. Ishida, K. Nagasaka, and J. Yamaguchi, "*A Small Biped Entertainment Robot Exploring Attractive Applications*", Proc. of the 2002 IEEE International Conference on Robotics and Automation, pp. 471-76, 2003.
3. M. Gienger, K. Löffler, and F. Pheiffer, "*Towards the Design of a Biped Jogging Robot*", Proc. of the 2001 IEEE International Conference on Robotics and Automation, pp. 4140-4145, 2001.
4. S. Kagami, K. Nishiwaki, J.J. Kuffner, Y. Kuniyoshi, M. Inaba, and H. Inoue: "*Online 3D Vision, Motion Planning and Bipedal Locomotion Control Coupling System of Humanoid Robot : H7*", Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1321-1326, 1998.
5. K. Kaneko et al.: "*Humanoid Robot HRP2*", Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1083-1090, 2004.

6. K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa: "*Pushing Manipulation by Humanoid considering Two-Kinds of ZMPs*", Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1627-1632, 2003.
7. K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa: "*ZMP Analysis for Arm/Leg Coordination*", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 73-79, 2003.
8. K. Harada, S. Kajita, F. Kanehiro, K.Fujiwara, K. Kaneko, K.Yokoi, and H. Hirukawa: "*Real-Time Planning of Humanoid Robot's Gait for Force Controlled Manipulation*", Proc. of 2004 IEEE Int. Conf. on Robotics and Automation, pp. 616-622, 2004.
9. K. Harada, H. Hirukawa, F. Kanehiro, K.Fujiwara, K. Kaneko, S. Kajita, and M. Nakamura: "*Dynamical Balance of a Humanoid Robot Grasping an Environment*", Submitted to 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2004.
10. K. Yokoyama, H. Handa, T. Isozumi, Y. Fukase, K. Kaneko, F. Kanehiro, Y. Kawai, F. Tomita, and H. Hirukawa: "*Cooperative Works by a Human and a Humanoid Robot*", Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 2985-2991, 2003.
11. K. Inoue, H. Yoshida, T. Arai, and Y. Mae: "*Mobile Manipulation of Humanoids – Real-Time Control Based on Manipulability and Stability–*", Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 2217-2222, 2000.
12. Takenaka: "*Posture Control for a Legged Mobile Robot*", Japanese Patent Application, H10-230485, 1998.
13. Y. Hwang, A. Konno, and M. Uchiyama: "*Whole Body Cooperative Tasks and Static Stability Evaluations for a Humanoid Robot*", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1901-1906, 2003.
14. F. Kanehiro et al.: "*Virtual humanoid robot platform to develop controllers of real humanoid robots without porting*", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2001.
15. H. Hirukawa et al.: "*OpenHRP: Open Architecture Humanoid Robot Platform*", Proc. of Int. Symp. on Robotics Research, 2001.
16. S.Kajita et al.: "*Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point*", Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1620-1626, 2003.
17. H. Inoue et al.: "*HRP: Humanoid Robotics Project of MITI*", Proc. of the First IEEE-RAS Int. Conf. on Humanoid Robots, 2000.

A ZMP considering the Grasping Force

As shown in Fig.8, let us consider the situation where a hand of a humanoid robot grasps an handrail by modeling the robot by the cart-table model[16]. Here, for simplicity, we model the hand of a humanoid robot by a parallel gripper with a one-dof translational joint as shown in the figure.

Let f_{H1} (≤ 0) and f_{H2} (≥ 0) be the reaction forces applied by the parallel gripper in the left-hand side and the right-hand side, respectively, of the handrail. Let f_{L1} (≥ 0) and f_{L2} (≥ 0) be the reaction forces between the table and the floor at the left-edge and the right-edge, respectively, of the contact segment, where $f_{L1} = mg - f_{L2}$ is satisfied. We neglect the effect of friction at each contact point to explain the principle as simple as possible. Also, we consider the case where the acceleration of the cart is large making $f_{H1} = 0$. As a source of ground reaction



(a) A humanoid robot holding a hand-rail

(b) The cart-table model

Fig. 8. Explanation of the proposed method

moment, we consider all forces acting on the cart-table system. Since the ground reaction moment τ_Z around the ZMP is zero, it can be formulated as

$$0 = -f_{L1}x_Z + f_{L2}(l - x_Z) - f_{H2}z_H. \tag{3}$$

Solving eq.(3) with respect to the position of ZMP, the following equation can be obtained:

$$x_Z = (f_{L2}l - f_{H2}z_H)/(mg). \tag{4}$$

Now, let us focus on the region of ZMP for keeping the dynamical balance of the cart-table system. When both $f_{L1} > 0$ and $f_{L2} > 0$ are satisfied, the cart-table model keeps the dynamical balance. It is equivalent to $0 < f_{L2} < mg$. Substituting this relationship into eq.(4), the set of ZMP keeping the dynamical balance can be obtained as:

$$\mathcal{X}_Z = \left\{ x_z \;\middle|\; -f_{H2}z_H/(mg) < x_z < l - f_{H2}z_H/(mg), f_{H2} \geq 0 \right\}, \tag{5}$$

where we note that, since we only consider the case of high acceleration in this section, we focus on the lower limit of the set of ZMP. Thus the inequality $x_z < l - f_{H2}z_H/(mg)$ in the above equation does not make sense. This set of ZMP is a function of the grasping force of the gripper. If the grasping force is zero ($f_{H2} = 0$), the set of ZMP is bounded by $x_Z > 0$ which means that the ZMP exists inside of the contact area between the table and the floor. On the other hand, when the hand is grasping a handrail, the set of ZMP is bounded by $x_Z > -f_{H2}z_H/(mg)$ which means that the cart-table system can keep the dynamical balance even if the ZMP exists out-side of the contact area. It also means that, if the hand grasps an handrail with stronger grasping force, the robot can keep the dynamical balance more easily.

Design and Modeling of a Selective-Actuation XYZ Flexure Parallel Mechanism

Huy-Hoang Pham[1] and I-Ming Chen[2]

School of Mechanical & Production Engineering, Nanyang Technological University
50 Nanyang Avenue, Singapore 639798
[1] Email: pbz0254391@ntu.edu.sg
[2] Email: michen@ntu.edu.sg

Abstract. This paper presents the design of a selective-actuation flexure parallel mechanism that can provide three independent translational motions. The mechanism can be used as an ultra precision positioning system. The modeling of the flexure parallel mechanism is then established based on a pseudo-rigid body model and the compensation of the flexure deformation. The factor of deformation allows us to formulate accurately the position control of the flexure mechanism. The dimension and free shape of the mechanism are determined based on the criteria of isotropic resolution transmission scale. The experiment shows the advantage of the proposed model versus the currently used model. The experiment also proves the validity of the resolution evaluation and the optimization.

1. Introduction

Flexure parallel mechanism (FPM) possesses no-error accumulation, high rigidity, high natural frequencies, no-backlash, no-friction, vacuum compatibility and no-need of lubrication. Therefore, FPM provides high accurate motions with small range and can be used as an ultra-precision manipulation system in fields such as optics, precision machine tools, and micro component fabrication.

Recently, several FPM-based precision positioning stages are developed (Chang et al. [1], Ryu et al. [2, 3], Koseki et al. [4], Yi et al. [5]). Most of the flexure mechanisms are designed based on not the precision criterion but the criteria of large range [1-4] and compliance/stiffness [5]. Only when the control of mechanism is implemented, the precision of the flexure mechanism has been taken into consideration. Some measures are proposed to manage the error distribution [1] or to predict the error and modify the control [2, 3]. Those post-design measures are not sufficient for obtaining the system with high accuracy. There exists a need of the pre-design measures, which guarantee right at the design stage that the mechanism can provide the motion with desired resolution. Another problem is how to model the flexure mechanism. The recent studies [1-4] simplify the modeling of a flexure mechanism by using pseudo-rigid body (PRB) model. This method does not lead to the accurate modeling because the deformation of the flexure members is not studied when establishing the model. Recently, to obtain a more precise PRB model, Yi et al. proposes the base-fixed flexure hinge as a two-

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 197–206, 2006.

link chain that consists of a revolute joint with torsion spring and a prismatic joint [5]. However, the elongation and the bending of the flexure hinge are actually tied together. Therefore, the replacement of a flexure hinge by a revolute joint and a prismatic joint is not sufficient and makes the modeling more complicated and unsuitable for a spatial flexure mechanism, when the number of joints and links increases. One other problem needs to be considered is the selective actuation (SA) of the mechanism. Usually, the cost of a precise actuator is very high. All of the degree-of-freedom (DOF) motions of the mechanism are not always used for a positioning task. Therefore, the mechanism needs to be designed so that the actuators can be removed to obtain the desired number of DOF without affecting on the precision of the mechanism. Thanks to this, the actuators also can be used for multiple tasks. As a precision positioning system can not be often re-configured, SA can be considered as the reconfigurability in precision mechanism. SA also aids the de-coupling control. The main content of the paper includes the design of a 3-DOF translation SA FPM, the novel modeling of FPM, the definition of the resolution transmission scale following by an optimal computation of the dimension and free shape of the FPM, and verification experiments.

2. Conceptual Design of a Selective-Actuation XYZ FPM

Several multi-axis positioners are developed based on flexure parallel mechanisms [1, 2, 4, 5 - 7]. However, some disadvantages such as the coupling of motions and parasitic motions limit the accuracy of the devices. It is stated in the introduction that a SA FPM overcomes the above disadvantages.

***Definition*: Selective-actuation mechanism**

***Selective-actuation mechanism** is a mechanism whose actuators can be removed so as to inhibit corresponding DOF whereas other DOF motions are not affected. Each actuator provides independent DOF motion without parasitic motions.*

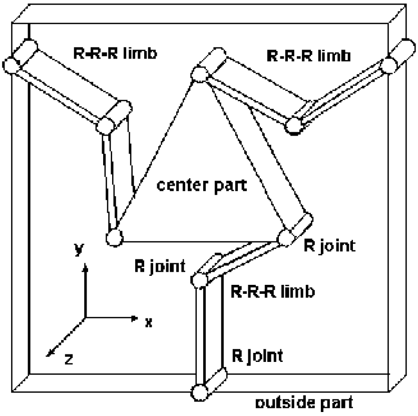


Fig. 1 Planar 3-DOF mechanism with R-R-R limbs.

The configuration of a SA XYZ FPM derives from a planar mechanism with three R-R-R limbs that possesses three DOF X, Y and θ_Z (Fig. 1). If we provide the center part with a motion along the Z-direction, the outside part obtains four DOF (X, Y, Z and θ_Z). Three planar 3-DOF mechanisms are fixed together at the outside parts to form a 3-dimension (3-D) frame so that three mechanism planes are perpendicular to each other (see Fig. 2). If we let each center part of the frame move along the corresponding direction perpendicular to the side containing the center part, each outside part individually can perform four DOF motions: X-side→(Y, Z, θ_X, X), Y-side→(Z, X, θ_Y, Y) and Z-side→(X, Y, θ_Z, Z). The outside part of the frame possesses only three translational DOF that is the intersection of three above DOF sets: $(Y, Z, \theta_X, X) \cap (Z, X, \theta_Y, Y) \cap (X, Y, \theta_Z, Z) = (X, Y, Z)$.

The developed XYZ SA FPM consists of three flexure prismatic joints built from double flexure linear slide and a 3-D frame and is actuated by three linear actuators (Fig. 2 and 3).

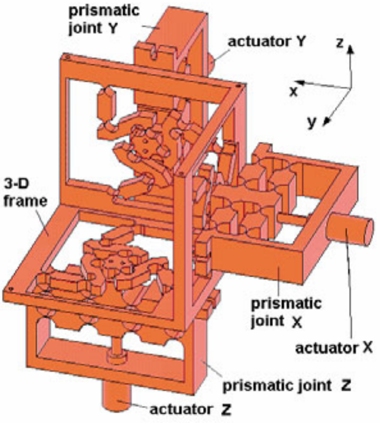


Fig. 2 3-D frame and three prismatic joints.

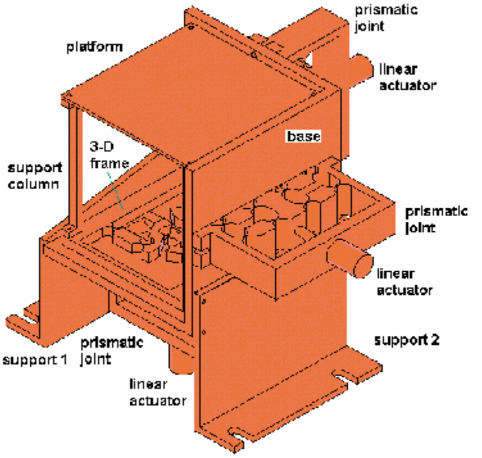


Fig. 3 SA XYZ FPM.

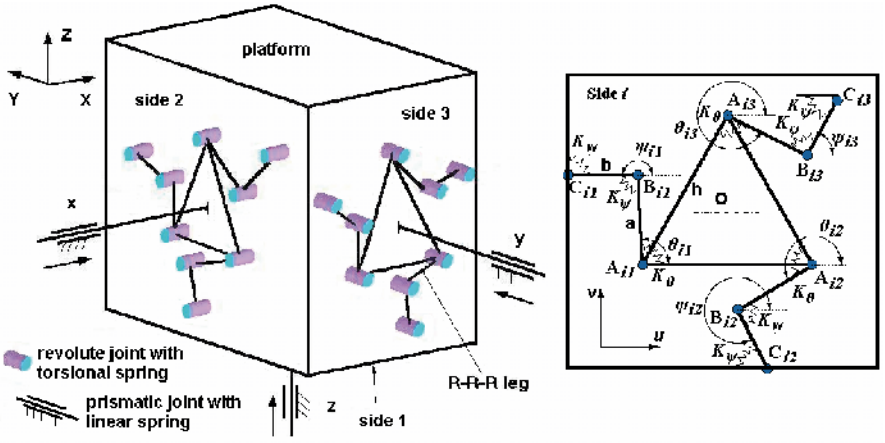
3. Modeling

The PRB model with consideration of deformation of flexure members (PRB-D model) is established by evaluating the effect of the flexure deformation and adding that effect to the original PRB model.

3.1 PRB model

The PRB model of the FPM is obtained by replacing the flexure hinges of the 3-D frame by the revolute joints and the double linear slides by the prismatic joints. Fig. 4a illustrates the general view of the PRB model having each of three perpendicular sides as a 3-DOF planar parallel mechanism (see Fig. 4b). Suppose the expected position vector of end-effector is $\boldsymbol{X} = (X, Y, Z)^T$ and the computed active joint variable vector (actuated joint variable) is $\boldsymbol{x} = (x, y, z)^T$. The PRB model gives us:

$$\boldsymbol{X} = \mathrm{H}(\boldsymbol{x}) = \boldsymbol{x} \tag{1}$$



(a) (b)

Fig. 4 Pseudo rigid-body model.
i = 1: u→x, v→y, w→z; i = 2: u→y, v→z, w→x; i = 3: u→z, v→x, w→y.

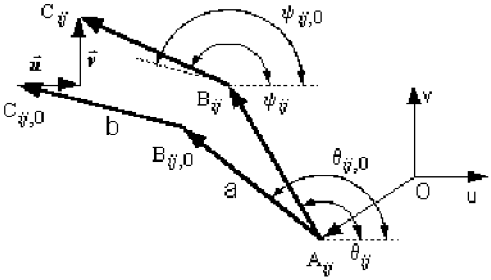


Fig. 5 Vector close loop of limb *j* on side *i*.

The passive joint variables θ_{ij} and ψ_{ij} and the passive joint displacement rates: $\dot{\theta}_{ij}$ and $\dot{\psi}_{ij}$ are obtained by solving the closed loop equations of limbs established based on Fig. 5 and their differentiation equations.

3.2 PRB-D model

The actual motion of the end-effector is assumed to be the combined effect of the deformation $\boldsymbol{E}_t$ and PRB modeling $\mathrm{H}(\boldsymbol{x}) = \boldsymbol{x}$:

$$\boldsymbol{X} = \boldsymbol{x} + \boldsymbol{E}_t \tag{2}$$

Assuming that the flexure mechanism complies with the linear deformation and superposition principles, the combined effect of deformation $\boldsymbol{E_t}$ is the accumulated error of all flexure members, so

$$\boldsymbol{E_t} = \sum_{i=1}^{n} \boldsymbol{E_i} \tag{3}$$

where n is the number of flexure members existing in the mechanism and $\boldsymbol{E_i}$ is the individual deformation of flexure member i and can be determined as

$$\boldsymbol{E_i} = \boldsymbol{G_i \varepsilon_i} \tag{4}$$

where vector ε_i is the deflection of flexure member i and matrix $\boldsymbol{G_i}$ expressing the deformation of flexure hinge i contributed to the overall effect on the end-effector motion can be achieved based on the configuration of limb containing that flexure hinge (see Fig. 6).

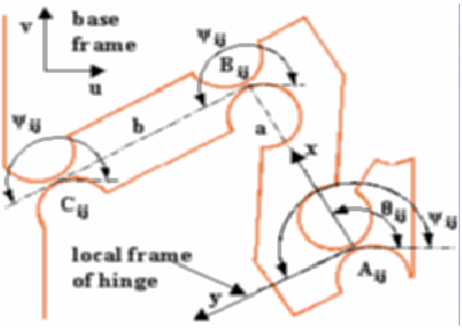


Fig. 6 Configuration of one limb.

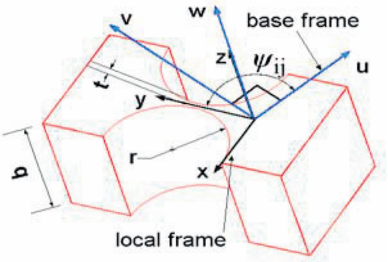


Fig. 7 Force transformation.

The deflection of flexure member i is caused by reaction forces or moments $\boldsymbol{F_i}$ applied to that member and the local stiffness matrix of that flexure member $\boldsymbol{K_i}$

$$\varepsilon_i = \boldsymbol{K_i^{-1} G_{fi} F_i} \tag{5}$$

where:

- Matrix $\boldsymbol{G_{fi}}$ transforming the reaction force from the base frame to the flexure-hinge frame i (local frame) is determined using Fig. 7 and relied only on the side.
- Reaction forces $\boldsymbol{F_i}$ are determined in the force analysis of the flexure mechanism based on the PRB model [10, 11].
- Local stiffness matrix $\boldsymbol{K_i}$ of a flexure hinge i is determined as

$$\boldsymbol{K_i} = \begin{bmatrix} K_{x-Fx} & 0 & 0 & 0 & 0 & 0 \\ 0 & K_{y-Fy} & 0 & 0 & 0 & K_{\alpha z-Fy} \\ 0 & 0 & K_{z-Fz} & 0 & K_{\alpha y-Fz} & 0 \\ 0 & 0 & 0 & K_{\alpha x-Mx} & 0 & 0 \\ 0 & 0 & K_{z-My} & 0 & K_{\alpha y-My} & 0 \\ 0 & K_{y-Mz} & 0 & 0 & 0 & K_{\alpha z-Mz} \end{bmatrix} \tag{6}$$

where the terms of local stiffness matrix can be cited from [8].

Consequently, the PRB-D model of the flexure mechanism is:

$$\boldsymbol{X} = \boldsymbol{x} + \sum_{i=1}^{n} \boldsymbol{G_i}\, \boldsymbol{K_i^{-1}}\, \boldsymbol{G_{fi}} \boldsymbol{F_i} \tag{7}$$

4. Resolution Transmission Scale and Optimal Design

4.1 Resolution Transmission Scale

Consider a non-redundant FPM with n degree-of-freedoms and denote $\boldsymbol{dq}$ = (dq_1, dq_2, …, dq_n) a vector of the infinitesimal displacements of n actuators and $\boldsymbol{dX}$ = (dX_1, dX_2, …, dX_n) a vector of the infinitesimal motions (displacements and/or orientations) of the end-effector. Let $\boldsymbol{J}$ (n x n matrix) be the coefficient matrix of resolution transmission from actuators to the end-effector (Jacobian matrix). The amplification effect of infinitesimal actuator displacements on the infinitesimal motion of the end-effector can be described by

$$\boldsymbol{dX} = \boldsymbol{J}\, \boldsymbol{dq} \tag{8}$$

To examine the resolution of the FPM, we observe the available motions of the end-effector when one of the actuators carries out the smallest controllable motion (equal to the resolution δ) of the actuator:

$$\boldsymbol{dq}^2 = \boldsymbol{dq}^T \boldsymbol{dq} = dq_1^2 + dq_2^2 + ... + dq_n^2 = \delta^2 \tag{9}$$

Assuming that the FPM operates in free singular workspace, Eq. (8) and Eq. (9) lead to:

$$\boldsymbol{dX^T} \left(\boldsymbol{JJ^T}\right)^{\boldsymbol{-1}} \boldsymbol{dX} = dX_1^2/\sigma_1^2 + dX_2^2/\sigma_2^2 + \cdots + dX_n^2/\sigma_n^2 = \delta^2 \tag{10}$$

where σ_1, σ_2, …, σ_n are the singular values of Jacobian $\boldsymbol{J}$ and

$$\det(\boldsymbol{JJ^T}) = (\sigma_1\ \sigma_2\ \dots\ \sigma_n)^2 \tag{11}$$

Eq. (10) proves that the smallest controllable motions (resolution) which the end-effector can perform along the principal axes are $\sigma_1\delta$, $\sigma_2\delta$,…, $\sigma_n\delta$ and the singular values can be considered as the resolution scales. The resolution of the end-effector is proportional to the resolution scale and varies following the motion direction. Therefore, we use a concept called "resolution transmission scale" to represent the average resolution transmission at one particular point.

Definition: ***Resolution transmission scale***

Resolution transmission (RT) scale is the geometric mean of the resolution scale along all principal directions σ_i.

$$R = \sqrt[n]{\sigma_1 \sigma_2 ... \sigma_n} \tag{12}$$

Eq. (11) and Eq. (12) show that the resolution transmission scale is computed as follows

$$R = \sqrt[2n]{\det(\boldsymbol{JJ}^T)} \tag{13}$$

Since Jacobian $\boldsymbol{J}$ is configuration dependent, the RT scale is a local performance measure that will be valid at a certain pose only. The global RT scale R_{global} computed by integrating the local RT scale over the workspace is used to evaluate the resolution transmission:

$$R_{global} = \int_W R dw \Big/ \int_W dw \tag{14}$$

where W is the workspace and dw is an infinitesimal area in which RT scale is considered as a constant.

The resolution of an ultra precision manipulation system must be uniform over the entire workspace. Therefore, to have a full assessment of resolution, we utilize an additional indicator - the uniformity U:

$$U = R_{min} / R_{max} \tag{15}$$

where R_{min} and R_{max} are the minimum and the maximum of RT scale over the entire workspace W.

4.2 Optimal Design

The RT scale needs to be minimized to obtain a highest resolution. However, Jacobian $\boldsymbol{J}$ is a coefficient matrix of the force transmission from actuators to the end-effector. The decrease of RT scale will lead to the increase of the actuation forces. To satisfy the precise motion and small actuation load, the global RT scale needs to be as close to one as possible. To descend the resolution variation, the uniformity must be kept as close to unity as possible. The optimal design of a FPM is formulated as:

Minimize: $|R_{global} - 1|$ (16)

Subject to: $|U\text{-}1| \leq \delta_U$ (17)

where δ_U is the allowable difference between U and 1.

The optimization variables: The dimension of the PRB and PRB-D models and the initial posture/free shape of FPM.

Searching method: Conjugate direction method is used to reduce computation resource and to obtain the searching direction faster.

The result of the optimization is:

Inputs: Aluminium alloy (yield stress $S_y = 500$ MPa and Young's modulus E = 71 GPa), dimension: a = 35 mm, h = 60 mm, and spring stiffness of $K_\theta = K_\psi = 1630$ N.mm/rad (9 N.mm/°), $K_S = 105$ N/mm.

Out puts: Dimension: $k_{ab} = b/a = 1.14$; Free shape/Initial posture: $\theta_{ij0} = 80°$, $\psi_{ij0} = 165°$ and optimal indicators: $R_{global} = 6.1$ and $U = 0.61$.

5. Experiments

An experiment is set up to verify the SA characteristic and the PRB-D model (see Fig. 8). The motion of end-effector is measured by a probe with a resolution of 0.1 μm, stiffness of 0.04N/mm and measurement range of -10 mm - +10 mm. When the actuator of X-direction is working, the unexpected motion dZ of end-effector along Z-axis is measured. The unexpected motion dZ is smaller than 7% (Fig. 9). To verify the advantage of the PRB-D model versus the PRB model, the solutions of the kinematics based on the PRB-D and PRB models are applied to control the FPM. The PRB model's error increases very early even when the platform displacement is under 100 μm whereas the error of PRB-D model is negligible. The error produced by the PRB-D model is only 1/3 of that by the PRB model (Fig. 10). It is possible to study the tendency of the error of PRB-D model in Fig. 10 and provide compensation for the control of the system.

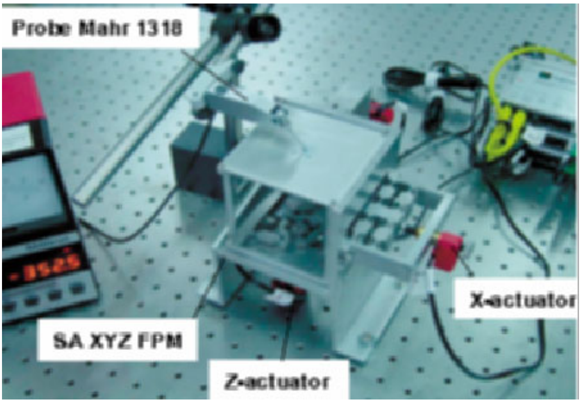
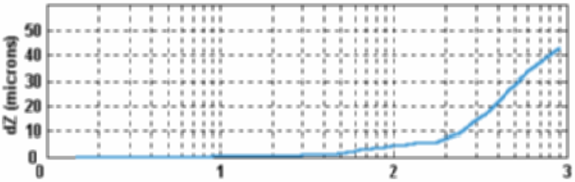


Fig. 8 SA XYZ FPM and experiment set up.



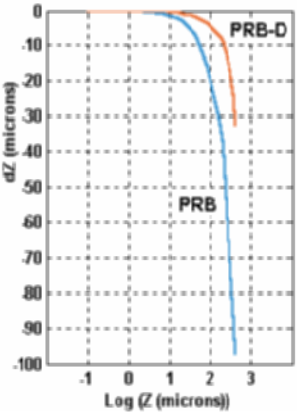


Fig. 9 Unexpected motion dZ versus the actuated motion X.

Fig. 10 Comparison between PRB and PRB-D models.

A feed back system is established for the control of the developed FPM (see Fig. 11). The motion of the end-effector of FPM is measured by the optical sensors Keyence LC-2440 (Keyence) with resolution of 0.01 μm and fed back to the controlling program. The FPM provides a series of small motions. We reduce the step size of the motions gradually to determine the smallest controllable motion of the FPM. The responses of end-effector position to the position commands with step sizes of 300, 20, 0.2 and 0.1 μm are illustrated in Fig. 12. The experiment proves that the resolution of developed FPM armed the feed back system is 0.2 μm as predicted by the design.

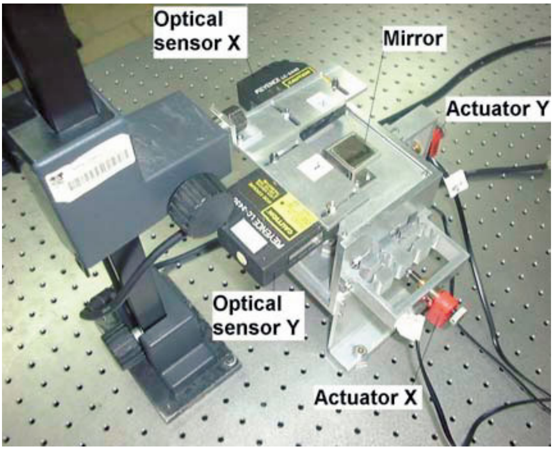


Fig. 11 FPM with close-loop control system.

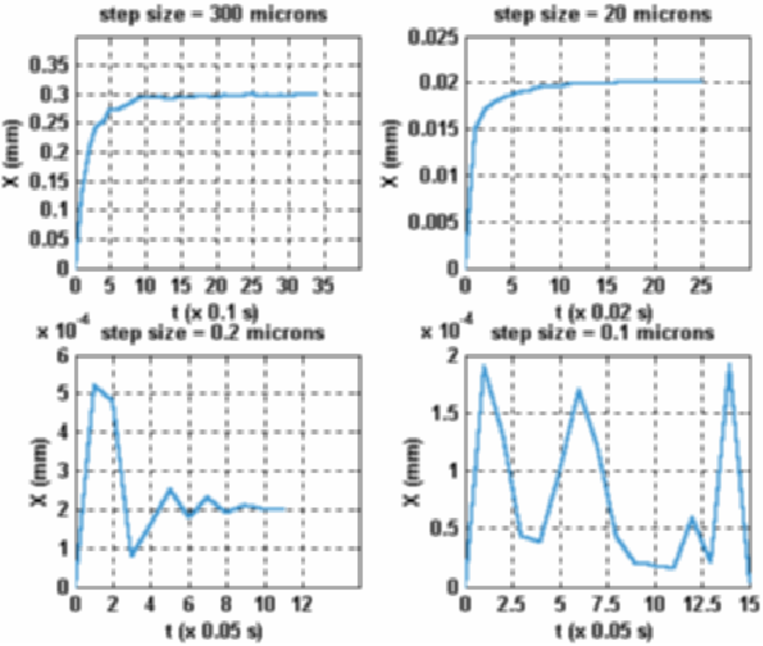


Fig. 12 Resolution of designed FPM.

6. Conclusion

A FPM with three translation de-coupled DOF motions is designed. The SA configuration allows the designed mechanism to be used for multi positioning task with number of DOF ranged from one to three. Moreover, the expensive precision actuators can be removed and reused for another tasks. The SA configuration allows the user easily and rapidly adjusts the mechanism to obtain the desired precise position. The design of a micro-motion mechanism is needed to consider the factors of resolution. Therefore, the RT scale is defined to evaluate the resolution of the mechanism. The defined scale is utilized as the criteria of the optimal design of the dimension and free shape of the FPM. To obtain the precise analytical model for formulating the optimal design and for position control, the PRB model considering the deformation of flexure members is specified. The experiment confirms the validity of the proposed model. Also, the experiment proves that we can obtain a desired resolution by the optimization.

References

[1] Chang, S., C. Tseng, H. Chien. *An ultra-precision $XY\theta_Z$ piezo-micropositioner - Part I: Design and analysis*. IEEE Trans. on Ultrasonics, Ferrorelectrics and Frequency Control, 1999. 46(4): pp. 897-905.

[2] Ryu, J., DG. Gweon, K. Moon. *Optimal design of a flexure hinge based $XY\theta$ wafer stage*. Precision Engineering, 1997. 2(1): pp. 18-28.

[3] Ryu, J., S. Lee, D. Gweon, K. Moon. *Inverse kinematic modeling of a coupled flexure hinge mechanism*. Mechatronics, 1999. 9: pp. 658-674.

[4] Koseki, Y. et al.. *Kinematic analysis of translational 3-dof micro parallel mechanism using matrix method*. Proc of IEEE/RSJ Int. Conf. on Intelligent robots and systems, Takamatsu, Japan, 2000. pp. 786-792.

[5] Yi, B.J. et al.. *Design and experiment of a 3DOF parallel micro-mechanism utilizing flexure hinges*. IEEE Int. Conf. on Robotics and Automation, Washington D.C., 2002. pp. 604-612.

[6] Davis, A. Philips, Melles Griot Limited. *Positioning mechanism*. United States Patent, App. no. 702353, 2000.

[7] Bacher, J.P., S. Bottineli, O. Ramambason, R. Clavel. *Project: Nanofactory: Ultra high precison electrodischarge machine (EDM) of small size*. http://www.epfl.ch/isr/hpr, High Precision Robotics Group, ISR-DMT-EPFL.

[8] Paros, J.M., L. Weisbord. *How to design flexure hinges*. Machine Design, 1965. 37: pp. 151-156.

[9] Griffel, D.H.. Linear algebra and its applications, Vol. 2: More advanced. Chichester: Ellis Horwood, pp. 144 - 150, 1989.

[10] Lobontiu, N.. *Compliant mechanisms: Design of flexure hinges*. CRC Press, pp.145 - 198, January 2003.

[11] Pham, H-H., I-M Chen. *Kinematics, workspace and static analysis of two degree-of-freedom flexure parallel mechanism*. ICARCV2002 Int. Conf. on Control, Automation, Robotics and Vision, Singapore, 2002. pp. 968 – 973.

On Contact Transition for Nonholonomic Mobile Manipulators

V. Padois[1], J-Y. Fourquet[1], P. Chiron[1], and M. Renaud[2]

[1] LGP-ENIT,Tarbes, France
{vpadois,fourquet,pchiron}@enit.fr
[2] LAAS-CNRS,Toulouse, France
renaud@laas.fr

Abstract This work is devoted to planning and execution of complex missions in Robotics. Robotics has evolved from an industrial, repetitive framework to application domains with much more variability of tasks, with increasing complexity in uncertain environment. This is clearly the case for Service Robotics e.g. but even industrial robots have now to work in environment not totally calibrated for the task they have to perform. The result is that the classical decomposition in static ordered local tasks cannot apply in presence of such a variability. It has a poor dynamic performance and cannot cope with uncertainties. Our work is organized around a complex mission: *"Go to the blackboard and write"* for mobile manipulators that have capabilities of locomotion and manipulation. It is a simple and intuitive example of a complex mission that relies on different sensors, exhibits different operating modes and needs to switch between different feedbacks and set-points. Our approach is based on Hybrid Dynamical Systems. It is focused on dynamical sequencing of control laws that ensures good transients, robustness and allows to update the mission at every transition from one mode to another. Simulation have been realized with *matlab Simulink* and *Stateflow* toolboxes and experimental validation is developed within the $G^{en}om$ controller on the $h_2$bis nonholonomic mobile manipulator.

1 Introduction

For many years, research community in robotics has proposed numerous solutions allowing manipulators and, more recently, **mobile manipulators** to solve specific tasks. In the literature, contributions concerning the motion in free space of the end-effector, whose location is denoted $\boldsymbol{\xi}$, are clearly separated from the ones dealing with constrained motion for which contact force $\mathbf{f}$ is imposed together with $\boldsymbol{\xi}$. In the first ones, it is assumed that no contact force can occur whereas the second category considers that the contact is realized from the beginning. Regarding manipulators, many modeling issues – at dynamic and kinematic level – and related control schemes can be found in reference textbooks [15,14,9]. Methods and models have also been proposed to deal with the control of mobile manipulators. Modeling issues can be found in [2,4] and control issues are exposed in [16,18].

Thus, the approach generally adopted when considering a complex problem is to decompose it in a sequence of elementary subproblems, and then to solve independently each subproblem. This approach has the merit to permit the study of each

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 207–216, 2006.

step of the sequence in a simple way since it disregards the other steps. It is particularly suited when the environment can be adapted to the task and when this task is repetitive enough to deserve *ad hoc* tuning. These features are usual in an industrial framework that requires a low versatility of the robotic system.

Now, new directions, both in service and manufacturing robotics, lead to embed these techniques in missions where uncertainty and variability are much more important. Wheeled locomotion, for instance, naturally produces positioning uncertainty. Environment is less calibrated and tasks are less repetitive. At the same time, it seems pertinent to build the solution from the library of classical methods by chaining them dynamically. So, it is necessary to adapt and combine existing strategies for local tasks in order to realize complex missions with good transients.

Even if the objectives of our work are more general, the work presented here is sufficiently rich to point out problems and features of a more generic complex mission in robotics. It is organized around one kind of mission:

From an imposed end-effector location, a free space motion is followed by a trajectory tracking over a surface along which a normal force is imposed,

and around one kind of system: *A* ***nonholonomic mobile manipulator****, kinematically redundant, built from a serial arm mounted on a wheeled mobile platform. This system is equipped with a force sensor that measures contact forces at the end-effector.*

We made the choice of a description based on the Hybrid Dynamical Systems formalism where switchings among a finite number of modes occur when particular events are detected. Modes are linked to different set-points and control laws. The emphasis is made on the dynamical nature of the sequence of elementary modes and more particularly on the contact transition problem (*i. e.* transition from free space to constrained effector motion). This issue has been emphasized very early for manipulators [12] but only a little number of authors has actually studied it (*e. g.* [10,7,17]). These works mainly deal with stability issues and impact modeling. Recently, Yu and Pagilla [20] have presented a method that allows to cope with geometric uncertainties of the contact surface location. Doh *et al.* [5] have also proposed a specific control law that aims at avoiding large bounces during hard contact transitions. With regard to mobile manipulators, the contact transition problem is an emerging field of research that has not been extensively studied yet. Nonetheless, one can refer to the work of Nagatani *et al.* [11] focused on static chaining of primitives ("*Open the door*") but also to the work of Kang *et al.* [8] focused on the utilization of inertial effect when controlling holonomic mobile manipulators.

2 Structuring and planning the mission

In our approach a mission is described by k modes or local tasks, and by a switching pattern that describes the sequencing of the local tasks. The switching pattern can be represented using a finite state machine. States corresponds to modes and transitions are activated by events: time elapsed or detection of a sensor threshold value. Using this formal description and the mission parameters, a first off-line planning step leads to "*a perfect plan*" that describes the ideal temporal evolution of the variables to control (*i. e.* the set-points or trajectories to track) and contains the calculated threshold values and time windows needed to define the transition conditions between modes. A state machine is then instantiated with these values. During the mission, this instantiated state machine creates indicators used to switch between the different phases of the plan and among a set of control schemes. At each transition, the "*perfect plan*" is updated thanks to the different signals and values of the variables describing the mission (*e. g.* the relative location of the end-effector in the environment frame is updated at contact time).

This update allows to cope with uncertainties due to poor parameter estimations of both the robot (drift of the platform, sensors resolution) and the environment. Thus, transition from a task to the following one occurs asynchronously but in a predictable time (or space) interval. Any large drift in the transition date or in the value of a controlled variable corresponds to an anomaly that cannot be interpreted as the result of these uncertainties and leads to a premature stop of the mission.

3 Modeling of a mobile manipulator

We consider the case of mobile manipulators composed of a mobile platform with two independent driving wheels and a serial manipulator with n_b joints. Such a system is depicted on figure 1 in a planar (right) or in a 3−dimensional version (left). This latter is the one used for the experiments. It has velocity controlled joints with high ratio gear reduction (*i. e.* the torque at each joints cannot directly be controlled) and so needs to rely on kinematic control schemes. Thus, we hereafter present some kinematics modeling results regarding mobile manipulators based on Bayle *et al.* work [2,3]. Nonetheless, dynamics are taken into account in impact reduction control scheme.

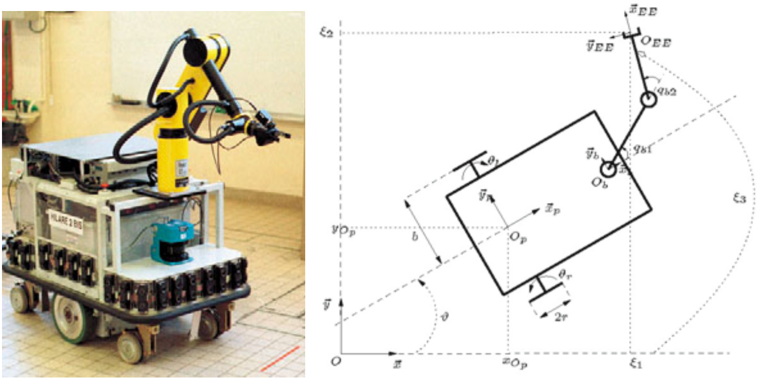


Figure 1. (*left*) H_2bis: a 3−dimensional mobile manipulator; (*right*) A planar mobile manipulator.

The configuration of such a mobile manipulator is completely defined using vector $\mathbf{q} = [\mathbf{q}_b \; \mathbf{q}_p]^T$ where $\mathbf{q}_b = [q_{b1} \ldots q_{bn_b}]^T$ and $\mathbf{q}_p = [\theta_r \; \theta_l \; x_{O_p} \; y_{O_p} \; \vartheta]^T$ respectively represent the manipulator configuration and the platform configuration. Its end-effector location (*i. e.* location of $\mathcal{R}_{EE} = (O_{EE}, \overrightarrow{x}_{EE}, \overrightarrow{y}_{EE}, \overrightarrow{z}_{EE})$ in $\mathcal{R} = (O, \overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z}))$ can be described using a minimal set of parameters $\boldsymbol{\xi} = [\xi_1 \ldots \xi_m]^T$. $\boldsymbol{\xi}$ is expressed as a non linear function of $\mathbf{q}$. Differentiating it, the relation between $\dot{\boldsymbol{\xi}}$ and $\dot{\mathbf{q}}$ is given by:

$$\dot{\boldsymbol{\xi}} = J(\mathbf{q})\dot{\mathbf{q}} \text{ with } J(\mathbf{q}) \text{ a } m \times n \text{ matrix and } n = n_b + 5. \tag{1}$$

In addition, the rolling without slipping constraint acting on the platform's wheels is nonholonomic and so, components of $\dot{\mathbf{q}}$ cannot vary independently. Then, one can define a vector $\mathbf{u} = [\mathbf{u}_b \; \mathbf{u}_p]$ of independent parameters (*i. e.* taking the nonholonomic constraints into account) such as: $\dot{\mathbf{q}} = T(\mathbf{q})\mathbf{u}$.
Defining $\overline{J}(\mathbf{q}) = J(\mathbf{q})T(\mathbf{q})$ equation (1) becomes: $\dot{\boldsymbol{\xi}} = \overline{J}(\mathbf{q})\mathbf{u}$. This equation completely describes the mobile manipulator kinematics. This model is called *instantaneous posture kinematic model* in [1]. A natural choice for $\mathbf{u} \in \mathcal{R}^{\bar{n}}$ ($\bar{n} = n_b + 2$) is $\mathbf{u}_b = \mathbf{q}_b$ and $\mathbf{u}_p$ can be chosen as $[\dot{\theta}_r \; \dot{\theta}_l]^T$ or $[v \; \omega]^T$ where v and ω are respectively the linear and angular velocities of the platform.

When $m < \bar{n}$, the mobile manipulator is said to be kinematically redundant. This property provides the capability to choose a particular kinematic control vector $\mathbf{u}$ among those giving the prescribed end-effector velocity $\dot{\boldsymbol{\xi}}$ by using the relation:

$$\mathbf{u} = \overline{J}(\mathbf{q})^{\sharp}\dot{\boldsymbol{\xi}} + (I - \overline{J}(\mathbf{q})^{\sharp}\overline{J}(\mathbf{q}))\mathbf{z}, \tag{2}$$

where $\overline{J}(\mathbf{q})^{\sharp}$ is any generalized inverse of $\overline{J}(\mathbf{q})$ and $\mathbf{z}$ any $\bar{n} \times 1$ vector. Note that, among the generalized inverses, authors generally choose the pseudo-inverse [6] and that the second right-hand term of equation (2) gives rise to the so-called "internal motion" control term since it does not provide any end-effector velocity.

4 Control schemes and use of the redundancy for the "*Go to the blackboard and write*" mission

An *ad hoc* decomposition of this mission is given by the finite state machine on figure 2.

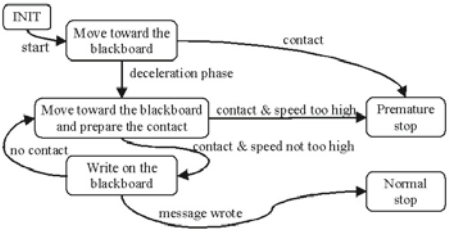


Figure 2. High-level finite state machine for the "*go to the blackboard and write mission*"

4.1 Operational kinematic control

Operational kinematic control uses a two stages control structure. The first stage is at the actuator level. Each actuator has its own velocity control loop. In our case, these controllers are digital PID regulators. The second stage is at end-effector level. Given $\dot{\boldsymbol{\xi}}^*$ and $\boldsymbol{\xi}^*$ desired end-effector speed and location to track, and a positive definite weighting matrix W_{reg}, the control vector defined by:

$$\mathbf{u} = \overline{J}(\mathbf{q})^\sharp(\dot{\boldsymbol{\xi}}^* + W_{reg}(\boldsymbol{\xi}^* - \boldsymbol{\xi})) + (I - \overline{J}(\mathbf{q})^\sharp \overline{J}(\mathbf{q}))\mathbf{z}, \tag{3}$$

ensures an asymptotic decreasing of $e = \boldsymbol{\xi}^* - \boldsymbol{\xi}$ toward 0 and $\mathbf{z}$ is chosen such as to minimize a scalar function (a *potential function*) $\mathcal{P}(\mathbf{q})$ [3].

4.2 Hybrid speed / force controller

The hybrid speed / force controller is a modified version of the well known work by Raibert and Craig [13].

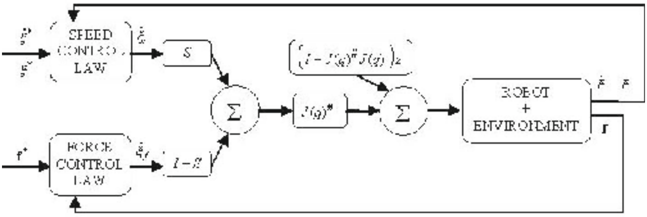


Figure 3. Hybrid speed / force scheme control.

Once the contact established, the robot's end-effector cannot independently exert a displacement and a force in the same direction. One has to choose the direction to be position or speed controlled and the direction to be force controlled. $\overrightarrow{n}_c$ being the unit vector normal to the environment surface at contact point, a smart choice for the reference frame $\mathcal{R}$ is $\overrightarrow{x} = -\overrightarrow{n}_c$. Thus, one can create a $m \times m$ diagonal selection matrix S where ones or zeros are placed on the diagonal respectively to indicate whether the component of $\boldsymbol{\xi}$ corresponding to the line in S is speed or force controlled. The control vector is then calculated as the sum of three terms:

$$\mathbf{u} = \mathbf{u}_s + \mathbf{u}_f + \mathbf{u}_r \;\text{ with} \tag{4}$$

$$\mathbf{u}_s = \overline{J}(\mathbf{q})^\sharp S \dot{\boldsymbol{\xi}}_s,\; \mathbf{u}_f = \overline{J}(\mathbf{q})^\sharp (I - S)\dot{\boldsymbol{\xi}}_f \text{ and } \mathbf{u}_r = (I - \overline{J}(\mathbf{q})^\sharp \overline{J}(\mathbf{q}))\mathbf{z}. \tag{5}$$

$\dot{\boldsymbol{\xi}}_s$ and $\dot{\boldsymbol{\xi}}_f$ are the control vectors whose simple versions are given by:

$$\dot{\boldsymbol{\xi}}_s = \dot{\boldsymbol{\xi}}^* + W_{reg_s}(\boldsymbol{\xi}^* - \boldsymbol{\xi}),\; \text{and } \dot{\boldsymbol{\xi}}_f = W_{reg_f}(\mathbf{f}^* - \mathbf{f}), \tag{6}$$

where W_{reg_s} and W_{reg_f} are two positive definite weighting matrices. A block description of this control scheme is given by figure 3.

4.3 Use of redundancy

We expose here a set of functions to optimize using the internal motion. Many other functions may be used but these ones seem to be relevant according to the considered mission.

Manipulability maximization: The manipulability notion was first introduced for manipulators (*cf.* [19] for a detailed presentation of this notion) but was also extended to mobile manipulators in [3]. The different manipulability measures are quantitative indicators representing the ease to instantly move the end-effector in any direction. Maximizing any of these indicators tends to avoid singular configuration of the system and thus to avoid high joints speed.

Impact force reduction: During the transition tasks, it is interesting to re-configure the mobile manipulator using internal motion so as to give it good inertial properties. Results concerning holonomic mobile manipulators are presented in [8]. The notion of impulse force $\hat{\mathcal{F}}$ is used in[17] and is defined as:

$$\hat{\mathcal{F}} = \lim_{\Delta t \to 0} \int_{t}^{t+\Delta t} \mathbf{f}(s) ds, \tag{7}$$

where t is the impact time. Using the dynamic model of the system (*i. e.* the model establishing the relation between physical effects of motion, actuating torques and contact forces at end-effector level), the magnitude $\hat{F}$ of $\hat{\mathcal{F}}$ is expressed as:

$$\hat{F} = \frac{-(1+e)\mathbf{v}_{impact}^{T}\mathbf{n_c}}{\mathbf{n_c^T} C(\mathbf{q})\mathbf{n_c}}. \tag{8}$$

$\mathbf{v}_{impact}^{T}$ is defined as the linear speed of the end-effector at the impact time, e is the restitution coefficient (*i. e.* a physical characteristic of the contact) and $C(\mathbf{q})$ is an inertia-related matrix.

To minimize $\hat{F}$, one can:

- plan a low impact speed but uncertainties in the plan execution may lead to higher values than the planned one,
- maximize, using internal motion, $\mathbf{n_c^T} C(\mathbf{q})\mathbf{n_c}$ which is configuration dependent. It directly acts on the inertial properties of the manipulator.

Collision avoidance: Techniques to avoid obstacles have extensively been studied in the case of mobile robots. However, the problem to solve here is more complicated since the end-effector is constrained by the manipulation task. Hence, to avoid collision of the platform with the surface of contact, the potential function to maximize is chosen as the distance between the surface and point O_p.

5 Experimental system and simulation results

Experiments are being developed on the system depicted on left figure 1. In order to easily test different control structures and schemes, we also developed a simulator using *Matlab* and *Simulink*. We hereafter detail the experimental and simulation frameworks and we then present some results obtained using the simulator.

5.1 Experimental framework

The mobile platform is actuated using two independent driving wheels (*i. e.* an HILARE type platform). The manipulator is a 6R serial arm called GT6A. The whole system, $H2bis$, is controlled using on board calculators equipped with the real time operating system *VxWorks* and running on *Motorola 68000* and *Apple Power PC* cards.

Control algorithms are implemented using $G^{en}om$*,* a generator of software control modules developed, as well as the robot, in the *RIA* team of the LAAS laboratory. At actuator level, the control sampling time is 5ms (the smaller one admitted by the control modules) whereas at operational level we chose it to be either 10 or 20ms. The $6-$axis *GIROBO* force / torque sensor bandwidth is 100Hz.

5.2 Simulation framework

Robot: The robot is modeled at kinematic and dynamic levels as a continuous system. Actuators (*dc motors*) are modeled as second order linear systems and we take into account load torques due to dynamic effects of motion or due to contact. We also take into account saturations of the actuators.

Sensors: Force sensor inaccuracy is modeled together with additive noise. Force signal is sampled with a period of 10ms to reflect the effects of its limited bandwidth.

Contact: Contact between the end-effector and the environment is locally modeled either as a {*spring // damper*} system (*i. e.* Kelvin-Voigt visco-elastic model) or as a spring system. When running simulations, we consider that the values of the parameters characterizing the contact are unknown or poorly estimated.

Controllers: Actuators' controllers are modeled as digital PID regulators whose sampling time is 5ms. The operational level controller is also implemented as a digital controller. We consider its sampling time as an off-line tunable parameter whose minimum value is 10ms.

5.3 "Go to the blackboard and write": *Simulation results*

System depicted on right figure 1 is chosen for simulation. $\boldsymbol{\xi}$ is chosen as the end-effector position and $\mathbf{u} = [\dot{q}_{b1}\ \dot{q}_{b2}\ \dot{\theta}_r\ \dot{\theta}_l]^T$. The reference frame is attached to the blackboard whose surface is defined by equation $x = 0$. Thus, $\mathbf{f}$ is defined as the vector whose components are the normal and tangential force to the blackboard. Once the contact established, the normal direction is force controlled whereas the tangential direction is speed controlled.

Impact force reduction Simulation whose parameters are described in table 1 illustrates the effects of using kinematic redundancy to reduce impact peak force.

Table1. Simulation parameters.

	real	estimated
contact point $[m]$	$\begin{bmatrix} 0 \ 0 \end{bmatrix}^T$	$\begin{bmatrix} 0.05 \ 0 \end{bmatrix}^T$
contact stiffness k_e $[N.m^{-1}]$	9000	$\hat{k}_e = k_e$
initial configuration $[rad]$ and $[m]$	$[\frac{\pi}{4} \ \frac{\pi}{4} \ 0 \ 0 - 1.75 \ - 0.3 \ \frac{\pi}{3}]^T$	
initial location $[m]$	$\begin{bmatrix} -2 \ 0.6 \end{bmatrix}^T$	
planned final location $[m]$	$\begin{bmatrix} 0 \ 0.5 \end{bmatrix}^T$	
force set-point $[N]$	$\begin{bmatrix} 10 \ 0 \end{bmatrix}^T$	
$W_{reg_s}[s^{-1}]$, $W_{reg_f}[m.N^{-1}.s^{-1}]$, S	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} \frac{10}{\hat{k}_e} & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	
Operational controller sampling time $[s]$	20×10^{-3}	

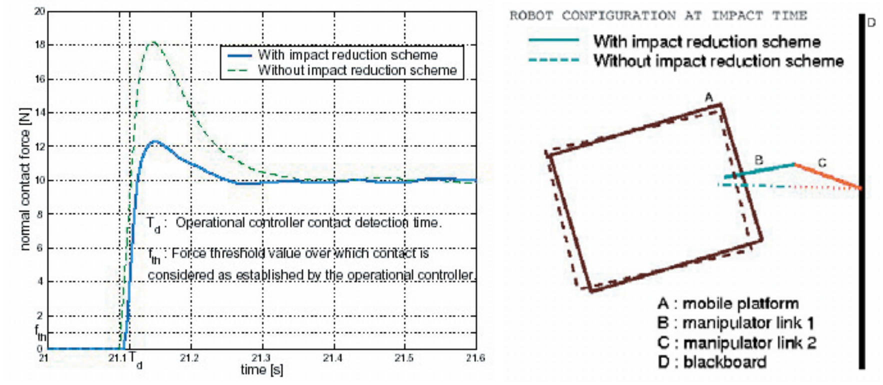


Figure 4. Impact force reduction: (*left*) Normal contact force comparison. (*right*) Robot configuration at impact time.

Figures 4 and 5 have been obtained with two simulations: the first one is such that $\mathbf{z} = 0$ in equation (3) whereas the second one uses a potential function aiming at minimizing impact forces (*cf.* equation (8)). This potential is activated during a given period of time ΔT_p around the planned impact time.

Figure 4 (*left*) shows that force overshoot is decreased from 80% to 20% of the final desired value. Figures 4 (*right*) and 5 show how internal motions generated by the potential modify the evolution of each coordinate. Inertial characteristics depend on the arm configuration and it is expected that the use of this particular potential mainly affects the manipulator generalized coordinates. Indeed, q_{b1} and q_{b2} are the more influenced by the internal motion resulting from the chosen potential. However, the other coordinates are also modified since motion of the platform must compensate for manipulator motion.

According to the current configuration, the action of a potential may cause rough variations of generalized velocities that could be inconsistent with the velocity bounds of the system. Hence, the potential is introduced here with a time-varying coefficient ensuring velocity continuity and allowable velocities and accelerations for the system.

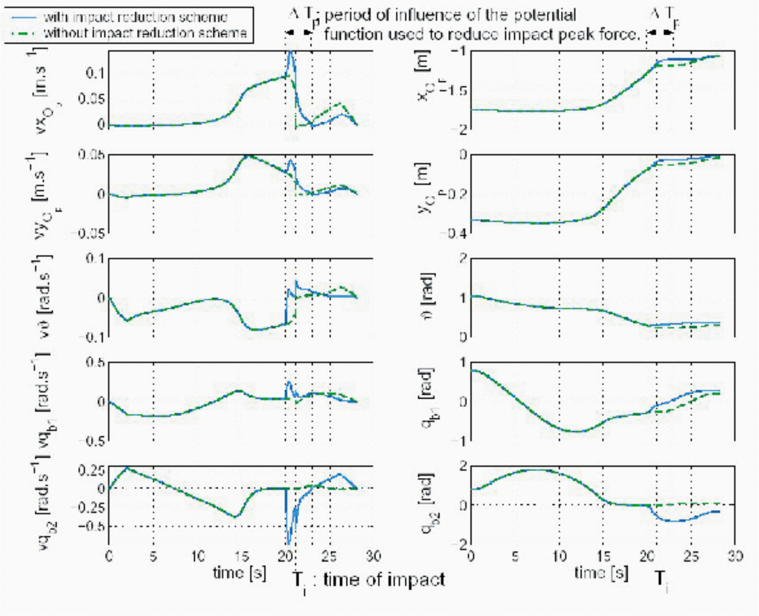


Figure 5. Impact force reduction: Effects on the position and speed of the different robot's bodies.

6 Conclusion

The aim of this study was to propose a method to control dynamic contact transition for nonholonomic mobile manipulators. The method is based on the decomposition of complex mission in elementary tasks to sequence. A particular attention is paid to transition phases. A simulator, taking into account the physical system's constraints, was developed and allows to test different control structures and schemes. The obtained results show the efficiency of the proposed method. In particular, a smart use of the kinematic redundancy of the system allows to correctly control the contact transition phase. The principle of the proposed approach was validated using simulation and is being confirmed using experiments currently developed on robot H_2bis. Further developments are also led to propose a systematic method to generalize the sequence and scaling of different potential functions in a same mission.

Acknowledgment: This work is partially supported by the French CNRS Robea program within the project *Dynamic sequencing of multi-sensor based tasks for complex motions execution in mobile robotics*.

References

1. B. Bayle. *Modélisation et commande cinématiques des manipulateurs mobiles à roues*. PhD thesis, LAAS-CNRS, Toulouse, France, 2001.

2. B. Bayle, J-Y. Fourquet, and M. Renaud. Kinematic modelling of wheeled mobile manipulators. In *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003.
3. B. Bayle, J-Y. Fourquet, and M. Renaud. Manipulability of wheeled mobile manipulation: application to motion generation. *The International Journal of Robotics Research*, vol. 22(7-8):565–581, July 2003.
4. G. Campion, G. Bastin, and B. D'Andréa Novel. Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. *IEEE Transactions on Automatic Control*, 12:47–62, February 1996.
5. N. Doh, W.K. Chung, and Y. Youm. On hard contact force control. In *IEEE/RSJ International Conference on Intelligent Robotic Systems*, pages 1528–1533, Takamatsu, Japan, November 2000.
6. K. L. Doty, C. Melchiorri, and C. Boniveto. A theory of generalized inverses applied to Robotics. *The International Journal of Robotics Research*, 12(1):1–19, February 1993.
7. J. M. Hyde and M. R. Cutkosky. Controlling Contact Transition. *IEEE Control Systems Magazine*, 14(1):25–30, February 1994.
8. S. Kang, K. Komoriya, K. Yokoi, T. Koutoku, and K. Tanie. Utilization of inertial effect in damping-based posture control of mobile manipulator. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 1277–1282, Seoul, Korea, 2001.
9. W. Khalil and E. Dombre. *Modeling, Identification and Control of Robots*. Hermes Penton Science, 2002.
10. J.K. Mills and D.M. Lokhorst. Control of robotic manipulators during general tasks execution: A discontinuous control approach. *The International Journal of Robotics Research*, vol. 12(2):146–163, April 1993.
11. K. Nagatani and S. Yuta. Door-opening behaviour of an autonomous mobile manipulator by sequence of action primitives. *Journal of Robotic Systems*, 13(11):709–721, 1996.
12. R.P. Paul. Problems and research issues associated with the hybrid control of force and displacement. In *IEEE International Conference on Robotics and Automation*, pages 1966–1971, Raleigh, North Carolina, March 1987.
13. M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, 1981.
14. L. Sciavicco and B. Siciliano. *Modelling and control of robot manipulators*. Springer, 1999.
15. M. W. Spong and M. Vidyasagar. *Robot Dynamics and Control*. J. Wiley and Sons, 1989.
16. Y. Umeda, D. Nakamura, T. Mukarami, and K. Ohnishi. Hybrid position/force control of a mobile manipulator based on cooperative task sharing. In *Proceedings of the 1999 IEEE International Symposium on Industrial Electronics*, Bled, Slovenia, 1999.
17. I.D. Walker. Impact configurations and measures for kinematically redundant and multiple armed robot systems. *IEEE Transactions on Robotics and Automation*, vol. 12(5):670–683, 1994.
18. Y. Yamamoto. *Control and coordination of locomotion and manipulation of a wheeled mobile manipulator*. PhD thesis, University of Pensylvania, 1994.
19. T. Yoshikawa. *Foundations of robotics - Analysis and Control*. The MIT Press, 1990.
20. B. Yu and P. R. Pagilla. A Switching Control Scheme for Constrained Robot Tasks. In *Proceedings of the 2001 IEEE Conference on Control Applications*, Mexico City, Mexico, September 2001.

Experimental Study of Dynamic Task / Posture Decoupling in Minimally Invasive Surgery Motions

Micaël Michelin[1], Philippe Poignet[1], Etienne Dombre[1]

[1] LIRMM – UMR 5506 CNRS / Université Montpellier II
Montpellier, France
michelin@lirmm.fr, poignet@lirmm.fr, dombre@lirmm.fr

Abstract. This paper deals with the use of an original dynamic task / posture decoupling control algorithm that allows a robot to achieve motions under the constraint of moving through a fixed point. This work takes place in the context of minimally invasive surgery where the tool is telemanipulated by the surgeon through the trocar fixed on the patient. The algorithm is based on the dynamic control in the operational space of a redundant robot: the total control torque is decoupled into a task behavior torque and a posture behavior torque. By minimizing the contact force applied to the trocar (or equivalently, by forcing to zero the distance between the instrument passing through the trocar and the current location of the trocar), we compute the posture behavior torque guaranteeing that the trocar constraint is satisfied. Implementation on a real robot has been done. Experimental results highlighting the performance of this algorithm are presented and discussed.

1 Introduction

In the ten past years, minimally invasive surgery (MIS) has become widespread in surgical operations. MIS consists in achieving operation through small penetration points in the body equipped with trocars. The surgeon uses dedicated instruments consisting in a long tube (30-40 cm) with a tool fixed at one end and a handle fixed at the other. MIS adds several difficulties in the surgical procedure. An important one is that the penetration point reduces the tool orientation capabilities and the amplitude of motion.

After training, the surgeon can overcome these difficulties in most abdominal operations. However, for microsurgery or cardiac surgery, it is necessary to make use of a robot working in a teleoperated mode. The surgeon can then focus on the tool motion rather than on the complex motion of the arm due to the constraint.

A few robotic systems have been designed to assist surgeons, such as Zeus from Computer Motion or Da Vinci from Intuitive Surgical. One of the main features of these systems is to mechanically create a fixed point that coincides with the penetration point. Zeus makes use of a passive universal joint [1]. Da Vinci and other prototypes such as FZK Artemis and UCB/UCSF RTW systems are designed as remote center devices [2]-[5]. An improvement to the passive fixed point of Zeus has been proposed in [6] to minimize the force constraint on the trocar by implementing an appropriate force-position control. Finally, let us mention a hybrid

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 217–226, 2006.

system consisting of a parallel structure outside the patient holding a serial structure located inside [7].

In this paper, we propose a novel approach based on the dynamic decoupling of the control torque of a redundant robot into a task behavior control and a posture behavior control. In fact, by minimizing the contact force or, equivalently, by forcing to zero the distance between the instrument passing through the trocar and the current location of the trocar (or the desired location, if the goal is to control it), we compute the posture behavior torque as the gradient of a cost function representing the distance. This approach allows us to control the penetration point required during MIS. It has been implemented on a real five degree-of-freedom (dof) robot, which achieves motion under the constraint of passing the instrument through the trocar.

The paper is organized as follows: section 2 recalls the dynamics formulation, the principle of task / posture decoupling, and the proposed algorithm. In section 3, we present the D2M2 robot and the experimental platform dedicated to beating heart surgery, as well as experimental results obtained when the tip of the surgical instrument tracks simple paths such as straight line and helix. These results are discussed in section 4.

2 Task / posture dynamic decoupling

2.1 Joint space and operationnal space dynamics

The joint space dynamic model for a N-dof open-chain manipulator is given by

$$\Gamma = A(q)\ddot{q} + b(q,\dot{q}) + g(q) \tag{1}$$

where $A(q)$ is the NxN joint space inertia matrix, $b(q,\dot{q})$ is a Nx1 vector combining joint space Coriolis forces and centrifugal forces, and $g(q)$ is the Nx1 joint space gravitational force vector.

The operational space behavior is given by

$$F = \Lambda(q)\ddot{X} + \mu(q,\dot{q}) + \rho(q) \tag{2}$$

where $\Lambda(q)$, $\mu(q,\dot{q})$, $\rho(q)$ represent respectively the inertia matrix, the Coriolis / centrifugal force vector, and the gravity force vector in the operational space [9].

2.2 Joint space and operationnal space correspondance

Joint space and operational space are related by the following expressions [10]

$$\Lambda(q) = (J(q)A^{-1}(q)J^{T}(q))^{-1} \tag{3}$$

$$\mu(q,\dot{q}) = \bar{J}^{T}(q)b(q) - \Lambda(q)\dot{J}(q)\dot{q} \tag{4}$$

$$\rho(q) = \bar{J}^{T}(q)g(q) \tag{5}$$

where $\bar{J}(q)$ is the dynamically consistent generalized inverse of the Jacobian matrix $J(q)$ that links joint velocities and operational velocities such that

$$\dot{X} = J(q)\dot{q} \tag{6}$$

2.3 Task and posture control torque decoupling

The control torque vector Γ applied on the joints of a redundant robot can be decomposed in two contributions [10][12]: the first one affects the end effector position (the task control torque Γ_{task}), the other one allows to change the joint configuration without causing end effector motion (the posture control torque $\Gamma_{posture}$)

$$\Gamma = \Gamma_{task} + \Gamma_{posture} \tag{7}$$

The task control torque Γ_{task} that acts in the operational space is given by

$$\Gamma_{task} = J^T F \tag{8}$$

where F is the force vector acting on the robot in the operational space.

The posture control torque $\Gamma_{posture}$ generating joint motion without causing operational motion is expressed as

$$\Gamma_{posture} = (I_N - J^T \bar{J}^T)\Gamma_{null} \tag{9}$$

where Γ_{null} is an arbitrary null space control torque vector that is chosen by the user, and $I_N - J^T \bar{J}^T$ is an appropriate projection matrix that maps Γ_{null} to the control torque. Γ_{null} can be written as

$$\Gamma_{null} = A\ddot{q}_{null} + b + g \tag{10}$$

where $\ddot{q}_{null}$ is the null space posture behavior.

By developing (7) and combining it with equations (1) through (5), it follows that (see appendix for computation details and Fig. 1 for an example of implementation)

$$\Gamma = J^T \Lambda(\ddot{X} - \dot{J}\dot{q} - J\ddot{q}_{null}) + \Gamma_{null} \tag{11}$$

2.4 Minimally invasive surgery context

MIS operations constrain the instrument manipulated by the surgeon to pass through a fixed penetration point. As explained in section I, existing medical devices such as Da Vinci or Zeus satisfy this constraint thanks to a dedicated kinematic design. We propose to use the decoupling control scheme presented above to meet this goal with a redundant kinematic architecture.

2.5 Optimization

The task / posture dynamic decoupling algorithm detailed above offers the possibility of using the null space to optimize an objective function associated to the computation of the arbitrary control torque Γ_{null}. In the context of minimally invasive surgery, Γ_{null} will be computed in order to force to zero the distance between the instrument and a given location of the trocar. It will result in internal joint motions guaranteeing that the instrument will respect the penetration point constraint.

In [8], the use of the optimization term of the inverse kinematic general solution allows to optimizing internal motions to achieve secondary tasks such as keeping the robot away from the joint limits. In our case, we choose the control torque Γ_{null} as the gradient $\nabla_q \phi(q)$ of a scalar positive definite function $\phi(q)$ of the joint position weighted by a negative term α that ensures the decreasing of $\phi(q)$. If this function is expressed as the projection of the penetration point on the instrument held by the robot, then the global control torque forces the robot to decrease this distance, and thus to respect the constraint. Given

$$\Gamma_{null} = \alpha\nabla\phi \tag{12}$$

the expression of the global torque is

$$\Gamma = J^T F + (I_N - J^T \bar{J}^T)\alpha\nabla\phi \tag{13}$$

where

$$\nabla\phi(q) = [\frac{\partial\phi(q)}{\partial q_1} .... \frac{\partial\phi(q)}{\partial q_N}]^T \tag{14}$$

Equation (11) can be written as

$$\Gamma = J^T \Lambda(\ddot{X} - \dot{J}\dot{q} - J A^{-1}(\alpha\nabla\phi - b - g)) + \alpha\nabla\phi \tag{15}$$

Fig. 1 illustrates the control scheme: the input X^d is the desired tool position and the input X_{Tr} corresponds to the penetration point position assumed to be fixed in the base frame. A proportional-derivative control is used for controlling the instrument.

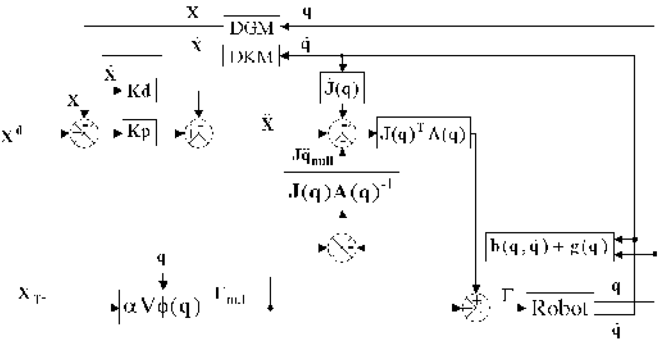


Fig. 1. Dynamic decoupling scheme. DGM and DKM stand respectively for direct geometric and kinematic models.

3 Experimental results

3.1 Experimental platform

We have implemented the algorithm on the D2M2 (Direct Drive Medical Manipulator) robot (Fig. 2). This robot has been designed for beating heart MIS. It is equipped with direct drive actuators to provide high dynamics and low friction.

The controller is a Pentium III 500 MHz with 256 Mo RAM PC running under RTX/Windows 2000. The sample rate is 1 KHz. D2M2 may be teleoperated via UDP communication with a Phantom 1.5 arm working as a master device under Windows XP.

The kinematic architecture of D2M2 is shown in Fig. 3: the first three joints have a Scara disposition (a prismatic joint, then two revolute joints, with parallel axes); the wrist has two orthogonal revolute joints; the instrument (supposed to have three dof to provide full intracorporal distal mobility) is attached on a force/torque sensor fixed on the wrist. The gravity is compensated by a counterweight mounted on the 1st axis. The models have been computed with the SYMORO (SYmbolic MOdeling of RObots) software [8] from the geometrical Denavit-Hartenberg parameters given in Table 1 and the dynamic parameters obtained from CAD data (Table 2).

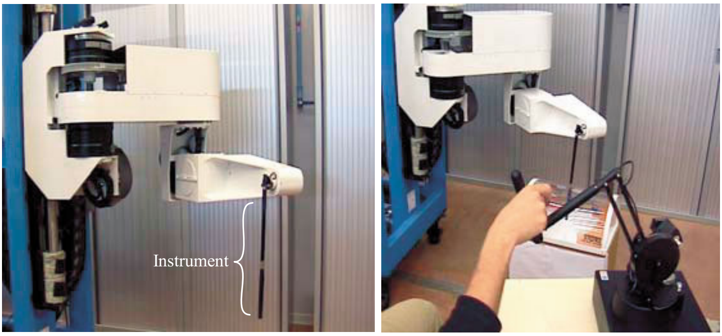


Fig. 2. The 5-dof robot D2M2 and the Phantom master device.

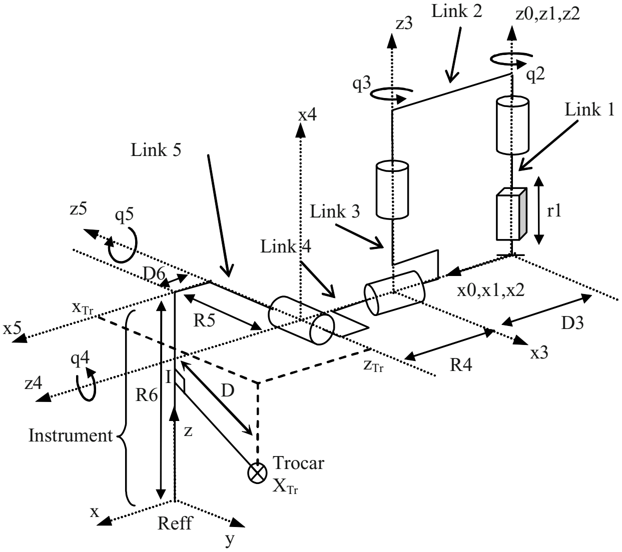


Fig. 3. Kinematic architecture of D2M2.

Table 1. D2M2 geometrical Denavit-Hartenberg parameters.

Joint j	Type	αj	dj (m)	θj	rj (m)
1	Prismatic	0	0	0	r1
2	Revolute	0	0	q2	0
3	Revolute	0	D3 = 0.4	q3 + π/2	0
4	Revolute	π/2	0	q4 + π/2	R4 =0.451
5	Revolute	-π/2	0	q5 - π//2	R5 = 0
Reff	Constant	-π//2	D6	0	-R6 = -0.405

Table 2. D2M2 dynamic parameters. The units used are the SI units: mass Mj in Kg, moment of inertia XXj, YYj, ZZj in Kg.m^2, first moment MXj, MYj, MZj in Kg.m, moment of inertia of rotor IAj in Kg or Kg.m^2 according to the type of joint, Coulomb friction FSj in Kg.m.s^{-2} or Kg.m^2.s^{-2}, viscous friction FVj in Kg. s^{-1} or Kg.m^2.s^{-1}.

M1 = 17.122	M3 = 1.654	XX4 = 0.726	ZZ5 = 0.0187
IA1 = 4	ZZ3 = 0.0232	YY4 = 0.726	MX5 = 0
FV1 = 1	MY3 = 0.109	ZZ4 = 0.0117	MY5 =-0.075
FS1 = 0	IA3 = 0.012	ZZ4 = 0.0009	MZ5 = 0
M2 = 7.726	FV3 = 0.5	IA4 = 0.0025	IA5 = 0.0025
ZZ2 = 0.359	FS3 = 0	FV4 = 0.05	FV5 = 0.0500
MX2 = 1.684	M4 = 6.123	FS4 = 0	FS5 = 0
IA2 = 0.015	MX4 = 0	M5 = 0.3	
FV2 = 0.9	MZ4 = -2.109	XX5 = 0.018	
FS2 = 0	MY4 =-0.075	YY5 = 0	

3.2 Cost function

The function ϕ is chosen as the square of the distance D between the fixed penetration point (trocar position X_{Tr}) and the instrument held by the robot (Fig. 3). This function, depending on q and X_{Tr}, is expressed in the frame 5 by:

$$\phi(q,X_{Tr}) = D(q,X_{Tr})^2 \tag{16}$$

$$\phi(q,X_{Tr}) = ({}^5x_I(q,X_{Tr}) - {}^5x_{Tr})^2 + ({}^5y_I(q,X_{Tr}) - {}^5y_{Tr})^2 + ({}^5z_I(q,X_{Tr}) - {}^5z_{Tr})^2$$

where subscript $_I$ denotes the projection of the trocar position on the instrument.

3.3 Real implementation

Basic paths for MIS such as straight line, circle and helix have been performed (Fig. 4). The gains of the control algorithm have been tuned from simulation results such that $Kp = 3000$, $Kd = 25$, and $\alpha = -4600$.

For a 20 cm straight line path, Fig. 5. shows that the tracking error is less than 2.10^{-2} m (left), and the distance between the trocar and the instrument is less than 2.10^{-2} m (right). The maximum velocity and acceleration reached by the tool are respectively 0.16 m.s^{-1} and 0.25 m.s^{-2}. The generated force on axis 1 and torques on the other axes (Fig. 6) remain within the range of capabilities of the actuators.

Results are similar with circular and helical paths. Fig. 7 shows the Cartesian error for a helical path.

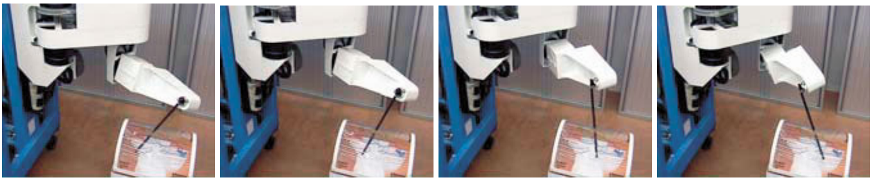


Fig. 4. Straight line motion of D2M2 under constraint of trocar.

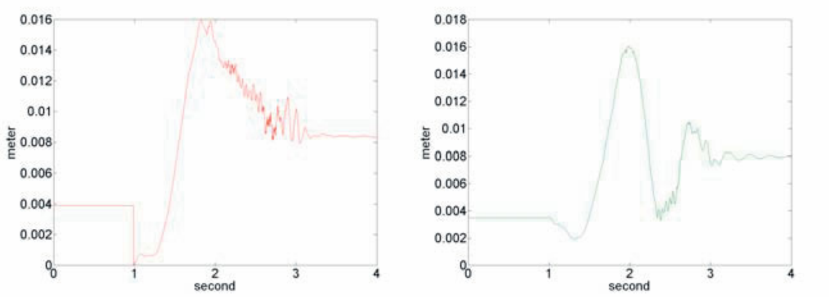


Fig. 5. Tracking error (left) and distance trocar–instrument (right) for a straight line path. $\alpha = -4600$.

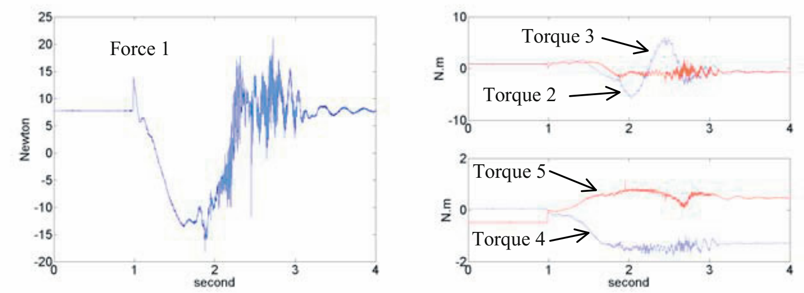


Fig. 6. Force on axis 1 and torques on other axes during the straight line path.

4 Discussion

The comparison of the distance D obtained with different values of α shows that D decreases significantly when the absolute value of α increases. The tracking error is not influenced by the term α, depending rather on the gains Kp and Kd. The control torques remain within the physical limits. However, we notice that torque oscillations increase with the absolute value of α, especially in certain joint configurations of the robot. This should be partially improved after running a proper identification procedure of the dynamic parameters. The linear guide of axis 1 will also been strengthened, which should reduce dramatically the structural vibrations induced at high velocity. But a major improvement has been obtained on the steady state behavior by reducing the sampling rate from 1 ms to 0.7 ms. This allows a better estimation of the joint velocities. As the computation time is less than 0.35 ms, the objective is yet to decrease the sampling rate up to 0.5 ms.

We have also shown that oscillations in the tracking error at the tool tip could be significantly reduced by damping the lateral motion of the instrument within the trocar (Fig. 9). This is done by adding a damping term to the cost function of equation (16)

$$\phi(q,\dot{q},X_{Tr}) = \alpha_1 D(q,X_{Tr})^2 + \alpha_2(\dot{X}_{Tr} + \dot{Y}_{Tr}) \qquad (17)$$

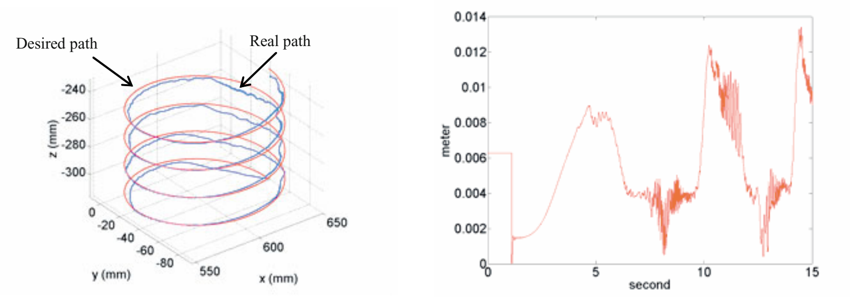


Fig. 7. Real and desired helical paths (left), and corresponding tracking error (right)

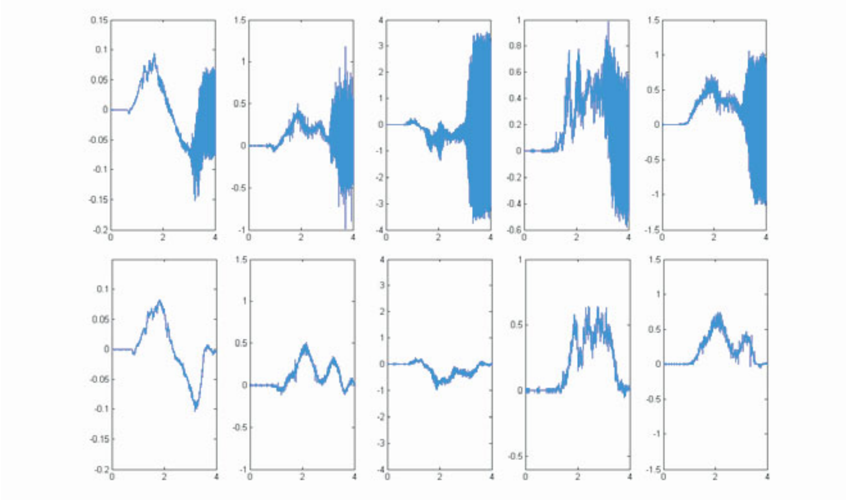


Fig. 8. Plot of joint velocities (1st column: axis 1 in m.s^{-1}; the other columns: axes 2 through 5 in rd.s^{-1}) with respect to time (s). The sampling period is 1 ms (1st row), and 0.7 ms (2nd row). $\alpha = -5100$.

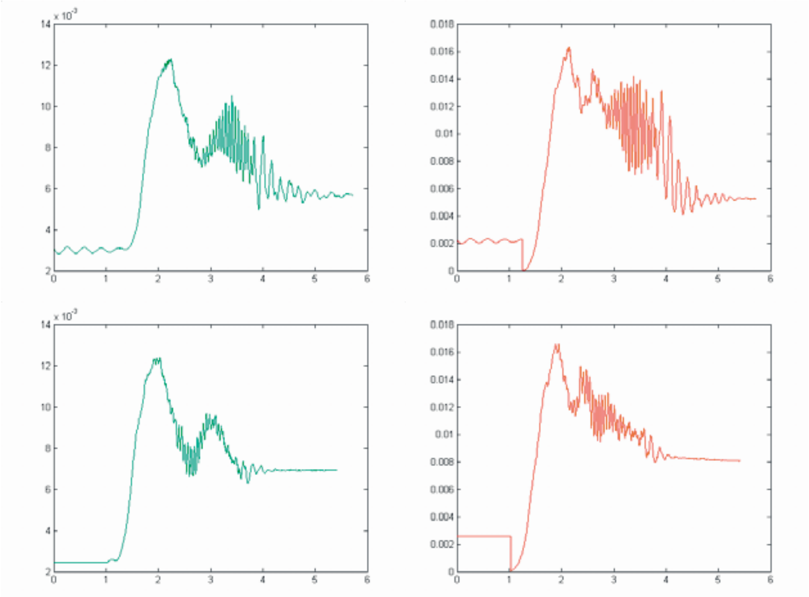


Fig. 9. Distance (m) trocar-instrument (left column) and tracking error (m) at the instrument tip (right column) with respect to time (s) for a straight line path: lateral motion within the trocar without damping (1st row) and with damping (2nd row). $\alpha = -7000$.

5 Conclusion

We have presented an efficient dynamic decoupling control scheme based on the optimized choice of the arbitrary control torque expressed in the null space. In the context of MIS, this control allows a redundant serial robot to achieve motions guaranteeing that the instrument passes through a penetration point materializing the trocar position. The optimization term of the global solution of the static model generates internal motions that tend to reduce the distance between the trocar and the instrument. The inputs of the control structure being the instrument tip position and the trocar position, we have shown on the real robot D2M2 that it was possible to practically decouple the task behavior (tool motion) from the posture behavior (satisfaction of the trocar constraint).

References

[1] http://www.computermotion.com.

[2] Guthart, G. and J.K. Salisbury. *The Intuitive Telesurgery System: Overview and Applications*. in *IEEE Int. Conf. on Robotics and Automation*, April 2000, San Fransisco, USA, pp. 618-621.

[3] Madhani, A.J., G. Niemeyer, and J.K. Salisbury. *The Black Falcon: A Teleoperated Surgical Instrument for Minimally Invasive Surgery*. in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, October 1998, Victoria B.C., Canada, pp. 936-944.

[4] Madhani, A.J. *Design of Teleoperated Surgical Instruments for Minimally Invasive Surgery*. PhD Thesis, Massachusetts Institute of Technology, Cambridge, USA, 1998.

[5] Rininsland, H. *ARTEMIS. A Telemanipulator for Cardiac Surgery*. European J. of Cardio-Thoracic Surgery, 1999, **16**(2).

[6] Krupa, A., C. Doignon, J. Gangloff, M. de Mathelin, L. Soler, and G. Morel. *Towards Semi-Autonomy in Laparoscopic Surgery Through Vision and Force Feedback Control*. in. *Int. Symp. on Experimental Robotics, ISER'00*, December 2000, Waikiki, Hawaii, USA, pp 189-198.

[7] Cavusoglu, M.C. and F.T. Tendick. *A Laparoscopic Telesurgical Workstation*. IEEE Trans. on Robotics and Automation, 1999, **15**(4).

[8] Khalil, W. and E. Dombre. *Modeling, Identification and Control of Robots*. Hermes Penton, London, 2002.

[9] Khatib, O. *Commande Dynamique dans l'Espace Opérationnel des Robots Manipulateurs en Présence d'Obstacles*. PhD Thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France, December 1980.

[10] Khatib, O. *A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Fromulation*. IEEE Journal of Robotics and Automation, 1987, **RA –3**(1).

[11] Russakow, J., O. Khatib, and S. M. Rock. *Extended Operational Space Formulation for Serial-to-Parallel Chain (Branching) Manipulators*. in *IEEE Int. Conf. on Robotics and Automation*, May 1995, Nagoya, Japan, pp. 1056-1061.

[12] Chang, K-S and O Khatib. *Operational Space Dynamics: Efficient Algorithms for Modeling and Control of Branching Mechanisms*. in *IEEE Int. Conf. on Robotics and Automation*, April 2000, San Francisco, USA, pp. 850-856.

Appendix

$$\Gamma = J^T(\Lambda\ddot{X} + \mu + \rho) + (I_N - J^T\bar{J}^T)\Gamma_{null}$$

$$\Gamma = J^T(\Lambda\ddot{X} + \bar{J}^T b - \Lambda\dot{J}\dot{q} + \rho) + (I_N - J^T\bar{J}^T)\Gamma_{null}$$

$$\Gamma = J^T\Lambda\ddot{X} + J^T\bar{J}^T b - J^T\Lambda\dot{J}\dot{q} + J^T\rho + \Gamma_{null} - J^T\bar{J}^T\Gamma_{null}$$

$$\Gamma = J^T\Lambda\ddot{X} + J^T\bar{J}^T b - J^T\Lambda\dot{J}\dot{q} + J^T\bar{J}^T g + \Gamma_{null} - J^T\bar{J}^T\Gamma_{null}$$

$$\Gamma = J^T\Lambda(\ddot{X} - \dot{J}\dot{q}) + J^T\bar{J}^T(b + g) + \Gamma_{null} - J^T\bar{J}^T(A\ddot{q}_{null} + b + g)$$

$$\Gamma = J^T\Lambda(\ddot{X} - \dot{J}\dot{q}) + \Gamma_{null} - J^T\bar{J}^T A\ddot{q}_{null}$$

$$\Gamma = J^T\Lambda(\ddot{X} - \dot{J}\dot{q}) + \Gamma_{null} - J^T\bar{J}^T J^T\Lambda J\ddot{q}_{null}$$

$$\Gamma = J^T\Lambda(\ddot{X} - \dot{J}\dot{q}) + \Gamma_{null} - J^T\Lambda J\ddot{q}_{null}$$

$$\Gamma = J^T\Lambda(\ddot{X} - \dot{J}\dot{q} - J\ddot{q}_{null}) + \Gamma_{null}$$

VII. Humanoids

Humanoids are the new addition to the robotics world which became popular in the last decade. A humanoid is complex not only in its construction but also modeling and control. It promotes a better understanding of human body and it also brings us a step closer to replicating human ability in performing a wide variety of tasks.

In the the first article, Yokio and co-workers at AIST introduce for the first time a tele-operated humanoid that operates a backhoe in place of a human operator. This brings together a lot of research and engineering effort to remotely control the humanoid, to protect the humanoid from the shock and vibration of the vehicle seat and "full-body operation" technology that controls the humanoid's total body movement. The last point helps prevent the humanoid from falling over.

As a humanoid possesses complex dynamics, it is often desired to specify the end-effector points of interest. In the second paper, Nishiwaki and colleagues present a method of generating walking motion based of the desired hand motion trajectory. With the hands as the end-effectors of interest, the trajectories for the rest of the body is generated and run simultaneously.

The third article by Mizuuchi *et al* presents the design and development of a reconfigurable humanoid system and a method to acquire body information coupled with a neural network. It shows the capability of realizing complex human body motion through feedback modification method used in conjunction with a neural network.

The only humanoid article in this chapter that comes from outside of Japan applies task planning strategy with Petri-net to coordinate task execution by highly redundant robots possessing multiple sub-systems, such as a humanoid, two-arm system, etc. Asfour and co-workers describe experiments carried out on an ARMAR humanoid that possesses 23 DOFs.

A higher level control is studied in generating the behavior of humanoids. In the last article of the chapter, a behavior imitation method is proposed by Asada and colleagues. Here, the humanoid was to learn and mimic the motion of a human demonstrator. Various efficient methods of representing the complex amount of data are proposed using the Self Organizing Map to produce various efficient posture and motion map representations.

A Tele-operated Humanoid Operator

Kazuhito Yokoi[1], Katsumi Nakashima[2], and Yoshitaka Yanagihara[3]

[1] National Institute of Advanced Industrial Science and Technology(AIST)
Central 2, 1-1-1 Umezono, Tsukuba 305-8568 Japan
Kazuhito.Yokoi@aist.go.jp
http://www.aist.go.jp/

[2] Kawasaki Heavy Industries, Ltd., 1-1, Kawasaki-cho, Akashi, 673-866 Japan
nakasima@tech.khi.co.jp
http://www.khi.co.jp/

[3] Tokyu Construction Co., Ltd., 3062-1 Tana, Sagamihara, Kanagawa, 229-1124 Japan
yanagihara.yoshitaka@tokyu-cnst.co.jp
http://const.tokyu.com/

Abstract. This is the first successful trial in the world to remotely control a humanoid robot so as to drive an industrial vehicle in lieu of a human operator. These results were achieved thanks to the development of the following three technologies: (1) "remote control technology" for instructing the humanoid to perform total body movements under remote control and the "remote control system" for executing the remote control tasks; (2) "protection technology" for protecting the humanoid against shock and vibrations of its operating seat and against the influences of the natural environment such as rain and dust; and (3) "full-body operation control technology" for controlling the humanoid's total body movements with autonomous control to prevent the robot from falling over. The humanoid has promising application potential for restoration work in environments struck by catastrophes and in civil engineering and construction project sites where it can "work" safely and smoothly.

1 Introduction

In recent years, many universities [9][10][12] and some companies [6][8] have produced humanoid robots. However the application area of the humanoids is still limited to research, advertisement, and entertainment.

Ministry of Economy, Trade and Industry (METI) of Japan ran the Humanoid Robotics Project (HRP) from 1998JFY to 2002JFY. The aim of the project was to find some suitable applications for humanoids. Five applications of humanoids were investigated in HRP [7]. For one of them, we developed a tele-operated humanoid that drives an industrial vehicle.

A humanoid has substantial advantages when operating a machine that human beings usually use, because a humanoid can operate the machine without requiring substantial modifications. Furthermore, by using other machines, a humanoid can expand its ability just like a human being. We have selected industrial vehicles as the target, because there is a strong demand for unmanned industrial vehicles that can be used in a hazardous area.

There have been many attempts until the present to robotize industrial vehicles themselves to work in dangerous work areas or in adverse environments

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 229–238, 2006.

[1][2][3][13]. In contrast, the use of a humanoid to operate the industrial vehicle instead of a human operator has two distinct advantages:

- This means that robot does not only drive the vehicle but is also capable of executing the ancillary work tasks (alighting from the vehicle to check the work site, carrying out simple repairs, etc.) and
- it permits the robotizing of all industrial vehicles without modification.

In this paper, we describe our attempt towards a tele-operated humanoid operator. The rest of the paper is organized as follows: In Section 2, we present humanoid robot HRP-1S and its protective clothing. In Section 3, we describe a remote control device of HRP-1S and some important items for the remote operation of a backhoe by a tele-operated humanoid such as avoidance of self-collision and vision assistance. Section 4 shows results of experiments. In Section 5, we conclude the paper.

2 Humanoid HRP-1S and its Protective Clothing

We introduced the humanoid robot HRP-1S to make a tele-operated humanoid operator. HRP-1S has 1600 mm height, 600 mm width, and 117 kg weight including batteries. It has 12 d.o.f. in two legs and 16 d.o.f in two arms including hands with 1 d.o.f. grippers. Each foot and wrist is equipped with a force/torque sensor. Two video cameras are mounted in the head.

When we employ a humanoid robot to drive an industrial vehicle outdoors in lieu of a human operator, we should protect the humanoid against the influences of the natural environment such as rain and dust. HRP-1S is not manufactured to work in the open air. In order to meet these requirements, we developed a protective clothing (Fig. 1).

We selected a nylon fabric for good ventilation and stretch as the fabric of the protective clothing. It consists of a hood, a vest, sleeves, pants, and boots. By carefully designing the shape of each part of the clothing and the structure of the seam, ventilated, drip-proof, and dust-proof capabilities are realized. Also the protective clothing sets no limitation to the motion range of each joint of the robot. The drip-proof capability was tested in a climate chamber and we confirmed that the clothing keeps drip-proof capability even in the heavy rain.

Fig. 1 also shows the control hardware system of HRP-1S. The real-time controller runs on the CPU board in the backpack of HRP-1S, whose operating system is ART-Linux. The motion control software of HRP-1S is developed by AIST in order to realize its whole body movement simultaneously [11].

3 Remote Control Device and Technology

We developed a remote control device for HRP-1S as shown in Fig. 2. The motion of arms, legs and head of HRP-1S can be operated remotely by using the control device.

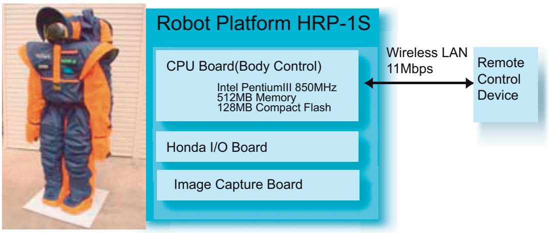


Fig. 1. Humanoid HRP-1S with Protective Clothing and its Control System

The arms of HRP-1S need to be operated precisely for manipulating control levers of an industrial vehicle, so we introduced master arms as the arm operation devices.

Commands such as "Walk forward five meters!" can be preprogramed and executed it by using a switch or voice command. However, in some situations, we strongly want to tele-operate both arms and legs simultaneously. For this reason, we introduced a special device that can be operated by foot in order to tele-operate the legs. We call this the "master foot".

The head of HRP-1S is operated with a small pointing device mounted on the master arm. The details of the master arm device and the master foot device are described next.

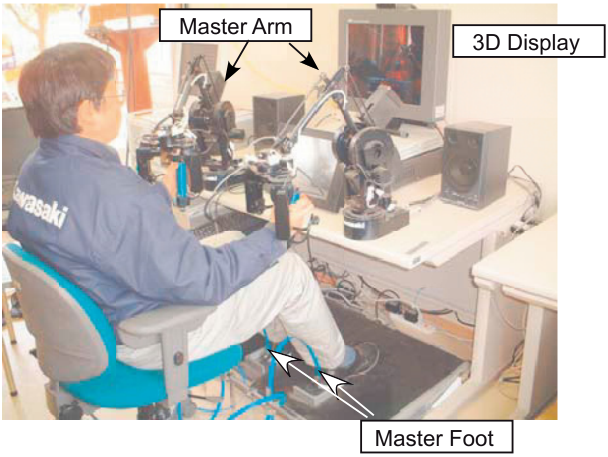


Fig. 2. Portable Remote Control Devices

3.1 Master Arm Device

There are many remote control devices proposed [14]. We have also developed a master arm device at phase one of HRP [15]. It has good performance, but it is too heavy to transport to an operation site. Another six d.o.f. master arm device was developed.

Direct drive motors are introduced as drive motors of the master arm because of their low friction performance and counter weights are equipped with a link mechanism for gravity balancing. To operate the master arm easily, the pantograph mechanism, which clears out interaction between translation and orientation motion of the grip, is installed on the master arm.

The motion of the hand of HRP-1S is operated according to the position and orientation of the grip of the master arm. The force and moment on the wrist of HRP-1S are fed back to the operator through the master arm.

3.2 Master Foot Device

Two tape type sensors were introduced for the master foot as shown in Fig.2. Each sensor is attached to each ankle of the operator and detects the position of the ankle. The weight of the tape sensor is so light that the operator feels free and easily equips.

The main purpose of the master foot is to operate the legs at the same time as when the robot arms are operated with the master arms.

The leg of the robot is controlled mainly parallel with the ground. For this reason, we restricted the input from the master foot only to position input. As the human operator feels force at the foot less precisely than at the hand, force feedback function is not installed on the master foot control.

3.3 Remote Control Technologies

Some sophisticate concepts, such as "Supervisory Control [16]" and "Shared Autonomy [17]", have been proposed for a tele-operation system.

When a humanoid robot is tele-operated in various kinds of situation, the operator often wants to change tele-operation method to a suitable one for easy operation. We selected three operational modes, as shown below, to suit situations when the robot steps into, sits down and manipulates levers.

1. The playback mode: The robot is controlled according to a reference motion pattern that was generated can be off-line. We use this mode for seating. Note that the seating motion pattern can be reproduced easily in real-time by setting the height of the seat.
2. The manual mode: The robot is manually tele-operated by using a remote control device. Both the master arm device and the master foot device can be used. It is used for the arm and leg control after seating and the head control.
3. The semi-automatic mode: The operator gives the robot a partial command of a whole body motion and the rest of the motion is automatically calculated. The main purpose of this mode is to reduce the responsibility of the operator in the manual operation. We applied this mode for walking to and stepping into the backhoe.

Next, we describe two functions of the semi-automatic mode.

3.4 Avoidance of Self-collision of Arms

Even if a shape of a humanoid robot is similar to the human's one, it is difficult for the human operator to predict self-collisions between the arm and other parts of the humanoid. We introduced a semi-automatic function; the operator commands the position and orientation of the hand of the robot, and the control system of the robot automatically avoids a self-collision.

HRP-1S has 7 d.o.f. arm that has one d.o.f. redundancy. It can change its elbow position while keeping the same hand position and orientation in the world coordinate system. The control system rotates the elbow point about the line joining the shoulder to wrist to avoid self-collision [18].

The operator commands the position and orientation of the each hand using the master arm and a default value of the each elbow-rotating angle using the switch on the master arm. In the semi-automatic mode, the elbow-rotating angle is updated automatically to avoid self-collisions between the arm and other parts of the robot.

3.5 Vision Assistance

Because the cockpit of the backhoe has complicated shapes with levers and HRP-1S has a big backpack, the route for the robot to approach the seat is very restricted. It is hard for a human operator to select a correct target position through which the robot should go. We introduced a vision assistance function for stepping into easily.

We put four landmarks on the cockpit. The operator commands HRP-1S to get an image from the camera installed in the head. The position of the landmark is detected by processing the image and then the system automatically calculates a suitable target position in order to prevent the collision between the body of the robot and the cockpit. Then the operator just confirms the position and the system executes the walk.

4 Experiments

At first, we succeeded in operating the proxy drive of a lift truck in a standing posture by a tele-operated humanoid using the same system as shown in Fig. 3 (left) [4]. The time to carry out the model tasks, that include the sequence of transferring a load on the pallet, is about 300 [sec]. It took approximately three times longer than the human operator did.

Second, we tried to drive a backhoe as shown in Fig. 3 (right), which is the most typical construction machine, by a humanoid [5].

The cockpit of backhoe was slightly reconstructed, because HRP-1S is a little bit fat and has very limited movable area of its four limbs. We believe rather strongly that a future humanoid robot will not require these reconstructions.

The points of reconstruction are flattening the floor of the cockpit for the robot and moving operation levers to a position where the vision system can recognize and within the joint angle limitation of both arms. The cockpit was also reconstructed because of the protection seat installation.

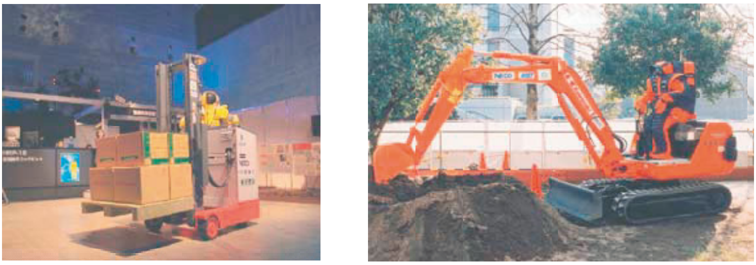


Fig. 3. Teleoperated humanoid drives a lift truck (left) and a backhoe (right)

The protective seat is a seat that protects the body of the humanoid from shock during seating and machine vibration while driving. By comparing the results of performance evaluation examinations and numerical analyses with a finite element method, we finally selected SUNPELCA^{TM} (Chemical cross-linking closed-cell PE, EVA foam) as a shock absorbent material and KG Gel^{TM} (thermoplastic gel) as vibrational absorbent material for the seat.

We have executed the following tasks using the tele-operated humanoid HRP-1S wearing the protective clothing in the experiment.

1. Getting on the cockpit: The humanoid went up the boarding bridge with preprogrammed walking pattern. Then the robot walked to the front of the seat of the backhoe. At this sequence, the operator used the vision assistance function to obtain suitable target positions and commanded them to the robot. The real-time walking pattern generator of the local control system of the robot combined side-walking, turning, stepping forward, and stepping backward automatically to reach the target position (Fig. 4).
2. Seating on the protective seat: HRP-1S sat down in two steps; 1) the robot puts its hip on the seat, 2) the robot puts its legs forward for increasing the sitting stability. The robot bent its upper body forward and put its hip back and down. After contacting with the seat, the robot made its upper body upright (Fig. 5). At the second step, HRP-1S put its legs forward in order to enlarge the supporting area made from both feet and the hip. This made the robot seat more stable. HRP-1S put down its right hand onto a side bar to support some of its weight and extended the supporting area. After that, HRP-1S lifted up its right leg, put it forward, and down (Fig. 6). Figure 7 shows the trajectory of right foot position, feet force and right hand force. When the right foot was lifted up, the reaction force of the right hand increased. When the right foot was put down, the reaction force of the right hand decreased to zero. On the other hand, when HRP-1S lifted up its left leg, it was no need to put down its left hand. Because the center of gravity of the robot was located in the supporting area made from its right leg and the hip.
3. Driving and excavating: In the cockpit of the backhoe, there are two travel levers and two control levers. The robot grasps both travel levers and tilts them forward

and drives the backhoe forward. By changing the tilt angle of the control levers through the tele-operated robot arm, the operator changes the rotational speed of the crawler of the backhoe and turnes the backhoe in the target direction. Figure 9 shows the trajectory and force vector of the left hand during the lever operation. The remote operator estimated the direction of the lever operation by feeling the feedback force. When the backhoe arrived at its destination, the operator tele-operated the robot's arm to grasp the control levers. Finally, the operator moved both control levers through the tele-operated robot and succeeded in excavations (Fig. 8).

In order to evaluate the workability, the time required to do total operation through HRP-1S was compared with the time required for the human operator seated at the cockpit directly to do the same operation. The time required for the experiments using HRP-1S is up to 3 times longer than it for the human operator.

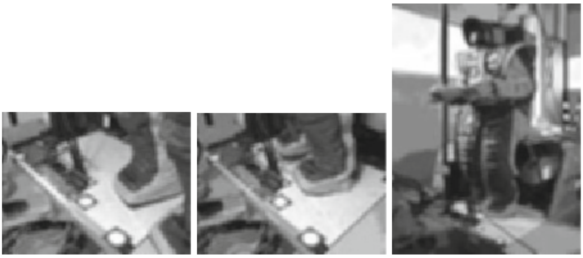


Fig. 4. The motion of riding on the backhoe

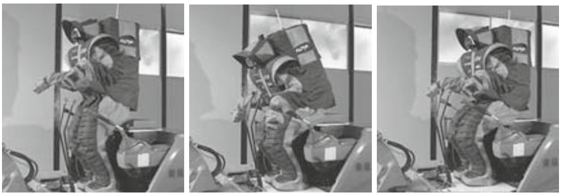


Fig. 5. Seating Motion

5 Conclusions

This paper presented the application of a tele-operated humanoid to tele-operate an industrial vehicle. This is the first successful trial in the world to remotely control a man-emulating robot to drive an industrial vehicle (backhoe) outdoors in lieu of

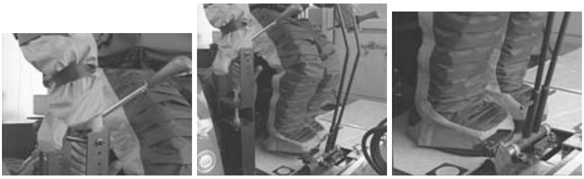
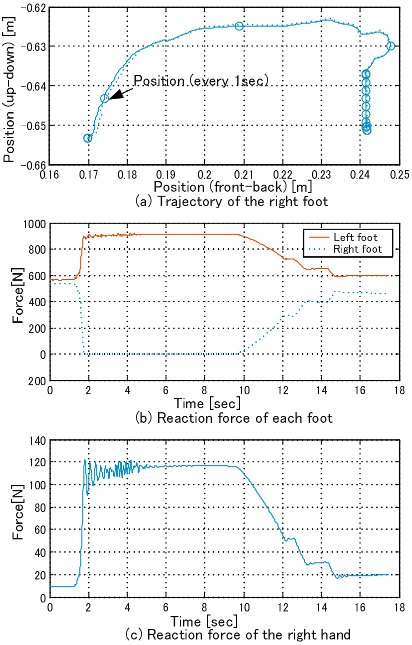


Fig. 6. The right hand putting on a device and the foot moving forward



(a) Trajectory of the right foot

(b) Reaction force of each foot

(c) Reaction force of the right hand

Fig. 7. The trajectory of right foot position and trend data of feet force and right hand force

Fig. 8. The tele-operated HRP-1S driving the backhoe in the open-air

a human operator. Furthermore, the robot’s operation was controlled while it wore protective clothing to protect it against the rain and dust. This too marks a world-first success demonstrating the robot’s capability of performing outdoor work even in the rain.

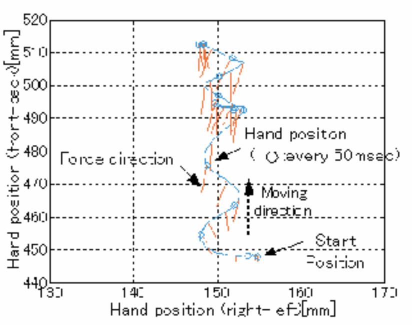


Fig. 9. The trajectory and force vector of the left hand at lever operation of the backhoe

The major insight gained from this success that has demonstrated a humanoid's ability to replace the human operator in operating commercially used industrial vehicles under remote control is the realization that humanoids are capable of moving in the same manner as humans. The humanoid's ability to carry out outdoor work tasks even in the rain by "wearing" protective clothing has widened the scope of the environmental conditions in which it is capable of executing work. From these two aspects there is every reason to expect that these results will make a substantial contribution toward the realization of practical work-performing humanoids.

When humanoid robots engaged in other types of work can be used for operational duties normally performed by human operators there will be a definite chance for a greater expansion of the humanoid market which in turn holds promise of further reductions in their production and operating costs.

Acknowledgments

We thank Manufacturing Science and Technology Center (MSTC), New Energy and Industrial Technology Development Organization (NEDO), Ministry of Economy, Trade and Industry (METI) for their entrusting development of the project "Humanoid Robotics". We express a great thank Mr. Kobayashi, Mr. Mifune, Mr. Hasunuma of Kawasaki Heavy Industries; and Mr. Ueno, Mr. Ohya, Mr. Gokyu, Mr. Endo of Tokyu Construction for their cooperation of R&D in the project. We also thank Prof. Inoue of the University of Tokyo; Dr. Tanie, Dr. Hirukawa, and Dr. Kanehiro of AIST; and Prof. Yokokohji of the Kyoto University for their various instructive advices.

References

1. E. Gambao and C. Balaguer, "Robotics and Automation in Construction," *IEEE Robotics and Automation Magazine,* vol. 9, no. 1, pp. 4–6, 2002.
2. R. A. Jarvis, "Sensor Rich Teleoperation of an Excavating Machine," *Proc. of Field and Service Robotics*, pp. 238–243, 1999.

3. S. E. Salcudean, et al., “Bilateral Matched Impedance Teleoperation with Application to Excavator Control,” *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 133–139, 1998.
4. H. Hasunuma, et al., “A Tele-operated Humanoid Robot Drives a Lift Truck,” *Proc. of IEEE Int. Conf. on Robotics and Automation,* pp. 2246–2252, 2002.
5. K. Yokoi, et al., “A Tele-operated Humanoid Robot Drives a Backhoe in the Open Air,” *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Sytems,* pp. 1117–1122, 2003.
6. M. Hirose, Y. Haikawa, and T. Takenaka, "Introduction of Honda Humanoid Robots Development," in *Proc. of Advanced Science Institute 2001*, No. 16, pp. 1–8.
7. H. Inoue, et al., "Overview of Humanoid Robotics Project of METI," in *Proc. of 2001 Int. Symp. Robotics*, pp. 1478–1482.
8. T. Ishida, et al., "Motion Entertainment by a Small Humanoid Robot based on Open-R," in *Proc. 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Sytems*.
9. S. Kagami, et al., "Design and Implementation of Software Research Platform for Humanoid Robotics : H6," in *2001 Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2431–2436.
10. J. Yamaguchi, et al., "Development of a Bipedal Humanoid Robot - Control Method of Whole Body Cooperative Dynamic Biped Walking," in *Proc. of 1999 IEEE Int. Conf. on Robotics and Automation*, pp. 368–374.
11. K. Yokoi, et al., "A Honda Humanoid Robot Controlled by AIST Software, " in *Proc. of IEEE-RAS Int. Conf. Humanoid Robots 2001*, pp. 259–264.
12. M. Gienger, et al., "Towards the Design of a Biped Jogging Robot," in *Proc. of 2001 IEEE Int. Conf. on Robotics and Automation*, pp. 4140–4145.
13. W. R. Hamel and P. Murray, "Observations Concerning Internet-based Teleoperations For Hazardous Environments," in *Proc. of 2001 IEEE Int. Conf. on Robotics and Automation*, pp. 638–643.
14. G. C. Burdea, *Force and Touch Feedback for Virtual Reality,* John Wiley & Sons, 1996.
15. H. Hasunuma, et al., "Teleoperation Master-arm with Gripping Operation Devices," in *Proc. of 9th Int. Conf. on Machine Automation*, pp. 567–572, 2000.
16. W. R. Ferell and T. B. Sheridan, "Supervisory Control of Remote Manipulation," *IEEE Spectrum,* pp. 81–88, 1967.
17. S. Hirai, et al., "Integration of a Task Knowledge Base and a Cooperative Manuvering System," in *Proc. of 1990 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* pp. 349–354, 1990.
18. J. Hollerbach, "Optimum Kinematic Design for a seven degree of freedom Manipulator," in *Robotics Research: the Second Int. Symp.* MIT Press, pp. 215–222, 1985.

Hand-position Oriented Humanoid Walking Motion Control System

Koichi Nishiwaki[1], Yasutaka Fukumoto[2], Satoshi Kagami[1], Masayuki Inaba[2], and Hirochika Inoue[1]

[1] Digital Human Research Center, AIST
2-41-6, Aomi, Koto-ku, Tokyo, 135-0064, Japan
k.nishiwaki, s.kagami, h.inoue@aist.go.jp

[2] Dept. of Mechano-Informatics, The Univ. of Tokyo,
7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan
yasutaka, inaba@jsk.t.u-tokyo.ac.jp

Abstract. This paper addresses a method of constructing a control system that realizes desired absolute hand position trajectory while walking. Controlling hand position in the absolute coordinate system with short delay while walking is a important function for realizing such motions as visual feedback reaching while walking, walking using handrail, and opening doors while walking. Based on previously developed online walking control system, fast dynamically stable walking trajectory generation system that preserves designed hand trajectory in absolute coordinate system, and sensor feedback balance maintaining system that also preserves given hand position are developed. Then a torso transition planner that generates desired torso movement of every one step from the absolute hand trajectory is implemented. A hand position operation experiment with "3D mouse", a reaching experiment with visual feedback, and a guiding experiment using 6-axis force sensor at wrists are shown to demonstrate the performance of the system.

1 Introduction

Researches on biped walking of humanoid robots have been carried out actively, and robust walking control and walking that follows online commands are achieved (ex. [1–5]). We also constructed walking control system that follows online commands, such as, walking direction, step cycle, and upper body motion[6].

Usual humanoids have more degrees of freedom than it is required for walking, and expected to make use of many degrees of freedom efficiently for whole-body motion in human environment. The ability of using hands for other tasks while walking will be useful, as it realizes such motions as visual feedback reaching while walking, walking using handrail, pushing handcart, and opening doors while walking.

In order to realize those motions, we developed a online walking control system that accept desired hand position in absolute coordinate system with up to 1 [kHz] cycle. (It is same as motor servo control cycle.) Then we also developed torso transition planning system that generates desired torso movement of every one step from the hand movement information. Desired hand position trajectory will be the only input, and torso motion of every step and footprints are automatically generated from the input by the system.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 239–248, 2006.

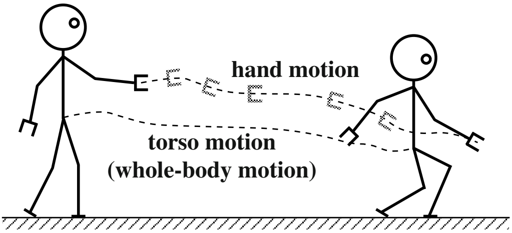


Fig. 1. Concept of hand-position oriented walking control.

2 Hand-position Oriented Walking Control

2.1 Approach and Overview of System

Our goal of this work is to realize a hand-position oriented humanoid walking control system that accepts the desired hand position trajectory in global coordinate system as input and generates and controls whole-body walking motion that satisfies the input (Fig. 1). Here obstacles around the robot are not considered, and the level plane is assumed for the ground in this work.

Almost humanoid walking is realized in 2 stages: dynamically stable motion trajectory generation using dynamics model, and sensor feedback modification of the trajectory. We have realized walking control system that follows online commands, such as, walking direction, step cycle, joint angle based arm motion. Dynamically stable motion trajectory generation is made online, and a planner that generates footprints from desired walking direction is implemented to realize the system[6].

We developed the hand-position oriented walking system based on this system. The overview of the system is shown in Fig. 2. Desired hand trajectory is passed to “Torso Transition / Footprint Planner”. The planner generates footprints from the information every one step. Then “Walking Pattern Generator” makes dynamically stable walking trajectory that realizes given footprints and hand position trajectory. This trajectory generation is carried out just after the footprint planning in the same cycle. As the trajectory generation is carried out in simulation environment, generated trajectory is modified while execution using sensor information in order to maintain dynamic balance. This is the role of the “Sensor Feedback Balance Compensator”. This compensator changes the horizontal torso position, but it does not change the hand position in absolute coordinate system from the given trajectory. It runs every 1[ms]. If this compensator accept the desired hand position from the walking pattern generator, the update cycle of the desired hand position of whole system will be 1 step, and it is too large for sensor feedback and remote control system. Therefore the compensator is designed to accept the desired hand trajectory directly from the input. Therefore the update cycle of the desired hand position is 1[ms].

2.2 Related Works

Several works have been done focusing on a hand when dealing with whole-body motions. Inoue et al. proposed a method that changes body posture and makes a step in order to keep manipulability at hands[7]. Sian et al. have implemented a

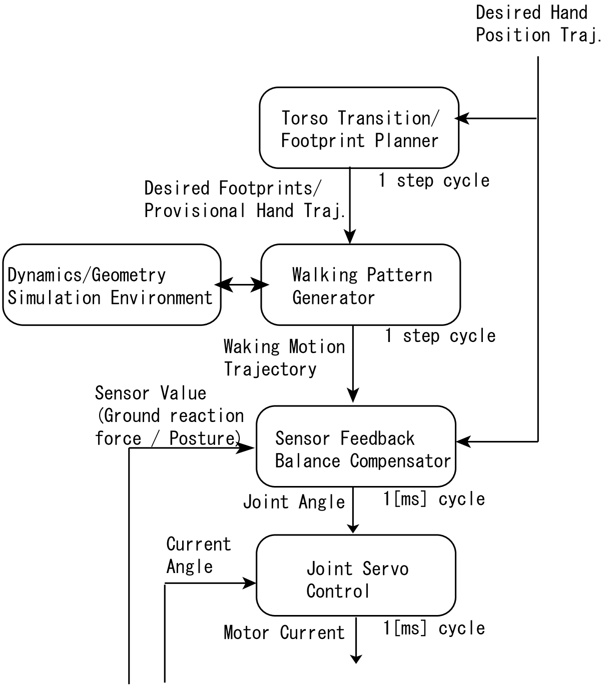


Fig. 2. Overview of the hand-position oriented walking control system.

"switching command based tele-operation system" which allows the operator to only manipulate specific points of the robot's body for generating stable whole-body motions[8]. Cooperative carrying of a panel with human and humanoid is presented by HRP2. Impedance control is employed for arm control and the hand position is not directly controlled in this demonstration[9].

We discuss how to realize desired hand trajectory while continuous walking is being executed in this paper.

3 Walking Pattern Generation

In this section, we will explain the "walking pattern generator". We adopted fast generation method of motion pattern that follows desired ZMP[10].

In general, ZMP trajectory can be analytically led from the robot motion trajectory, however leading robot trajectory that satisfies given ZMP trajectory analytically is difficult, since it has to solve non-linear, interfered 2^{nd} order differential equation with joint constraints. A fast generation method of walking trajectory that follows given ZMP trajectory by modifying horizontal torso trajectory from given initial trajectory is explained, in the following part.

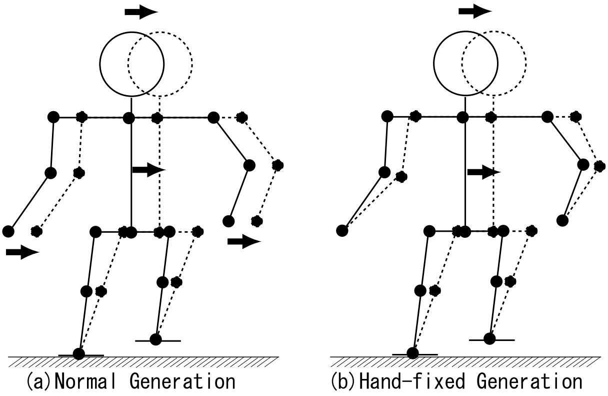

(a) Normal Generation (b) Hand-fixed Generation

Fig. 3. Trajectory modification to obtain dynamically stable one.

3.1 Linearlize and Non-Interferencize the Dynamics Equation about ZMP

Let i^{th} robot link position, mass, inertia tensor, angular velocity vector be $\mathbf{r}_i = (x_i, y_i, z_i)^{\mathrm{T}}, m_i$, $\mathbf{I}_i$, ω_i, and gravity be g. Set the coordinates as x-y plane be the ground and as z-axis be negative gravity direction. Let ZMP be $\mathbf{P} = (x_p, y_p, 0)^{\mathrm{T}}$.

Then equation to lead ZMP from robot motion is described as follows (the same for y_p);

$$x_p = \frac{\sum m_i z_i \ddot{x}_i - \sum \{m_i(\ddot{z}_i + g)x_i + (0,1,0)^{\mathrm{T}}\mathbf{I}_i\dot{\boldsymbol{\omega}}_i\}}{-\sum m_i(\ddot{z}_i + g)}. \tag{1}$$

Let robot trajectory be described as follows;

$$\begin{aligned}\mathbf{A}(t) = (&x_1(t), y_1(t), z_1(t), \theta_1(t), \phi_1(t), \psi_1(t),\\ &\cdots, x_n(t), y_n(t), z_n(t), \theta_n(t), \phi_n(t), \psi_n(t)).\end{aligned} \tag{2}$$

ZMP trajectory $\mathbf{P}_A(t) = (x_{p_a}(t), y_{p_a}(t), 0)^T$ can be solved by Eq. (1). Now consider to generate trajectory that follows desired ZMP $\mathbf{P}^*_A(t)$ by only modifying $x_i(t), y_i(t)$ in $\mathbf{A}(t)$ to $x^{'}_i(t), y^{'}_i(t)$.

$$x^*_p = \frac{\sum m_i z_i \ddot{x}^{'}_i - \sum \left\{m_i(\ddot{z}_i + g)x^{'}_i + (0,1,0)^{\mathrm{T}}\mathbf{I}_i\dot{\boldsymbol{\omega}}_i\right\}}{-\sum m_i(\ddot{z}_i + g)}. \tag{3}$$

This problem is solving $x^{'}_i$ that satisfies Eq. (3) (the same for $y^{'}_i$). Eq. (1) $-$ Eq. (3) gives the following equation;

$$x^e_p = \frac{\sum m_i z_i \ddot{x}^e_i - \sum m_i(\ddot{z}_i + g)x^e_i}{-\sum m_i(\ddot{z}_i + g)}. \tag{4}$$

Here, $x^e_p = x^*_p - x_{p_a}, x^e_i = x^{'}_i - x_i$.

3.2 Fixing feet and hands

Since there is redundancy, consider $x_i^e = x^e$, that is, modifying the horizontal position of all the link in same distance. In reality, feet position can not be changed relative to the ground, however upper body can satisfy this modification. When modifying upper body position in horizontal plane, the position changes of leg links, which are led by inverse-kinematics, are approximately proportional to the upper body in horizontal plane and little for rotational and vertical component.

Therefore in the previous implementation we changed the upper body position in same distance, and leg link position is changed according to analytical (not redundant) inverse kinematics (Fig. 3(a)). For this work position of hands also should not be changed from initial trajectory. Therefore position of hands is also fixed, and position of arm links is decided by the Jacobian-based inverse kinematics (Fig. 3(b)).

3.3 Numerical Solution of Differential Equation

Set $x_i^e = x^e$ to Eq. (4), then following equation is obtained;

$$-\frac{\sum m_i z_i}{\sum m_i(\ddot{z}_i + g)}\ddot{x}^e + x^e = x_p^e. \tag{5}$$

In order to solve numerically time is discretized to $0, \cdots, t_m$ with time step Δt. Acceleration at each time $\ddot{x}^e(t_i)$ can be represented as follows;

$$\ddot{x}^e(t_i) = \frac{x^e(t_{i+1}) - 2x^e(t_i) + x^e(t_{i-1})}{\Delta t^2}. \tag{6}$$

Then Eq. (5) can be expressed as trinomial equations. Setting boundary conditions as $x^e(0)$, $x^e(t_m) = 0$, $x^e(i)$ ($i = 1$ to $t_m - 1$) are obtained.

These values are the modification distance of every sampling time. In order to reduce the effect of the approximation, calculated trajectory is set to initial trajectory, and this procedure is repeated.

4 Sensor Feedback Modification

The original role of the sensor feedback modification layer is to handle the disturbance caused by modelling error of the robot and environment. Our robot falls down in several steps because of the difference between real world and modelled world without sensor feedback. We added a function of the updating hand posture according to the desired hand position input in absolute coordinate system. Low latency hand position control is achieved by this direct input.

This layer changes the horizontal position of the torso to maintain the balance. Position of feet is not changed from the given trajectory in absolute coordinate system, and position of hands is set to the input. Inverse kinematics is solved both for legs and arms.

Position of the robot in absolute coordinate system is obtained by dead reckoning. Therefore posture of the arms is calculated by desired absolute position of hands and torso position based on dead reckoning. Compensating the error caused by the dead reckoning is the role of outer loop, such as, visual feedback and remote operation.

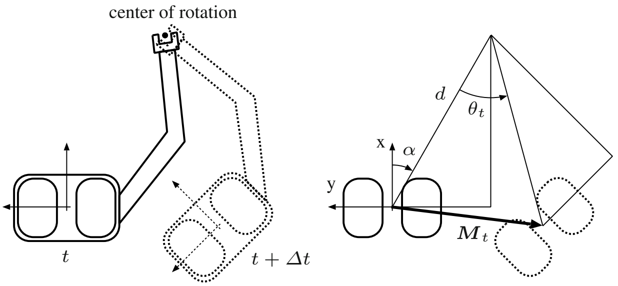


Fig. 4. Calculation of the torso trajectory.

5 Inverse Kinematics of Arms

We designed the robot arms to have 7 DOF for each, in order to extend the workspace. Pseudo jacobian is calculated to solve inverse kinematics of redundant system.

$$d\boldsymbol{\theta} = \boldsymbol{J}^{\#} d\boldsymbol{x} + (\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J})\boldsymbol{k} \tag{7}$$

$\boldsymbol{k}$ is chosen to make the joint which is the nearest to the angle limitation keep away from the limitation. However, solution falls to singularity point or joint angle limitation in some cases. Two emergency modes are prepared for such cases. One is to quit fixing hand position, and stop changing angles of the arm. This is used when the hand is in the air. The other is to cut off the power of motors in the arm. This option is used when the hand is grasping something connected to the environment.

6 Planning of Torso Trajectory

This system requires the inverse kinematics of the arm to be solved at all times. However, since it is hard to guarantee existence of the solutions of the inverse kinematics in advance, the following method is adopted. An initial arm posture is heuristically decided to be in the middle of the workspace and the robot tries to keep that configuration in average for the entire motion period.

The torso trajectory is determined so that the hand motion is canceled. A walking motion is generated to satisfy the desired torso motion vector $\boldsymbol{M_t}$ (horizontal translation $M_{t,x}, M_{t,y}$ and rotation around vertical axis θ_t) given at time t. Now suppose that the hand is rotating with respect to a point in space as shown in Fig. 4. In order for the torso to follow the hand by the time $t + \Delta t$, the torso motion vector $\boldsymbol{M_t}$ should be

$$\begin{aligned} M_{t,x} &= 2d \sin\frac{\theta_t}{2} \sin\left(\alpha + \frac{\theta_t}{2}\right), \\ M_{t,y} &= -2d \sin\frac{\theta_t}{2} \cos\left(\alpha + \frac{\theta_t}{2}\right), \end{aligned} \tag{8}$$

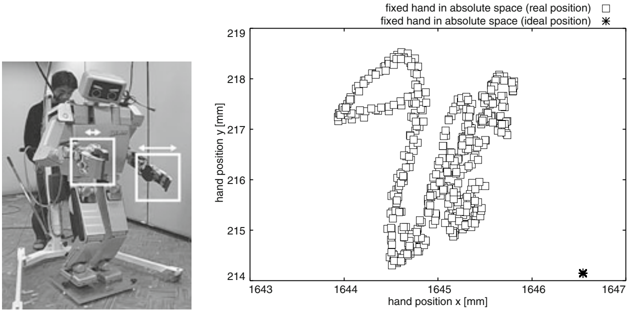


Fig. 5. Evaluation of the fixed hand while the robot is walking.

where d denotes the initial distance between the hand and the torso frame, and α denotes the initial rotational angle of the origin of the hand frame with respect to the torso frame. The third element of the vector $\boldsymbol{M_t}$ is equivalent to the rotational angle of the hand frame, which is θ_t. The horizontal translation of the hand motion should be equal to that of the torso motion. So the horizontal translation of the hand motion is simply added to $M_{t,x}$ and $M_{t,y}$ in Eq. (8). Therefore, if the initial parameters d and α are selected appropriately, the torso motion vector lets the body follow the hand motion. Then footprints are planned to realize the torso motion in average[11].

7 Experiments and Evaluation

The described functions are implemented, and online walking control system that realize desired hand trajectory in absolute coordinate system is constructed on humanoid “JSK-H7”.

7.1 Evaluation of the System

For the first experiment, desired position of right hand is fixed to a point in order to evaluate the accuracy of the system. The oscillation of the hand was about 20 to 30 [mm]. The error is caused by the low rigidity of the links and joint servo error. In order to estimate the amount of joint servo error, direct kinematics is solved by using the current angles measured by encoders attached to the motors. Fig. 5 (right) shows the calculated result. Position of the right hand fluctuates in the maximum width of about 5[mm], and the average positions is not even near the goal hand position. Bias of the position is considered to be caused by gravity force and friction of the system. And the effect of low rigidity is relatively larger than the servo error.

7.2 Remote Control Experiment

Remote control system with 3D mouse (Magellan, 3Dconnexion) is constructed. The input from the mouse is converted to the desired velocity of a hand in the global coordinate system. The control point can be set to arbitral position in the hand link frame. As shown in Fig. 6, the robot successfully pushed the chair by two hands by setting the control point at the middle of the two hands, and controlling the both hand with same value.

7.3 Sensor Feedback Experiment

In Fig. 7, a person leads the robot by guiding its hand. 6-axis force information at the hand is calculated based on the output of the 6-axis force sensor at the wrist. The effect of gravity is approximately canceled by subtracting initial sensor output. Desired hand velocity is decided proportional to the 6-axis force. It is comfortable for the leading person that the dynamics compensation motion of the torso does not transmitted to the paerson. Furthermore, as the robot's hand follows the guiding person's hand in advance of the body, the person doesn't disturb the torso motion for stable walking.

7.4 Visual Feedback Experiment

Reaching out for an object by watching it continuously becomes essential for a robust interaction between a hand and an environment. As the hand approaches the

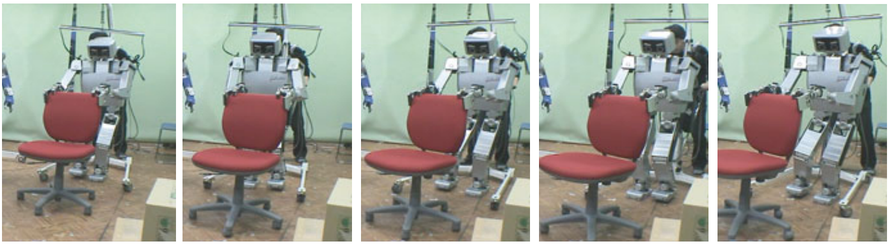


Fig. 6. Pushing a chair by remote control.

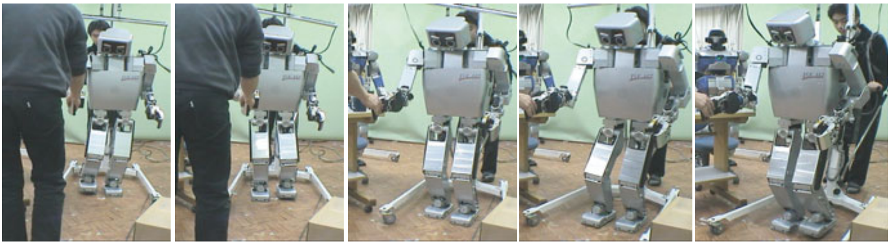


Fig. 7. Following a person using 6-axis force sensor at the wrist.

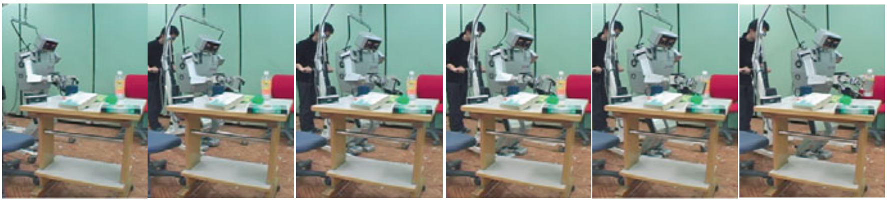


Fig. 8. Reaching while walking using visual feedback.

target object by visual servo, the robot grasps the object if it is settled in a certain distance from the hand.

Color markers are put on the hand and the target object. The 3D-position of the marker is calculated by means of trigonometrical measurement using the images of stereo camera on the head. Even though they have exactly the same color, they are distinguishable from each other using the arm posture. The 2D-position of the hand marker in the image coordinate system is estimated from the geometric information of the body. The robot judges whether the target object is in its hand or not by looking at a constant point near the visible hand marker. Because the robot observes the relative positions of the hand and the target, the gap between the actual and the ideal positions of the world frame does not have to be considered. The world frame moves as the robot's feet slip on the ground.

8 Conclusion

The method of constructing walking system that accepts trajectory of desired hand position and automatically generates and executes the walking motion is presented. The system is realized by designing and implementing three modules, torso transition planner, walking pattern generator, and sensor feedback balance compensator, that run cooperatively. We thought the system is indispensable for remote operation and sensor feedback tasks involved with hand position control. Some experiments of remote control and sensor feedback are shown that the system works well for such kinds of tasks.

For future work, we should develop more robust inverse kinematics to realize hand position, as most of the experimental failures arise from failing in solving inverse kinematics to realize the desired hand position during walking. We think using more DOFs than DOFs in the arm is important to make better inverse kinematics module. Giving up controlling some DOFs that is not important for a task also makes more robust inverse kinematics. Improving torso transition planner to use some prediction to decide the next step is another future work.

References

1. Kazuo Hirai. Current and future perspective of Honda humanoid robot. In *Proc. of 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97)*, pages 500–508, 1997.
2. Jin'ichi Yamaguchi, Sadatoshi Inoue, Daisuke Nishino, and Atsuo Takanishi. Development of a bipedal humanoid robot having antagonistic driven joints and three dof trunk. In *Proc. of the 1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 96–101, 1998.
3. Ken'ichirou Nagasaka, Masayuki Inaba, and Hirochika Inoue. Walking pattern generation for a humanoid robot based on optimal gradient method. In *Proc. of 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics*, 1999.

4. Kazuhito Yokoi, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Shuji Kajita, and Hirohisa Hirukawa. A honda humanoid robot controlled by aist software. In *Proc. of the IEEE-RAS International Conference on Humanoid Robots*, pages 259–264, 2001.
5. Yoshiharu Kaneshima, Yusuke Sugahara, Shumpei Ando, Motohiro Sato, Hun ok Lim, and Atsuo Takanishi. Quasi real-time motion pattern generation for bipedal humanoid robots. In *Proceedings of 19th Annual Conference on Robotics Society of Japan*, pages 987–988, 2001.
6. Koichi Nishiwaki, Satoshi Kagami, Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*, pages 2684–2689, 2002.
7. Kenji Inoue, Yusuke Nishihama, Tatsuo Arai, and Yasushi Mae. Mobile manipulation of humanoid robots –body and leg control for dual arm manipulation–. In *Proc. of International Conference on Robotics and Automation (ICRA'02)*, pages 2259–2264, 2002.
8. Neo Ee Sian, Kazuhito Yokoi, Shuuji Kajita, and Kazuo Tanie. Whole body teleoperation of a humanoid robot integrating operator's intention and robot's autonomy – an experimental verification –. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*, pages 1651–1656, 2003.
9. K. Yokoyama et al. Cooperative works by a human and humanoid (in japanese). In *Proc. of The 21th Annual Conference of the Robotics Society of Japan*, page 3A35, 2003.
10. S. Kagami, K. Nishwiaki, T. Kitagawa, T. Sugihara, M. Inaba, and H. Inoue. A fast generation method of a dynamically stable humanoid robot trajectory with enhanced zmp constraint. In *Proc. of IEEE International Conference on Humanoid Robotics (Humanoid2000)*, 2000.
11. Koichi Nishiwaki, Satoshi Kagami, James J. Kuffner, Masayuki Inaba, and Hirochika Inoue. Online humanoid walking control system and a moving goal tracking experimant. In *IEEE International Conference on Robotics and Automation (ICRA2003)*, pages 911–916, 2003.

Body Information Acquisition System of Redundant Musculo-Skeletal Humanoid

Ikuo Mizuuchi, Yuto Nakanishi, Tomoaki Yoshikai, Masayuki Inaba, and Hirochika Inoue

Dept. of Mechano-Informatics, The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656 Japan
{ikuo,nakanish,yoshikai,inaba,inoue}@jsk.t.u-tokyo.ac.jp
http://www.jsk.t.u-tokyo.ac.jp

Abstract. We propose a design of musculo-skeletal humanoid whose muscles can be easily added, removed, or re-arranged, which can be compared to the ability of growing up of a human's (or an animal's) body. This paper presents a design and development of the reconfigurable complex-body robot, and a method to build a system which can acquire the body information and can be used for realization of motions of human-like nonlinear musculo-skeletal robots. It shows that even reconfigurable complex-body robot's motion can be achieved by a feedback modification method in conjunction with neural network. We did basic validation experiments by giving target postures one by one, measuring the realized postures, and comparing them with the target postures. We also carried out whole-body motion experiments using motion captured sequences using the presented system.

1 Introduction

When robots come into people's home, they will have to do numerous kinds of diverse tasks and be absolutely safe. To meet these requirements, one of the directions of progress is thought to be flexible and complex body robots, which have large-DOF and many sensors like humans. Humanoids that have mechanical flexibility and redundancy[1–5] have a possibility to solve the problem, but a generic method to manage the complex robot's body will have to be established. That is a target of our research. From this point of view, we have developed several kinds of flexible humanoid robots[6,7] and been proceeding to try to solve the problem. It is difficult to build an analytical model of a human-like musculo-skeletal system[8,4,9] effective to the real system, because the relation between actuator command and resulting state is nonlinear. There are interference between muscles and between muscles and the body as well as friction. Lots of works about learning of humanoid (e.g. [10–13]) have been about acquisition of information on structure of motions or tasks such as motion patterns, sensory-motor coordination, or interaction between environment. The humanoid presented in this paper, on the other hand, does not even have information on the relationship between an actuator state and the resulting posture. Capturing the information can be compared to acquiring a body image. Asada et al. propose a design principle of humanoid robots[14] and show a possibility and a direction of humanoid robots equipped with various sensors and actuators.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 249–258, 2006.

Yoshikawa et al. showed a method to obtain self-body information from multi-modal sensory information[15].

2 Design of Reconfigurable Musculo-Skeletal Humanoid

We have developed a redundant muscle-driven endoskeletal robot leg (Fig.9, Fig.10) which can replace the lower body of the previously developed whole-body flexible-spine humanoid[5], to tackle with the problem of highly nonlinear complex body management. Each of the muscles consists of an electric motor, a gear unit, an encoder, steel wires, pulleys, and a tension sensor. Each motor and sensor is a united package and is placed in the middle of a muscle. The both ends of each muscle are the terminals of wire, which can be fixed to any two parts of the body. There are a lot of wire-fixable points at the parts of the body, and we can easily add/remove muscles or change the arrangement of them. The automatic acquisition system will adapt to the change of the configurations of muscles.

2.1 Rearrangeability of Actuators

As for joint-motor systems and exoskeletal structure of normal humanoid robots(e.g.[16–18]), it is not easy to rearrange (change, add, or remove) the actuators. In joint-motor systems, in which each joint has one DOF and actuated by one motor, adding or removing a DOF (joint and motor) needs re-design and re-producing of mechanical structure again. In a rigid exoskeletal robot, it is difficult to replace a motor whose size or shape is different. Endoskeletal structure actuated by parallel muscle-tendon systems can be a solution to the problem. Ball-and-socket joints which have three degrees of freedom would help the reconfigurability, for power property (max torque, max speed, etc) of this kind of joint can be modifiable only by changing the arrangement of muscles which is actuating the joint. It can also be said that a ball-and-socket joint realizes a simple and compact mechanical structure of three degrees of freedom, compared with serial three rotational joints. Endoskeletal structure provides the redundant spaces for muscle rearrangement; a region where muscles are reinforced would become thicker both in human's case and in humanoid's case. The system allow to have also multi-articular muscles and cooperative muscles[19] like human and animal.

Below is a comparison between endoskeletal structure and exoskeletal structure.

- Endoskeletal structure is:
  - easy to increase/decrease muscles, easy to modify attached positions of muscles
  - possible of human-like muscle-arrangements
  - easy to modify the shape of cover or skin
  - difficult to implement cover or skin
- Exoskeletal structure is:
  - easy to attach skin function (e.g. tactile sensor)
  - difficult to re-arrange, add, or remove muscles
  - different from human structure
  - difficult to change the shape design

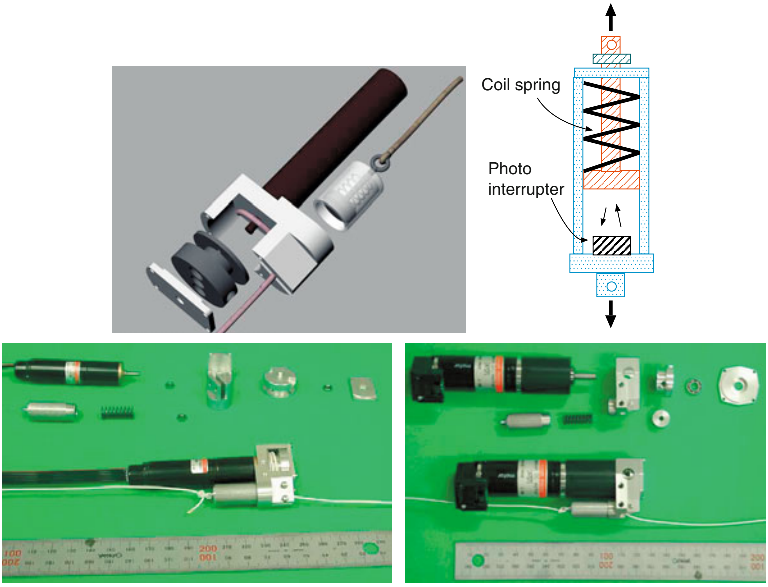


Fig. 1. Muscle unit (upper-left: muscle-unit structure, upper-right: tension-sensor structure, lower-left: muscle-unit for 4.5W motor, lower-right: muscle-unit for 20W motor)

2.2 Design of Muscle Unit

To implement the rearrangeable muscle humanoid, we have designed and developed muscle units (Fig.1). The upper figure of Fig.1 shows the mechanical structure of the muscle unit. A unit consists of a DC motor with gearhead and a rotary encoder, a tension sensor, two pulleys, and a casing. Two wires are connected to each unit; one wire is connected to the tension sensor and the other is wound up by the pulley which is fixed to the motor's spindle. The other pulley, which is in the slit of the casing and is not seen in the upper left figure of Fig.1, changes the direction of the wire. The lower figures of Fig.1 show two types of the developed muscle units using MAXSON's 4.5W and 20W DC motors. The mechanism of the tension sensor is seen in the upper right figure of Fig.1. The length of the coil spring in the cylinder changes according to the muscles' tension. By measuring the distance between the piston and the bottom of the cylinder, the tension is detected. The distance is measured by a photo interrupter which uses an infrared LED and an infrared photo-transistor.

In conventional muscle-driven systems, the motor and pulley for pulling a muscle was usually placed at the end of the wire. In our design the motor is placed at the middle of the muscle wire. Therefore the fixing positions of the both ends of a muscle can be freely decided and also easily modified. Thus easy rearrangeability of muscles is realized.

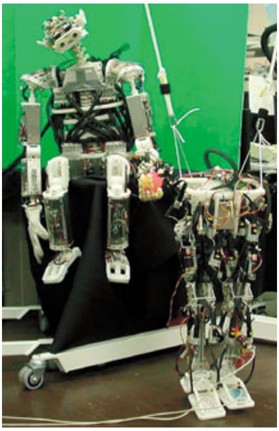


Fig. 2. Kenta and its new leg

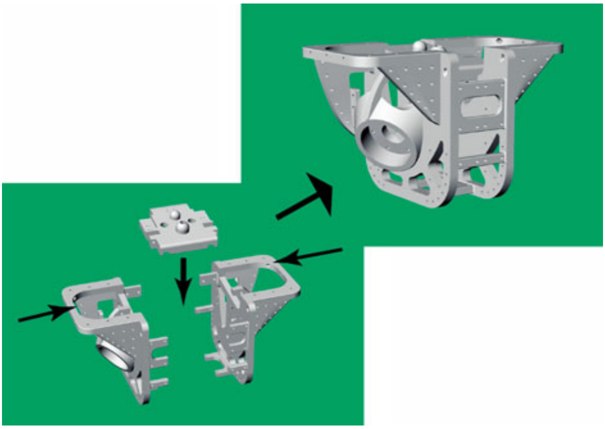


Fig. 3. Divided structure of the pelvis

2.3 Design of Redundant Musculo-Skeletal Lower Body of Humanoid

A new human-like muscle-driven endoskeletal lower body of humanoid has been designed and developed. This can replace the muscle-driven flexible-spine humanoid "Kenta" which we previously developed[7,5] (Fig.2). Kenta's arms and legs are exoskeletal structure while Kenta's vertebral torso and neck are endoskeletal structure. In this paper, we adopted the endoskeletal structure, considering the human-like characteristics such as coupled drive mechanism[19]. The structure of human's body gives useful information and good inspiration to the design of the skeleton and joints. Human's joints are held and bound by numerous muscles and strong tendons and ligaments. It is not realistic to try to reproduce the same structure and mechanism. Instead of precisely re-building human's skeletal structure, the design has been done referring to the good points of human's structure and mechanisms, and at the same time the design incorporated some originally designed structure as well.

Around crotch joint The pelvis (Fig.3) has a role of the base of spine, and is connected to the upper body through the spine and to the upper thigh through the crotch joints. The designed pelvis consists of three parts like human's as seen in Fig.3 (3D CAD model); left and right hip bone and sacrum. The motors which actuate the muscles of the spine will be placed on the upper surface of the hip bones. Between the hip bones, the motor driver and controller circuits for the legs are stored. The 3D complex shapes were formed by SLS(selective laser sintering) method and the material is a kind of polyamide resin. In order to enable to increase/decrease muscles and to modify the attaching points of muscles, there are many holes in the pelvis.

Each crotch joint has three degrees of freedom by a ball-and-socket joint. The articular surface inclines forward and outward like human's crotch joint. The range of movement is: $-15°$(backward) to $+120°$(forward) in pitch rotation, $\pm45°$ in roll rotation, and $\pm30°$ in yaw rotation, though the range around each axis is greatly

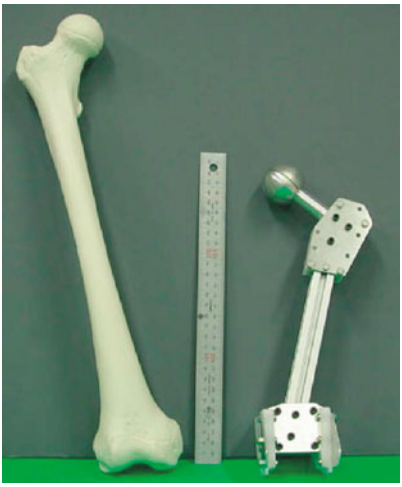


Fig. 4. A human's thigh bone and the developed thigh bone

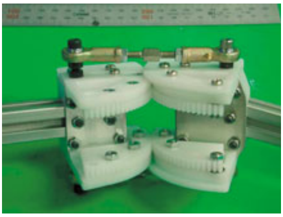


Fig. 5. two-axis rolling contact joint of knee; distance of the two axes is fixed and the gear surface avoids slipping.

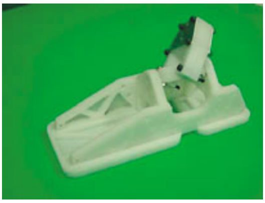


Fig. 6. The ankle and foot

dependent on the posture of the joint. The neck of thigh bone and the ball of crotch joint are made of extra super duralumin. The friction between the materials of the ball (duralumin) and socket (polyamide resin) is quite small.

Each thigh bone is also like human's shape; it curves sharply near the neck of thigh bone and the main axis is not parallel to the line between the crotch joint and ground contact point of foot (Fig.4). The advantages of this shape can be thought as follows: (1) well matching to the inclination of hip joint's face, (2) avoiding the collision with the other leg, (3) the wide range of movement, and (4) the long moment arm when a muscle actuates a joint. A disadvantage is the concentrated stress at the neck of thigh bone and sharp curving area. Fig.4 shows a human's thigh bone and the developed thigh bone. The main axis of each the part is an extruded aluminium-alloy frame, to which various parts such as muscle attaching parts, motor driver circuit boards, coverings, etc. can be attached to or re-arranged on the aluminium-alloy frames using bolts without any additional process like cutting or drilling.

Knee joint Each knee joint is a one-DOF joint that has two axis with rolling contact. The distance between the two axes is fixed by turnbuckles at the both sides, and the contact surface is a gear to avoid slipping. Fig.5 shows the knee joint. One of the advantages of the type is the large range movable range. Without taking large offset of the joint axis or the thigh bones, the knee can bend in 160°. The minimum angle of the developed knee joint is not 0° but $-4°$ for having stability when standing. As another feature of the two-axis rolling joint, the center of rotation moves and the moment arm changes as the joint bends. As some other advantages, the joint moves smoothly even when a large force is put at the joint, and it will be possible to have the shock absorbing function by inserting a soft element between the contacting faces.

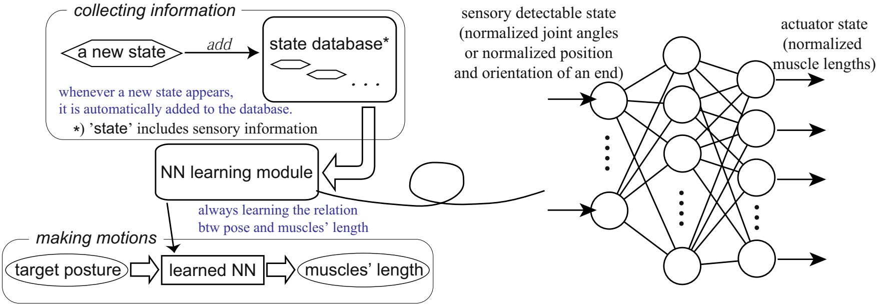


Fig. 7. The body information acquisition system

Ankle joint The developed ankle joint consists of orthogonal two axes (roll and pitch), a universal joint. Though human's ankle is a saddle joint with sliding contact, this kind of joint is affected by friction and we suppose it is difficult to adopt a saddle joint as the ankle at the current stage. Fig.6 shows the developed foot. Human's foot is a sophisticated structure which can absorb shock while walking with the arch and have quite many degrees of freedom, and there's much to learn from it. There is a study on toe joint of a whole body humanoid and using it for walking[20]. In this paper, the developed foot does not have any degrees of freedom, but we are planning to design some degrees of freedom at the foot.

3 Body Information Acquisition System

In order to realize a system which can keep the information on the nonlinear relation of sensor and actuator and always update the information, we have adopted neural network. The system keeps a database of the sensor-actuator state, and a three layered network is always learning by back-propagation. When the distance between an instant sensor-actuator state and every element in the database is larger than a threshold, a new element created by the state is added to the database. All the element in the database are used as supervisor data and learning process is always running. The elements of less frequency of appearance become less used as supervisor data. The system can be applied to the developed muscle-driven robot in which the arrangement of muscles is easily modified. Fig.7 shows the developed system.

4 Feedback Modification Method

When the robot moves using the system, a real sensory state might differ from the target state. As a solution to the case it occurs, we have tried a method using the relation of differential information. In order to decrease the error observed in the sensory state space (e.g. joint angle space), the amount of modification in the actuator space (e.g. muscle length space) is decided from the current and target sensory states ($\boldsymbol{\theta}$ and $\boldsymbol{\theta}_r$).

$$\boldsymbol{L}_{mod} = f(k(\boldsymbol{L} - \boldsymbol{L}_r)) \tag{1}$$

$\boldsymbol{L}_{mod}$ is the quantity of modification in the actuator state space. $\boldsymbol{L}$ and $\boldsymbol{L}_r$ are calculated from the trained neural network, which is described in Section **3** and Fig.7. The neural network inputs the current and target sensory states ($\boldsymbol{\theta}$ and $\boldsymbol{\theta}_r$), and outputs the corresponding actuator states ($\boldsymbol{L}$ and $\boldsymbol{L}_r$). $\boldsymbol{\theta}$ is from detected current sensory information and while $\boldsymbol{\theta}_r$ is provided by a motion pattern provider (a motion pattern generator). k is a fixed gain and decided through experiments. f is a limiter function which scales a given vector so that absolute value of every element of the vector should not exceed a limit.

This procedure means calculating the Jacobian matrix; the differential value in the actuator state space ($\boldsymbol{L} - \boldsymbol{L}_r$ corresponds to $\delta\boldsymbol{L}$) is calculated from the differential value in the sensory state space ($\boldsymbol{\theta} - \boldsymbol{\theta}_r$ corresponds to $\delta\boldsymbol{\theta}$), using the neural network; Jacobian matrix $\boldsymbol{J}$ between sensory and actuator state spaces is represented as:

$$\dot{\boldsymbol{L}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} \tag{2}$$

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial l_1}{\partial \theta_1} & \cdots & \frac{\partial l_1}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial l_m}{\partial \theta_1} & \cdots & \frac{\partial l_m}{\partial \theta_n} \end{bmatrix} \tag{3}$$

θ_1 to θ_n are the element of $\boldsymbol{\theta}$, and l_1 to l_m are the elements of $\boldsymbol{L}$.

In the experiments described in Section **5** as an example, muscle lengths (corresponding to $\boldsymbol{L}$) are modified so as to decrease the difference of the target and realized joint-angles (corresponding to $\boldsymbol{\theta} - \boldsymbol{\theta}_r$). $\boldsymbol{L}_{mod}$ is the quantity of modification of servo-reference of muscle lengths. Although the developed robot does not have any joint-angle sensors, we detect the joint-angles using a three dimensional position/orientation measuring device called FASTRAK produced by Polhemus.[21].

5 Experiments

5.1 Posture Control

Based on the proposed approach, a software system which contains the relation between posture (joint-angles) and muscle-lengths was developed (Section **3**). Using the system, the muscle-driven robot (Section **2**) was able to realize commanded postures. Fig.8 shows the relation between target and realized postures without the feedback control using the Jacobian matrix. The average error was about 4 degrees and the maximum error was about 5 to 13 degrees as shown in the figure. By the feedback control using Jacobian matrix, the errors were decreased but tensions of muscles became unnecessarily large because of the inner antagonistic tensions. Detecting the extending muscles and making them extend more may help to decrease the unnecessary power consumption.

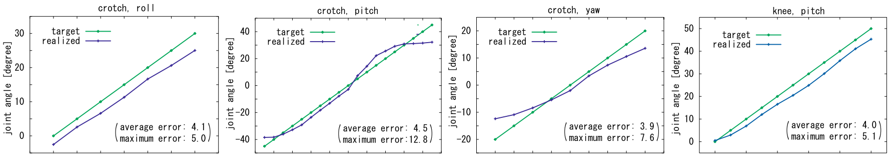


Fig. 8. Target joint-angle and realized joint-angle using the trained NN

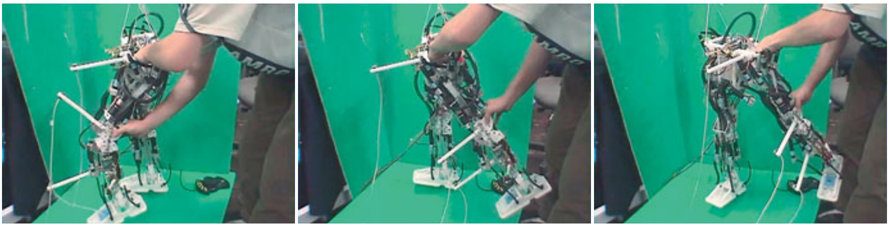


Fig. 9. Remembering sensor-actuator relationships

Fig.9 shows the snapshots in which a human directly moves the robot-leg. This is a teaching phase and tensions of muscles are being controlled to keep a certain tension[6,7,4]. Even when not in the teaching phase, the supervisor data is being updated at any time.

5.2 From Motion-Captured Data of Human

For demonstrating the effectiveness of the approach, experiments of imitating human's motions have been carried out. Using the system, we tried to make the robot to play some motion-captured sequences. Each frame of the motion-captured sequence has information on position and orientation of the parts. By mapping each joint-angle calculated from the frame's information to the robot and by controlling by the system, the motion-captured sequence are reproduced. Fig.10 shows the snapshots of a motion from a motion captured walking sequence, and Fig.11 shows a motion of stepping over something also from a motion captured sequence.

6 Concluding Remarks

This paper has presented a method of capturing the body information of human-like musculo-skeletal humanoid system as well as a design of reconfigurable-muscle humanoid. It allows to perform motions of such kind of robots. Using a neural network, a complex-body motion can be achieved in association with a feedback modification method. It might be hard to apply the presented method to subjects which need precise and high speed control such as two-leg balancing during walking, because in general the errors of neural network cannot be got rid of and feedback

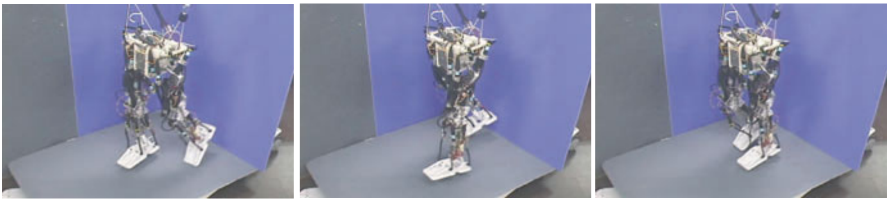


Fig. 10. A walking motion from motion-captured sequence

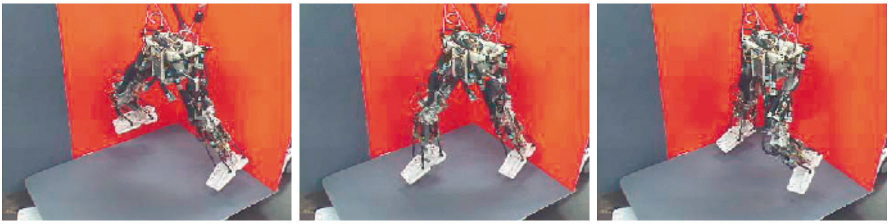


Fig. 11. A stepping over motion from motion-captured sequence

modification needs some time. A feedforward control method may be able to solve the feedback problem. From another point of view, the presented method can manage the integration of various sorts of sensors, and we can order the robot to act by indicating a target sensory state. We suppose complex-body humanoids need a paradigm shift of design and control which has been focusing on rigidity and accuracy. Design and control of a human does not seem to be always rigid and accurate.

References

1. T. Morita and S. Sugano, "Development and evaluation of seven-d.o.f. mia arm," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 462–467, 1997.
2. R. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati, and M. Williamson, "The Cog Project: Building a Humanoid Robot," in *Computation for Metaphors, Analogy and Agents* (C. Nehaniv, ed.), vol. 1562 of *Springer Lecture Notes in Artificial Intelligence*, pp. 52–87, Springer-Verlag, 1998.
3. Shadow Robot Company, "The shadow biped." http://www.shadow.org.uk/projects/biped.shtml.
4. I. Mizuuchi, S. Yoshida, M. Inaba, and H. Inoue, "The development and control of the flexible-spine of a human-form robot," *Advanced Robotics*, vol. 17, no. 2, pp. 179–196, 2003.
5. M. Inaba, I. Mizuuchi, R. Tajima, T. Yoshikai, D. Sato, K. Nagashima, and H. Inoue, "Building spined muscle-tendon humanoid," in *Robotics Research: The Tenth International Symposium*, pp. 113–130, Springer Verlag, 2003.
6. I. Mizuuchi, M. Inaba, and H. Inoue, "A Flexible Spine Human-Form Robot — Development and Control of the Posture of the Spine —," in *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2099–2104, 2001.

7. I. Mizuuchi, R. Tajima, T. Yoshikai, D. Sato, K. Nagashima, M. Inaba, Y. Kuniyoshi, and H. Inoue, "The Design and Control of the Flexible Spine of a Fully Tendon-Driven Humanoid "Kenta"," in *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2527–2532, 2002.
8. T. Komura, Y. Shinagawa, and T. L. Kunii, "Calculation and visualization of the dynamic ability of the human body," *The Journal of Visualization and Computer Animation*, vol. 10, no. 2, pp. 57–78, 1999.
9. Y. Nakamura, K. Yamane, I. Suzuki, and Y. Fujita, "Dynamic computation of musculoskeletal human model based on efficient algorithm for closed kinematic chains," in *Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines*, pp. SaP–I–2, 2003.
10. C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)* (D. F. Jr., ed.), pp. 12–20, Morgan Kaufmann, 1997.
11. K. Nagasaka, A. Konno, M. Inaba, and H. Inoue, "Acquisition of Visually Guided Swing Motion Based on Genetic Algorithms and Neural Networks in Two-Armed Bipedal Robot," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pp. 2944–2949, 1997.
12. F. Kanehiro, M. Inaba, and H. Inoue, "Action acquisition framework for humanoid robots based on kinematics and dynamics adaptation," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pp. 1038–1043, 1999.
13. I. Mizuuchi, T. Yoshikai, D. Sato, S. Yoshida, M. Inaba, and H. Inoue, "Behavior developing environment for the large-dof muscle-driven humanoid equipped with numerous sensors," in *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, pp. 1940–1945, 2003.
14. M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robotics and Autonomous Systems*, vol. 37, pp. 185–193, 2001.
15. Y. Yoshikawa, H. Kawanishi, M. Asada, and K. Hosoda, "Body scheme acquisition by cross modal map learning among tactile, visual, and proprioceptive spaces," in *Proceedings of the 2nd International Workshop on Epigenetic Robotics*, pp. 181–184, 2002.
16. K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The Development of Honda Humanoid Robot," in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pp. 1321–1326, 1998.
17. K. Nishiwaki, T. Sugihara, S. Kagami, F. Kanehiro, M. Inaba, and H. Inoue, "Design and Development of Research Platform for Perception-Action Integration in Humanoid Robot : H6," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, vol. 3, pp. 1559–1564, 2000.
18. K. Kaneko, F. Kanehiro, S. Kajita, K. Yokoyama, K. Akachi, T. Kawasaki, S. Ota, and T. Isozumi, "Design of prototype humanoid robotics platform for hrp," in *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Lausanne, Switzerland), pp. 2431–2436, October 2002.
19. S. Hirose and M. Sato, "Coupled drive of the multi-DOF robot," in *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pp. 1610–1616, 1989.
20. K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Toe joints that enhance bipedal and fullbody motion of humanoid robots," in *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, pp. 3105–3110, 2002.
21. Polhemus, Inc., "Fastrak: The fast and easy digital tracker."
http://www.polhemus.com/fastrak.htm.

Coordinated Task Execution for Humanoid Robots

T. Asfour, D.N. Ly, K. Regenstein, and R. Dillmann

Forschungszentrum Informatik Karlsruhe (FZI)
Haid-und-Neu-Str. 10-14, D-76131 Karlsruhe, Germany
http://www.fzi.de
Email: asfour@ira.uka.de, ly@fzi.de, regenstein@fzi.de, dillmann@ira.uka.de

Abstract. This paper presents a framework for the coordinated execution of tasks in robotic systems with a high degree of freedom such as humanoid robots. Focusing on tasks to be executed by different subsystems of the robot (e.g. mobile platform, two redundant arms, a head with vision and acoustic system), a motion coordination scheme is presented. The coordination scheme is based on the synchronization of the motion of each subsystem while performing a common task. The validity of the proposed coordination scheme is experimentally demonstrated by different tasks of the humanoid robot e.g. two-arm tasks, head-arm tasks or platform-arm tasks.

1 Introduction

Our current research interest is the development of a control architecture to achieve manipulation task goals for a humanoid robot. In particular, we address the programming and coordinated execution of manipulation tasks in a household environment. Therefore, it is an important issue to coordinate the multiple subsystems of a humanoid robot in carrying out tasks in dynamic unstructured environments. The control architecture must provide the possibility to integrate the motor, perception and cognition components necessary for natural multimodal human-humanoid interaction and human-humanoid cooperation. In the literature, considerable research effort has been focused on various problems related to motion coordination of robot systems. So far,many coordination schemes for multiple arm systems have been reported: the master/slave control [7], the centralized control [14] and the decentralized control ([8], [6]). To design control architectures for humanoid robots, some approaches have been suggested: subsumption architecture [4] and task-oriented approaches [10]. This paper is organized as follows. Section 2 briefly describes the humanoid robot ARMAR. The control architecture for coordinated task execution is introduced in section 3. The coordination strategy using Petri nets is given in Section 4. Section 5 presents the implementation and the experimental results of coordinated execution of different tasks.

2 System Configuration – The Humanoid Robot ARMAR

The humanoid robot ARMAR [3] has 23 mechanical degrees-of-freedom (DOF). From the kinematics control point of view, the robot consists of five subsystems:

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 259–277, 2006.

Head, left arm, right arm, torso and a mobile platform. The upper body of ARMAR has been designed to be modular and light-weight while retaining similar size and proportion as an average person. The control system of the robot is divided into separate modules. Each arm as well as torso, head and mobile platform having its own software- and hardware controller module. The head has 2 DOFs arranged as pan and tilt and is equipped with a stereo camera system and a stereo microphone system. Each of the arms has 7 DOFs and is equipped with 6 DOFs force torque sensors (ATI, [15]) on the wrist. The current mobile platform of ARMAR consists of a differential wheel pair and two passive supporting wheels. It is equipped with front and rear laser scanner (Sick, [17]). Furthermore, it hosts the power supply and the main part of the computer network.

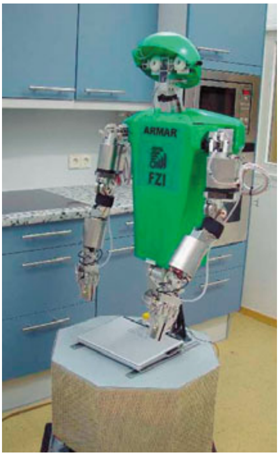


Fig. 1. The humanoid robot ARMAR with five subsystems: head, two arms, torso and mobile platform.

3 Control Architecture for Humanoid Robots

In this section, we introduce our control architecture. First, we summarize the design criteria. Second, we introduce the proposed and hierarchically organized control architecture. The control architecture was designed according to the following global criteria:

- Flexibility and modularity to cope with various tasks and to allow the addition of further tasks and hardware and software modules in a simple manner. This is a very important feature for the process of integration.
- Real-time performance to allow a prompt response to varying environments and exceptions which can occur during the task execution.

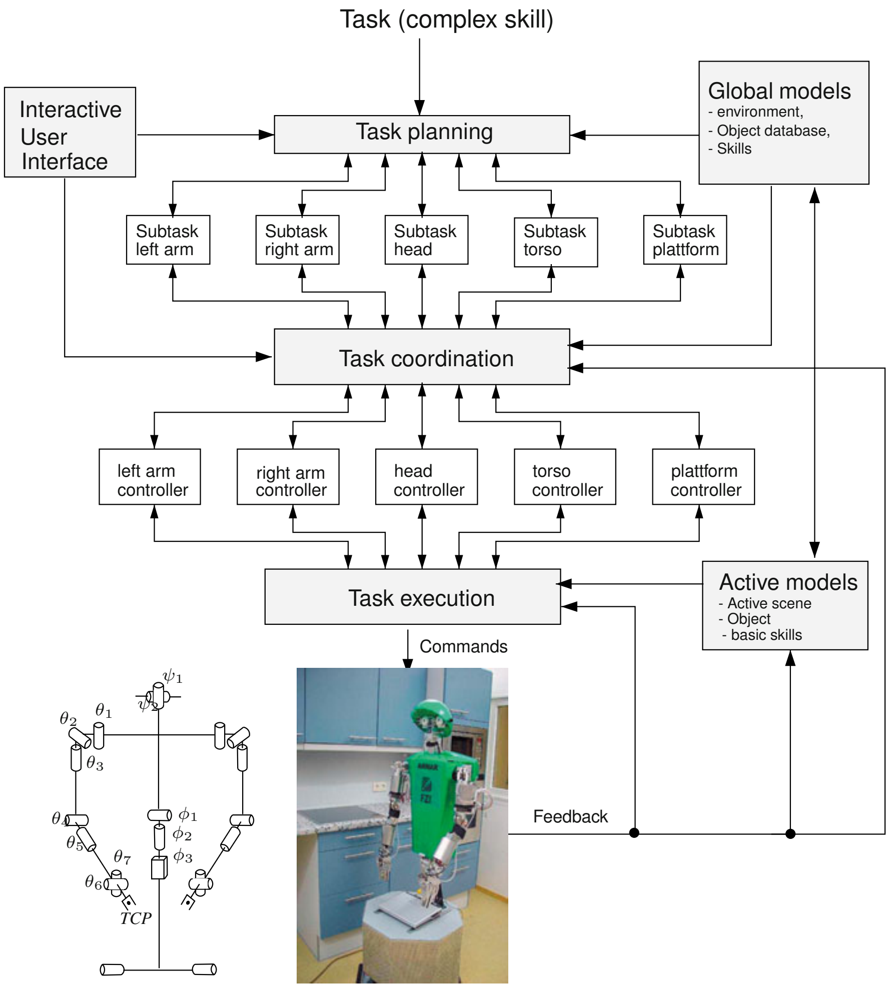


Fig. 2. Hierarchical control architecture for coordinated task execution in humanoid robots: planning, coordination and execution level.

According to the definition of intelligent machines given by Sardis ([12], [13]) we decomposed the overall control system of the humanoid Robot ARMAR into three levels as shown in figure 2. A given task is decomposed into several subtasks, representing the sequence of actions the subsystems of the humanoid robot must carry out to accomplish the task goal. The coordinated execution of a task requires the scheduling of the subtasks and their synchronization with logical conditions, external and internal events. Figure 2 shows the block diagram of the control architecture with three levels:

- The task planning level specifies the subtasks for the multiple subsystems of the robot. This level represents the highest level with functions of task representation. It generates the subtasks for the different subsystems of the robot autonomously or interactively by a human operator. The generated subtasks for the lower level contain the whole information necessary for the task execution, e.g., parameters of objects to be manipulated in the task or the 3D information about the environment. According to the task description, the subsystems controllers are selected here and activated to achieve the given task goal.
- The task coordination level generates sequential/parallel primitive actions for the execution level in order to achieve the given task goal. The subtasks are provided by the task planning level. Like on the planning level the execution of the subtasks in an appropriate schedule can be modified/reorganized by an operator using an interactive user interface.
- The task execution level is characterized by control theory to execute specified sensory-motor control commands. This level uses task specific local models of the environment and objects. In the following we refer to those models as *active models*.

The active models are first initialized by the global models and can be modified and enhanced during the progress of the task execution. Internal system events and execution errors are detected from local sensor data. These events/errors are used as feedback for the task coordination level in order to take appropriate measures. For example, a new alternative execution plan can be generated to react to internal events of the robot subsystems or to environmental stimuli.

4 Coordination Strategy for Task Execution

The execution of coordinated tasks demands a mechanism for synchronization of actions allowing a deterministic switch between sequential/parallel actions of the robot. Therefore, a framework for coordinated execution of tasks using condition/event Petri nets was developed. Among the existing models of discrete event systems, Petri nets have been widely used to model dynamic systems [5]. In our work, we use Petri nets to efficiently represent both control and data flow within one formalism.

4.1 Petri nets

Petri nets have been widely used in both theoretic works to model dynamic systems and applications, especially in the modeling of manufacturing processes Petri nets are a graphical and mathematical formalism for modeling, simulation and formal analysis of discrete event systems [5], [1]. Petri nets allow the representation of both control and data flow within one formalism. There are many Petri net variants and definitions, which are extentions of so-called condition/event nets:

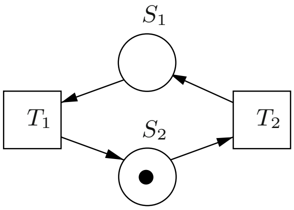
Definition (Condition/event net) A condition/event Petri net is defined by the 4-tuple $N = (P, T, A, m_0)$, where

- $P = \{p_1, \ldots, p_{n_p}\}$ is a finite set of places,
- $T = \{t_1, \ldots, t_{n_t}\}$ is a finite set of transitions,
- $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$
- $A \subseteq (P \times T) \cup (T \times P)$ is a set of arcs,
- $m_0 : P \rightarrow \{0, 1\}$ is the initial marking. It defines the initial number of undistinguishable tokens on each place $p \in P$.
- **Enabling rule:** A transition $t_j \in T$ is enabled if all input places of t_j contain a token and all output places are empty.
- **Firing rule** An enabled transition may fire. On firing it removes the tokens from all its input places and places one token in each of its output places.

The set of places describes the states of the system, and the set of transitions defines events that can change the state of the system. The state of a Petri net is represented by its markings, i.e., the distribution of tokens among places. In the usual graphical representation of Petri net graphs, places are depicted as circles and transitions as rectangles. The marking, that is, the distribution of tokens on places, represents the state of a Petri net model. Transition firings change the token distribution and thus the state of the system. They may reflect the occurrence of events or the execution of an operation.

4.2 Coordination of Task Execution

Figure 3 represents a Petri net for modeling one subsystem with the associated places and transitions. The shown initial marking indicates the state *ready* of the arm. The task execution can be invoked by firing the transition T_2 which leads to the state *active*.



Conditions (places)	
S_1	Subsystem is active
S_2	Subsystem is ready

Events (Transitions)	
T_1	Task execution is completed
T_2	Task execution is not completed

Fig. 3. Petri net for modeling of one subsystem (left) with its conditions and events (right).

The Petri net in figure 4 results from the synchronous composition of three nets for the coordinated execution of three tasks through the three different subsystems A, B and C. The descriptions of the transitions and events including the pre- and postconditions are given in table 1 and in table 2. The coordination of different subsystems motion takes place through the common transitions 7, 8, 9 and 10. Firing

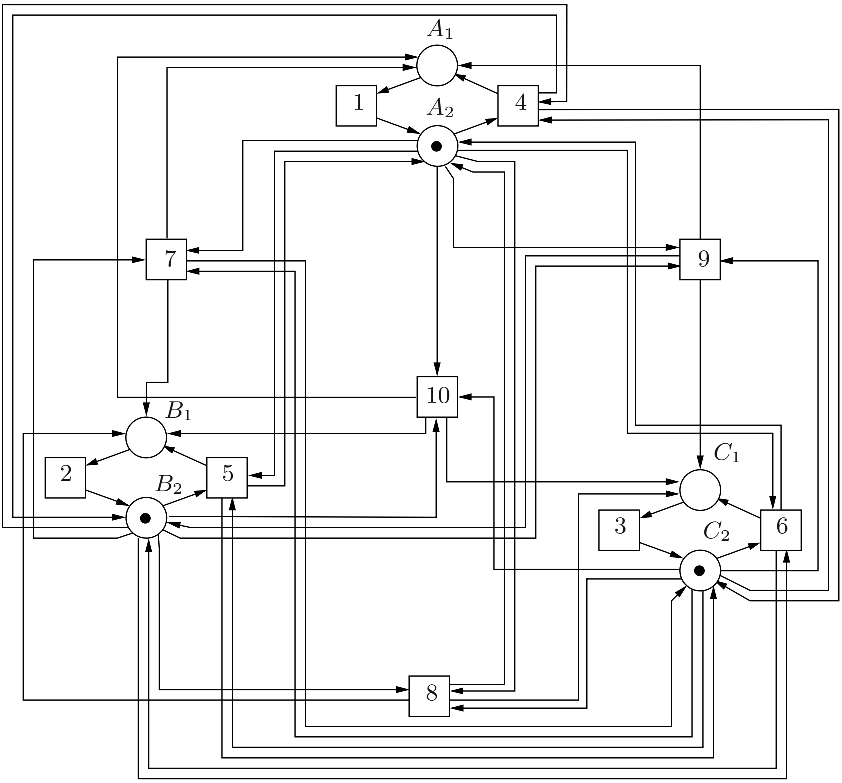


Fig. 4. Petri Net for coordinated task execution of three subsystems A, B and C

the transition 7 denotes respectively the task execution through the subsystems A and B, under the condition that both subsystems are ready. Firing of transition 10 denotes the simultaneous and parallel execution of tasks through the subsystems A, B and C.

5 Implementation and Experiments

The different coordination nets for multiple cooperative tasks of the robot subsystems are modeled and implemented in the software control framework MCA ([16]). It provides a standardized module concept with unified interfaces and allows an easy building of control groups with more complex functionalities from basic modules. Furthermore, it provides graphical tools for debugging and visualization of the groups structure mentioned above as well as a graphical user interface with various entities to input motion commands and to output the state of sensor values, parameters or return values at the different levels of the architecture. The control architecture is also mapped into a hardware control system. At the task planning and coordination

Table 1. Conditions for the Petri net in figure 4

Conditions	Description
A_1	Subsystem A is active
A_2	Subsystem A is ready
B_1	Subsystem B is active
B_2	Subsystem B is ready
C_1	Subsystem C is active
C_2	Subsystem C is ready

Table 2. Transitions and their meaning in the Petri net in figure 4

Events	Description	Pre-condition	Post-condition
1	Task of subsystems A is completed	A_1	A_2
2	Task of subsystems B is completed	B_1	B_2
3	Task of subsystems C is completed	C_1	C_2
4	New task for subsystem A	A_2, B_2, C_2	A_1, B_2, C_2
5	New task for subsystem B	A_2, B_2, C_2	A_2, B_1, C_2
6	New task for subsystem C	A_2, B_2, C_2	A_2, B_2, C_1
7	New tasks for subsystems A and B	A_2, B_2, C_2	A_1, B_1, C_2
8	New tasks for subsystems B and C	A_2, B_2, C_2	A_2, B_1, C_1
9	New tasks for subsystems A and C	A_2, B_2, C_2	A_1, B_2, C_1
10	New tasks for subsystems A, B and C	A_2, B_2, C_2	A_1, B_1, C_1

levels embedded PCs with broadband bus system (firewire) are used whereas DSP-FPGA-based controller modules are used at the execution level. They communicate with the embedded PCs via CAN-Bus. All components of the embedded control components are programmed and running under Linux, kernel 2.4.20 with the Real Time Application Interface RTAI 24.1.11. The coordination strategy has been applied and implemented in our humanoid robot. Several manipulation and locomotion experiments have been carried out to prove its suitability.

Mobile manipulation tasks: Mobile manipulation capabilities are the key of humanoid robot applications in household environments. The navigation and manipulation methods are integrated to execute service tasks like taking a bottle from a person, collision free driving to a mission target and placing it onto a table. In the case of this scenario the firing of the transitions of the associated coordination nets is specified by different events. At the beginning all subsystems of the robot arm are in the initial state *ready*. One of the arms, e.g., the right arm switches to the state *active* when the person is detected and starts the execution of the grasping task. In the case of a successful grasp, the mobile platform starts driving to the target position. The placing operation is then done through coordinated actions of the controllers of the right arm, platform and head. In all phases of the task each motion controller periodically acquires local sensory data and interprets it in order to take suitable measures.

Dual arm tasks: As an initial investigation of methods for dual arm cooperative tasks, we chose to implement an object level controller for the control of the position of the center of a lightweight dumbbell, which has to be lifted up by both arms. The arm trajectories are derived from the object trajectory. In order to achieve a coordinated execution of the task, synchronization points are included along each arm trajectory. These points represent conditions for firing of associated transitions in the coordination net.

Visual person tracking and auditory tracking: In order to make the interaction with the robot easier and more reliable, we integrated the algorithms, which has been originally developed in [11] for the visual perception of the user into the control system of our robot to demonstrate the motion coordination of the head and the mobile platform. The 3D-positions of the user's head and hands are mapped into joint angles of the robot head and arms. For the mapping of the hand positions into a robot arm postures, we use the method presented in [2] in order to generate human-like arm postures. Furthermore, an acoustic localization algorithm using a stereo microphone system, which was developed in [9], is also integrated. Once the head and hands of a person have been detected, the humanoid robot mimics the head and hand motion of the person. When the robot loses the person, the attention of the robot is regained through the acoustic localization. Our experiments indicate a robust visual and acoustic tracking of a person even when the head and the platform are moving.

6 Conclusion

The paper has introduced a hierarchical control architecture for humanoid robots as well as a framework for the coordinated execution of humanoid robots tasks. A Petri net based coordination strategy has been presented and several manipulation and locomotion experiments have been carried out to prove the suitability of the proposed strategy. The communication between different controllers of the robot and the control mechanism for coordinated task execution is efficiently specified within the Petri net framework. The papers shows that Petri nets can be used as an efficient tool for the coordinated motion control in robotic systems with a high degree of freedom such as humanoid robots. Many functionalities were integrated through an event-based coordination scheme in order to realize various tasks in household environments.

Acknowledgment

This work has been performed in the framework of the german humanoid robotics program SFB 588 (project R1) funded by the *Deutsche Forschungsgemeinschaft (DFG)*.

References

1. R. David and H. Alla, “Petri Nets for Modeling of Dynamic Systems,” *Automatica*, vol. 30, pp. 175–202, 1994.
2. T. Asfour and R. Dillmann, “Human-like Motion of a Humanoid Robot Arm Based on Closed-Form Solution of the Inverse Kinematics Problem,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), 27-31 October, 2003.
3. T. Asfour, K. Berns and R. Dillmann, “The Humanoid Robot ARMAR: Design and Control,” The 1st IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS 2000), 7-8 September, 2000.
4. R. A. Brooks, “A Robust Layered Control System for a Mobile Robot,” IEEE Journal on Robotics and Automation, vol. RA-2, pp. 14–23, 1986.
5. C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems,* Kluwer Academic Publ. 1999.
6. P. Hsu, “Control of Multi-Manipulator Systems: Trajectory Tracking, Load Distribution, Internal Force Control and Decentralized Architecture,” *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1234–1239, 1989.
7. S. Arimoto, F. Miyazaki and S. Kawamura, “Cooperative Motion Control of Multiple Arms or Figures,” *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1407–1412, 1987.
8. O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg and A. Casal, “Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation,” Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 96), vol. 2, pp. 546 – 553. 1996
9. D. Bechler, M. Schlosser and K.Kroschel. “Acoustic 3D Speaker Tracking for Humanoid Robots with a Microphone Array,” *Proceedings of the 3rd IEEE International Conference on Humanoid Robots (Humanoids 2003)*, Karlsruhe, Germany.
10. Y-J. Cho, J-M. Park, J. Park, S-R. Oh and C. W. Lee “A Control Architecture to Achieve Manipulation Task Goals for a Humanoid Robot,” *Proceedings of the IEEE Intern. Conf. on Robotics and Automation*, Leuven, Belgium, pp. 206–212, 1998.
11. K. Nickel and R. Stiefelhagen. “Pointing gesture recognition based on 3d-tracking of face, hands and head orientation,” *International Conference on Multimodal Interfaces*, Vancouver, Canada, 2003.
12. G. N. Saridis and K. P. Valvanis, “Analytic Design of Intelligent Machines,” *Automatica*, vol. 24, pp. 123–133, 1988.
13. F. Y. Wang, K. J. Kyriakopoulos, A. Tsolkas and G. N. Saridis, “A Peti Net Coordination Model for an Intelligent Mobile Robot,” *IEEE Transactions on System, Man and Cybernetics*, vol. 21, no. 4, pp. 777–789, 1991.
14. T. J. Tarn, A.K. Bejczy and X. Yun, “Coordinated Control of Two Robot Arms,” *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1193–1202, 1986.
15. ATI Industrial Automation Homepage: http://www.ati-ia.com/sensors.htm
16. Modular Controller Architecture (MCA2): http://mca2.sourceforge.net/
17. Sick-Homepage: http://www.sick.de/de/products/

Imitation Learning Based on Visuo-Somatic Mapping

Minoru Asada*, Masaki Ogino, Shigeo Matsuyama, and Jun'ichiro Ooga

Adaptive Machine Systems and *HANDAI FRC, RoboCup Humanoid Robot Project Leader
Graduate School of Engineering
Osaka University, 2-1, Yamada-Oka, Suita, Osaka 565-0871, Japan
{asada, ogino, shigeo, ooga}@er.ams.eng.osaka-u.ac.jp
http://www.er.ams.eng.osaka-u.ac.jp/

Abstract. Imitation learning is a powerful approach to humanoid behavior generation, however, the most existing methods assume the availability of the information on the internal state of a demonstrator such as joint angles, while humans usually cannot directly access to imitate the observed behavior. This paper presents a method of imitation learning based on visuo-somatic mapping from observing the demonstrator's posture to reminding the self posture via mapping from the self motion observation to the self posture for both motion understanding and generation. First, various kinds of posture data of the observer are mapped onto *posture space* by self organizing mapping (hereafter, SOM), and the trajectories in the posture space are mapped onto a *motion segment space* by SOM again for data reduction. Second, optical flows caused by the demonstrator's motions or the self motions are mapped onto a *flow segment space* where parameterized flow data are connected with the corresponding motion segments in the motion segment space. The connection with the self motion is straightforward, and is easily acquired by Hebbian Learning. Then, the connection with the demonstrator's motion is automatic based on the learned connection. Finally, the visuo-somatic mapping is completed when the posture space (the observer: self) and image space (the demonstrator: other) are connected, which means observing the demonstrator's posture associcates the self posture. Experimental results with human motion data are shown and the discussion is given with future issues.

1 Introduction

Recent progress of humanoids such as ASIMO [9] and QRIO [6] has been attracting many people for their human-like behaviors towards symbiotic relationship between humans and robots. However, the current design and implementation of these behaviors mainly owes to the designers' deep knowledge and skills. In order to realize the truly symbiotic relationship, the robots are expected to be much more adaptive and flexible so that they can understand and generate human's various motions.

Imitation learning is one of the most powerful approach to humanoid behavior generation [11]. The most existing methods assume the availability of the information on the internal state of a demonstrator such as joint angles, which humans usually cannot directly access to imitate the observed behavior [3][7]. Some studies have been conducted without this assumption. Asada et al. [1] proposed a method for learning by observation based on the demonstrator's view recovery and adaptive visual servoing. Kuniyoshi et al. [5] showed the view-based imitation based on the similarity in the optical flows and on the association with motor commands.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 269–278, 2006.

These studies are intended not simply for efficient behavior generation, rather for understanding how humans learn to imitate the observed behaviors. However, the variety of the behaviors seems limited due to their simple flow matching methods.

This paper presents a method of imitation learning based on visuo-somatic mapping from observing the demonstrator's posture to recalling the self posture. First, various kinds of posture data of the observer are mapped onto a *posture space* by self organizing mapping [4] (hereafter, SOM), and the trajectories in the posture space are mapped onto a *motion segment space* by SOM again for data reduction. Second, optical flows caused by the demonstrator's motions or the self motions are mapped onto a *flow segment space* where parameterized flow data are connected with the corresponding motion segments in the motion segment space. The connection with the self motion is straightforward, and is easily acquired by Hebbian Learning. Then, the connection with the demonstrator's motion is automatic based on the learned connection. Finally, the visuo-somatic mapping is completed when the posture space (the observer: self) and image space (the demonstrator: other) are connected, which means observing the demonstrator's posture associates the self posture. Experimental results with human motion data are shown and the discussion is given with future issues.

2 A System Overview

2.1 Basic assumptions

Here, we assume the followings to realize the visual imitation based on the visuo-somatic mapping:

1. No *a priori* knowledge on the link structure, that is, connections between joints.
2. No *a priori* knowledge on the body part (joint) correspondence between the demonstrator and the observer.
3. Both the demonstrator's and the self motions can be observed in terms of a temporal sequence of joint vectors in image space.
4. The joint angles of the self posture can be observed, but no relationship between the self posture and the visual feature space is given.
5. Currently, we focus on the mirror image imitation. This means the right (left) side of the demonstration corresponding to the left (right) side side of the observation.

2.2 Imitation System

Fig. 1 shows the proposed system, consisting of two sub-processing systems, *Visual Information processing system* and *Somatic Information processing system*. In these processing sub-systems, row sensory data are mapped onto the corresponding two dimensional Self-Organizing Maps (SOMs) [4]. The images observing the demonstrator's motion are first mapped onto an *image space* which includes the posture image of the demonstrator, and then *flow segment space* in which the changes in posture are represented. The flow segment space is also utilized to represent the self motion, too. On the other hand, the self somatic sensory data are mapped onto a *posture space* and their changes are mapped onto a *motion segment space*. After generation of these maps independently, the flow segment space and the motion segment space are connected based on Hebbian learning.

The connection between the flow segment space and the motion segment space is easily carried out by using Hebbian learning based on the simultaneous activations of segments in both spaces during the self motions. Once this connection is acquired, the connection between the flow segment space for the demonstrator's motion and the motion segment space is automatic based on the learned connection between the flow segment space and the motion segment space. Through these connections, the mapping from the image space of the demonstrator's posture to the self posture space is enabled, that is, visuo-somatic mapping can be obtained.

In the followings, the details of each sub processing system are explained in section 3, and the mapping among them are shown in section 4.

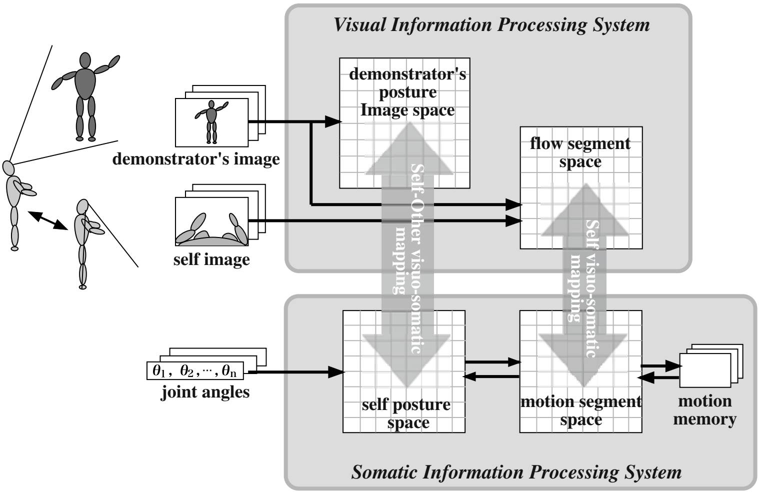
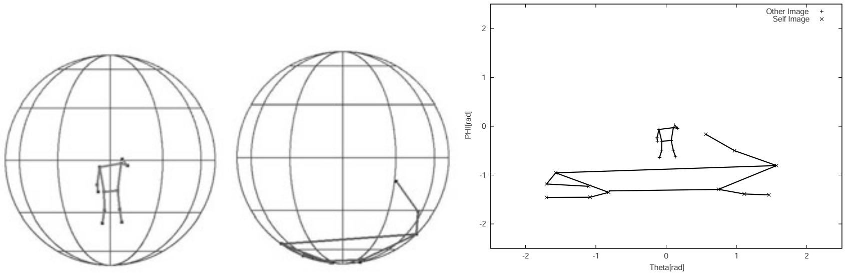

Visual Information Processing System
demonstrator's posture Image space
flow segment space
demonstrator's image
self image
Self-Other visuo-somatic mapping
Self visuo-somatic mapping
$\theta_1, \theta_2, \cdots, \theta_n$
joint angles
self posture space
motion segment space
motion memory
Somatic Information Processing System

Fig. 1. System overview

2.3 Sensory Data

We prepare the sensory data by using human motions acquired by a motion capturing system. The captured data, which are three dimensional data sets in the global coordinate system, are converted to the two dimensional data on a virtual camera images captured by the observer (self). The angles between links in a human model are also calculated to be used as the self posture data. A joint angle vector consisting 16 joint angles is mapped onto the self posture space and the segmented trajectories on the map are mapped on motion segment space. The spherical image projection from the camera position at the observer head is assumed to capture the whole self body image. Fig. 2 shows examples of the self body image (a) and the demonstrator's one (b) on the spheres, and their development onto a plane (c).

Twelve kinds of motions are captured from the human motion performances. They are combinations of motion, side, and part such as "raise," "wave," and "rotate" as motions, "left," "right," or "both" as sides, and "hand," and "knee" as parts. Also a "walking" motion is added as a whole body motion.



(a) Image of a demonstrator

(b) Image of Self body

(c) The development of the spherical projection

Fig. 2. Image data to be input to the system

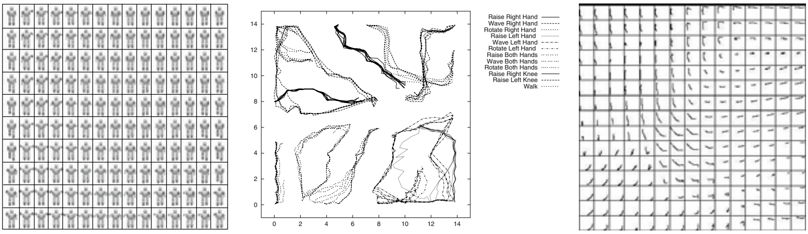
3 Construction of SOMs for behavior recognition and generation

3.1 Posture space and motion segment space

We construct a *posture space* SOM from the *somatic information*, which is a sequence of the vectors consisting of sixteen joint angles between links calculated from the captured motion data. The size of *posture space* SOM is 15 × 15, and it is constructed by 240 [frames] (8 [sec]) per each motion. Fig. 3 (a) shows the resultant SOM, where 15 × 15 postures are representative ones from row data.

Since the posture data are input sequentially, we can visualize how posture data are connected each other in the posture space. Fig. 3 (b) shows such data indicating that the trajectries of motions are roughly segmented and construct the clusters corresponding to performed actions.

These trajectories are divided into small segments, each of which consists of 10 [frames] temporal sequence of posture data, and are clustered into another SOM, *motion segment space*, (Fig. 3 (c)).



(a) Posture space

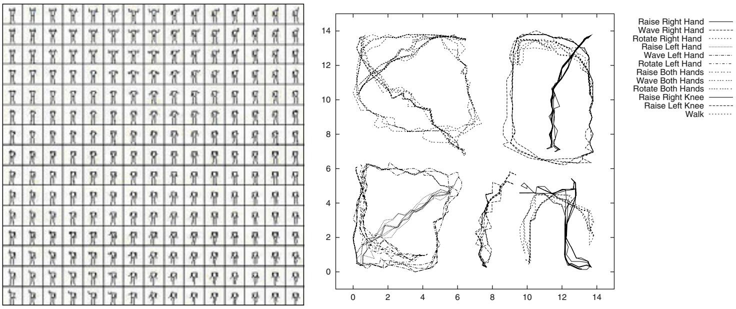
(b) Trajectories of motions on posture space

(c) Motion Segment Space

Fig. 3. The SOMs constructed in the self somatic information processing system

3.2 Demonstrator's posture image space

A demonstrator's posture image space (hereafter, image space in short) consists of the representative image position vectors obtained by self organizing mapping of image positions of joints of the human model. Fig. 4 (a) shows the image space where various postures are clustered into 15 × 15 representative postures. Similar to the posture space based on the somatic information, we can visualize how posture image data are connected each other in this space. Fig. 4 (b) shows such data which indicate that the trajectories of motions are roughly segmented and construct the clusters corresponding to performed actions.



(a) *Demonstrator's posture image space*

(b) Trajectories of motions on demonstrators posture image space

Fig. 4. Demonstrator's posture image space

3.3 Flow segment space construction

The problem here is how to associate the observed flow caused by the demonstrator's motion with the self motion. If the flows by the demonstrator's motions are similar to the flows by the self motions, the desired association seems easy to find because the connection between the observed self motion and the self motion segment can be easily found based on the simultaneous activations during the self motion. However, it would not be so due to the viewpoint difference. Then, as the common features of the flow segments, we chose the direction and the relative position of the flow segments. The directions of flow segments by the demonstrator (other) and the observer (self) are very similar to each other although there are slight differences in the directional changes between them. The relative positions are quantized into four regions (top left, top right, bottom left, and bottom right) by setting the centroid of posture image vectors as the origin. These four regions are called attention areas. The positions of joints are similar to each other between the demonstrator's and the observer's in spite of large shape difference.

By using the quantized directions and the normalized magnitudes of the flows, and the attention area, the flow segment space is constructed. A data structure for the flow segment space is shown in Fig. 5 (a).

Flow segment space The directions of the flows are segmented when the sign of horizontal or vertical element of flow vector is inverted. In each segment, the directions are averaged. Suppose the time when n-th flow inversion happens is T_n, then the averaged flow direction is given by

$$\S_i^{\phi}(t) = \frac{1}{T_{n+1} - T_n} \int_{T_n}^{T_{n+1}} \phi_i^F(s)ds \quad (T_n < t < T_{n+1}), \tag{1}$$

where $\phi_i^F(t)$ indicates the flow direction of body segment i at time t. The averaged direction data are sorted by their length of the flow vectors. And the i-th data is assigned to the i-th layer in *flow segment space*. In each layer, the unit that has the nearest direction to the input data is activated.

Attention area space Although the positions of flow vectors in the robot's view are quite different between the self and the demonstrator, the relative positions among them (upper right, upper left, lower right and lower left) are roughly maintainted well. Using this feature, the *attentional area space* describes what part of the self image includes first flow vectors.

Let N_f the number of observed points of the self and the demonstrator's body and the regions around the center of observed points R_1, R_2, R_3 and R_4, then the total flow speed included in each region R_i is given by

$$F_j(t) = \int_{i=1}^{N_f} p_i(t)||v_i(t)||, \tag{2}$$

$$p_i(t) = \begin{cases} 1 \; if & u_i(n) \in R_j \\ 0 & else \end{cases} \quad (j = 1, \cdots 4), \tag{3}$$

where $v_i(t)$ is the observed flow vector, and $u_i(t)$ is the position vector of observed point i at time t. Note that i does not correspond to the labeled point of the body. We define the relative total strength of flow among regions as

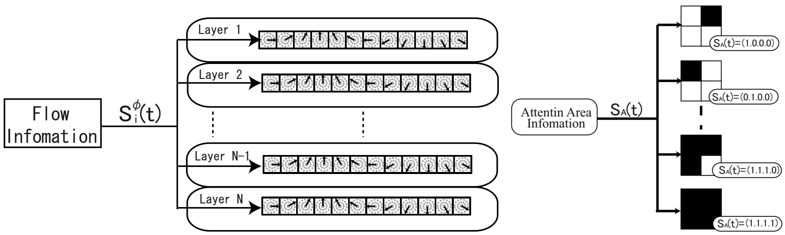
$$A_j(t) = \frac{F_j(t)}{\sum_{n=1}^{4} F_n(t)} \quad (j = 1, \cdots 4). \tag{4}$$

The input vector to *attentional area space*, $S_A(t)$, consists of binarized $A_j(t)$,

$$S_A(t) = (A_1^S(t), A_2^S(t), A_3^S(t), A_4^S(t)) \tag{5}$$

$$A_j^S(t) = \begin{cases} 1, if & A_j(t) \geq 0.20 \\ 0, & else \end{cases} \tag{6}$$

Attention area space consists of all the combinations of activated areas, $2^4 = 16$.



(a) Flow direction (b) Attention area

Fig. 5. Flow segment space

4 Mapping between visual and somatic field

4.1 Self Visual-Somatic sensation Mapping

The simultaneous activations of the units in the flow segment space and the self posture space during self motion make it possible to find correspondence between the units in those spaces.The connection coefficients between the units in each space are learned based on Hebbian learning. All the connection coefficients are initialized to 0s, and during the self motion the coefficient, w^{AB}, which is the connection coefficient betwen the i-th unit in space A and the j-th unit in space B, is updated during self motion, as follows,

$$w_{ij}^{AB}(t+1) = w_{ij}^{AB}(t) + \varepsilon(y_i^A(t)y_j^B(t) - y_i^A(t)^2 w_{ij}^{AB}(t)). \tag{7}$$

At the same time, the time sequences of the activated units in motion segment space during various motions are memorized as the motion modules in *motion memory*.

4.2 Recognition of other person's motion

After acquisition of self visual-somatic mapping, the input of image data observing a demonstrator's motion activates the units in *motion segment space* through *visual information processing system* via connections between them. Let the activation level of the i-th unit in flow direction $y_i^F(t)$ and that of the j-th unit in attention area $y_j^A(t)$, the activation level of the k-th unit in motion segment space, $y_k^M(t)$ is given by

$$y_k^M(t) = \sum_{i=1}^{N_F} w_{ik}^{FM} y_i^F + \sum_{j=1}^{N_A} w_{jk}^{AM} y_j^A \tag{8}$$

The quantization in the flow segment space is coarse and the mapping between the flow segment space and the motion segment space is not one-to-one mapping. The motion of a demonstrator activates multiple units in the motion segment space at a time, which makes it difficult to identify the corresponding motion module. So, we compare the temporal sequences of activated units of observed motion with those of memorized motion modules in the motion segment space. To do that, we define the evaluation function, E_m, which indicates the similarity of the time sequence of acvtiated units of an observed motion to that of the m-th memorized motion as follows,

$$E_m = \max_{s_t} \int_0^T \sum_{i=1}^{N_M} y_i(t) m_i(t_s + t) dt. \tag{9}$$

Thus, the observed motion is recognized as the same as the motion module that maximizes E_m.

4.3 Mapping between a self posture image space and a demonstrator's posture image

Recalling the self motion from the observation of a demonstrator's motion makes it possible to correlate the demonstrator's posture image space in visual information processing system with the self posture space in somatic sensation information processing system. When observing a demonstrator's motion, the unit in image space and the unit in posture space activate simultaneously. So we can use Hebbian learning again between these two maps.

Fig. 6 shows the recalled posture (the rightmost figure) from the observed image (the leftmost figure). The two maps in the middle of the figures describe the activated units in image space(left) and posture space (right) after Hebbian learning.

5 Discussion

In this paper we proposed a learning system for imitation based on visuo-somatic mapping. This system excludes the pre-designed model of the demonstrator as much

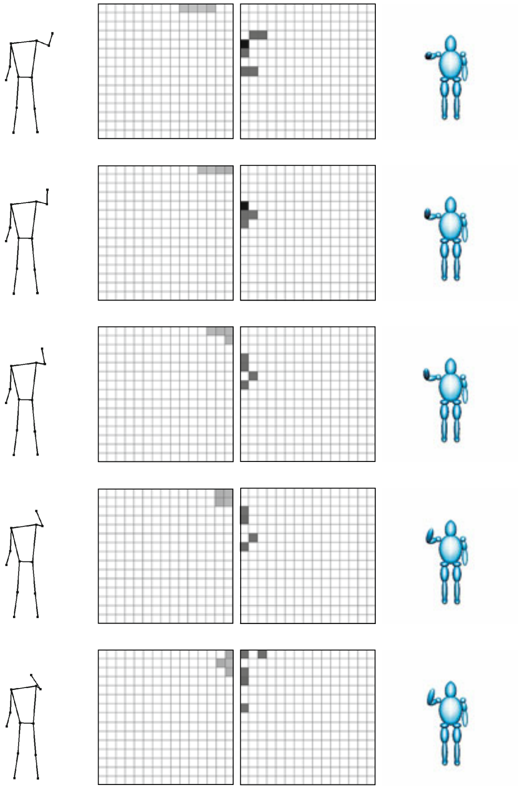


Fig. 6. Recalling somatic sensation by Self-Other visuo-somatic sensation mapping (The left two figures show demonstrator's image and its mapping on a demonstrator's posture image space. The right two figures show the correponding representative vectors in the self posture space and the corresponding self postures.

as possible. The demonstrator's model is made through demonstrator's images in a demonstrator's posture image space. The model of self is not pre-designed, either. It is constructed by self-organizing the self motion information in self posture space and motion segment space. The primitive visual features are related to the representative vectors in motion segment space during self motion. This connection induces the self motion when observing demonstrator's motions and further mapping between demonstrator's posture image and self posture space is made. After constructing the visuo-somatic mapping, this system can directly activate the self posture corresponding the observed demonstrator's image.

Although initial aims to construct the visuo-somatic mapping through learning are accomplished in this system, it has many problems for practical use as an imitation system. First this system assumes that an observer always stands face to face with a demonstrator, and this sytem does not have concept about the translation or rotation to the ground of the demonstrator. An observer can recognize only jestures of the demonstrator. Second, the resultant visuo-somatic mapping is not so accurate as to make new motion modules only from observation of demonstrator's motion, because the sequence of the activated postures is not smooth.

For the first problem, we are now extending our model so that it can describe the transition and rotation of a demonstrator relative to the ground. The second problem can be solved by using velocity information acquired by another pathway. Acquiring new motions which are not experienced through observation is the next challenge for us.

Acknowledgments This study was performed through the Advanced and Innovational Research program in Life Sciences from the Ministry of Education, Culture, Sports, Science and Technology of the Japanese Government.

References

1. M. Asada, Y. Yoshikawa, and K. Hosoda, "Learning by observation without three-dimensional reconstruction," In *Proc. of the 6th Int. Conf. on Intelligent Autonomous Systems (IAS-6),* pp. 555–560, 2000.
2. A. Billard, M. Mataric: "Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture", *Robotics and Autonomous Systems*, 37, pp.145–160, 2001.
3. T. Inamura, I. Toshima, and Y. Nakamura. "Acquisition and embodiment of motion elements in closed mimesis loop," In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1539–1544, 2002.
4. T. Kohonen: *The Self Organization and Associative Memory*, Springer-Verlag, 1989.
5. Y. Kuniyoshi, Y. Yorozu, M. Inaba, and H. Inoue, "From visuo-motor self learning to early imitation – A neural architecture for humanoid learning," In *Proc. of the 2003 IEEE Int. Conf. on Robotics and Automation*, pp. 3132–3139, 2003.
6. Y. Kuroki, T. Ishida, and J. Yamaguchi, A Small Biped Entertainment Robot, In *Proc. of IEEE-RAS Int. Conf. on Humanoid Robot*, pp. 181-186, 2001.
7. A. Nakazawa, S. Nakaoka, K. Yokoi, T. Harada, and K. Ikeuchi: "Imitating human dance motion through motion structure analysis", In *Proc. of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2539–2544, 2002.
8. E. Oja, J. Karhunen: "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix", *J. Math. Anal. and Appl.*, 106, pp. 69–84, 1985.
9. M. Hirose, Y. Haikawa, T. Takenaka, and K. Hirai, Development of Humanoid Robot ASIMO, In *Proc. Int. Conf. on Intelligent Robots and Systems*, 2001.
10. R. Pfefer and C. Scheier.: *Understanding Intelligence*, MIT Press, Cambrridge, Massachusetts, 1999.
11. S. Schaal., "Is imitation learning the route to humanoid robots?," *Trends in Cognitive Science*, pp. 233–242, 1999.

VIII. Designs

Designing is the drawing board where many new and innovative ideas in robotics are born. Robots today no longer come in the form of arms and wheels. Complex task requirements dictate the need for many new designs.

The first article by Sugiyama and Hirai starts the chapter with a refreshingly different type of robot: a deformable robot in the shape of only a wheel. This wheel is controlled to crawl, climb and even jump.

The second article updates the world on the progress of the design of an exoskeleton system by Kazerooni's group from UC Berkeley. The system promises to convert any person into "superman" with the design of a wearable robotic system that helps the person to carry load on his backpack. It reduces fatigue and would one day make moving heavy load on your back like a walk in the park.

The third article by Suthakorn and Chirikjian presents an attempt to bring into reality an idea that was conceived over half a century ago: man-made self-replicating machine. This trait of a living organism is still relatively underdeveloped in the world of robotics. As the authors admitted that an autonomous reproduction of a man-made machine is still not a reality, this article presents an experiment of a replication process of transistor circuit by self-inspection. The authors concluded that self-replication by self-inspection appeared to be a more robust and less complicated alternative to that by the conventional Von Neumann universal constructor.

In the fourth article, Tonietti and colleagues discuss the importance of robot safety. The concept of robot safety is discussed and the design parameters are reviewed to achieve what they term as the *maximum suitable level of injury risk*. This is a very important and relatively new area of research that would enable robots to be safely integrated into human environments.

The last article by Yesin, Vollmers and Nelson investigates the design of microrobots that can be introduced into the human body and actuated through external magnetic fields. It also covers the position sensing problem of intraocular robots where visual servoing was used to control the position of the microrobots in the eye.

Crawling and Jumping by a Deformable Robot

Yuuta Sugiyama and Shinichi Hirai

Dept. Robotics, Ritsumeikan Univ.
Kusatsu, Shiga 525-8577, Japan
hirai@se.ritsumei.ac.jp
http://www.ritsumei.ac.jp/se/˜hirai/

Abstract. We describe crawling and jumping by a deformable robot. Locomotion over rough terrain has been achieved mainly by rigid body systems including crawlers and leg mechanisms. This paper presents an alternative method of moving over rough terrain, one that employs deformation. First, we describe the principle of crawling and jumping as performed through deformation of a robot body. Second, in a physical simulation, we investigate the feasibility of the approach. Next, we show experimentally that a prototype of a circular soft robot can crawl and jump.

Keywords deformation, locomotion, crawl, jump

1 Introduction and State-of-the-Art

Rough terrain locomotion has mainly relied on rigid body systems, such as crawlers and leg mechanisms. This paper presents an alternative approach that uses deformation.

Locomotion mechanisms consisting of rigid body systems have drawbacks: large weight that may cause impact to humans and difficulty in recovery from their overturning. Recently, mechanisms that can recover from their overturning have been studied [1,2], but these mechanisms tend to be complicated. An alternative approach to light-weighted and simple mechanisms is thus required. Recent researches on soft actuators such as shape memory alloy (SMA) wires and polymer gel actuators has yielded impressive results [3–5], and soft actuators have been used to drive leg mechanisms and soft body robots [6]. Unfortunately, soft actuators still have drawbacks. They tend to generate a small force, and those that generate a large force need either a high driving voltage over 1,000V, making it difficult to build self-supporting robots, or a wet environment. To overcome this problem, we employ soft actuators to controllably deform a robot body, enabling it to crawl over and jump on rough terrain. Crawling and jumping using deformation can cope with rougher terrain than rigid body systems can. Additionally, soft body deformation reduces the damage in collision with humans.

In this paper, we propose a circular soft robot and describe its performance in a simulation and in a practical experiment. First, we describe the principle of crawling and jumping as performed through deformation of a robot body. Second, in a physical simulation, we investigate the feasibility of the approach. Next, we show experimentally that a prototype of a circular soft robot can crawl and jump.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 281–290, 2006.

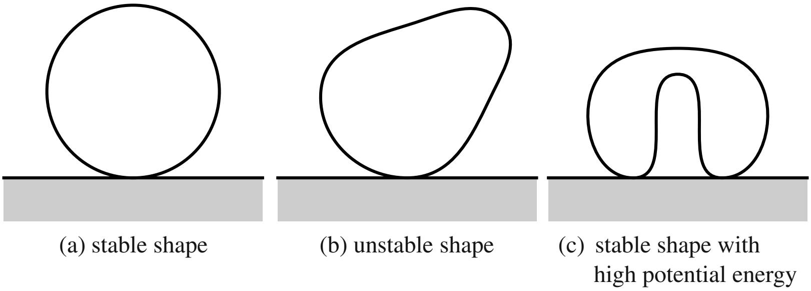


Fig. 1. Principle of crawling and jumping

2 Principle of Crawling and Jumping by Deformation

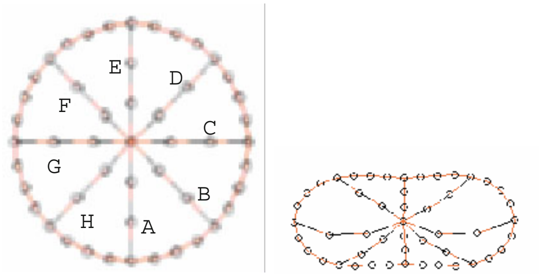
Suppose a robot is in stable on the ground, as illustrated in Figure 1-(a). Self-deformation of the robot body generates a moment by a gravitational force around the area the robot is in contact with the ground. The moment causes the robot to move on the ground. If the robot deforms from a stable shape into an unstable shape described in Figure 1-(b), it rotates clockwise and moves towards the right. Successive deformation of the robot body, which can be generated by actuators, enables a continuous crawling motion along the ground. Thus, the proposed crawling approach uses gravitational potential energy.

Deformation allows elastic potential energy to be stored which, if released rapidly enough, can generate a force large enough to make the robot jump. Now suppose the robot deforms from one stable shape into another, which has large high potential energy as illustrated in Figure 1-(c). If the potential energy is released rapidly enough, the robot will jump. The high-energy shape shown in Figure 1-(c) turns, with a small disturbance, into the stable shape shown in Figure 1-(a), generating the force required for the jump. Thus, the proposed approach uses elastic potential energy. Actuators inside the robot body can be used to store this elastic energy. The forces required to store the elastic energy is generally much smaller than those required to perform a jump.

3 Feasibility Assessment through Physical Simulation

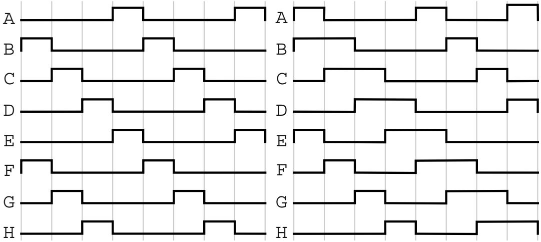
In this section, in a physical simulation, we assess the feasibility of a deformable robot to crawl and jump. As mentioned in the previous section, crawling and jumping can be performed using the elastic potential energy associated with deformation. Let us verify this approach through a physical simulation before we go on to a prototype of a deformable robot.

Let us simulate the behavior of the circular soft robot illustrated in Figure 2. The circular soft robot consists of a circular elastic shell with a set of soft actuators inside, as shown in Figure 2-(a). The robot has eight SMA coils labelled as A through H. Extending or shrinking actuators deforms the robot body, i.e., the circular shell, as



(a) initial shape (b) deformed shape

Fig. 2. Circular soft robot



(a) pattern BF (b) pattern ABE

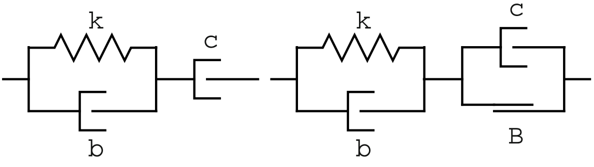
Fig. 3. Voltage patterns applied to SMA coils

shown in Figure 2-(b). We apply open-loop PWM control to the coils. A periodic voltage pattern is applied to the set of SMA coils during crawling. As illustrated in Figure 3, periodic voltage patterns are denoted by the set of coils active during the first time step.

The elastic shell of the robot is modeled as an elastic object while actuators are modeled as rheological objects [7], so as to be able to describe the inelastic nature of the SMA coils and polymer gel actuators. We can specify the contraction rate, maximum contraction, and maximum generated force of an SMA coil using a three-element model with a slider.

The extension of an elastic shell is described by a Voigt model, while its bend is modeled as an elastic element. The Voigt model for extension is a parallel connection of an elastic element k_{body} and a viscous element b_{body}. The elastic element for bend deformation is denoted as k_{bend}. We have experimentally identified model parameters for the elastic shell of a prototype of a circular soft robot in advance: $k_{body} = 500$N/m, $b_{body} = 0.1$N/(m/s), and $k_{bend} = 0.0015$Nm/rad.

Let us first formulate the passive deformation of an SMA coil that deforms in response to an applied external force. SMA coils show both viscoelastic and plastic deformation properties, which suggests their deformation can be modeled by a three-element model. A three-element model is a serial connection of a Voigt element and a viscous element, as illustrated in Figure 4-(a). The elastic coefficient k and damping coefficient b specify the Voigt element, while the viscous coefficient c characterizes



(a) three-element model (b) three-element model with slider

Fig. 4. Model of SMA coils

the viscous element. Let x be the length of a three-element model. Let x_v and x_d be the lengths of a Voigt model and a viscous element, respectively. The three-element model can be formulated as follows:

$$x = x_v + x_d, \tag{1}$$

$$f_{pas} = -kx_v - b\dot{x}_v, \tag{2}$$

$$f_{pas} = -c\dot{x}_d \tag{3}$$

where f_{pas} describes a passive force generated by the element.

A three-element model can extend as long as an external force is applied to it. To avoid such limitless extension, we employ a three-element model with a slider illustrated in Figure 4-(b). The slider is specified by force limit B. In a three-element model with a slider, eq.(3) is replaced by the following equation:

$$-c\dot{x}_d = \begin{cases} f_{pas} & \text{if } fx \leq Bx_v^{\text{init}} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

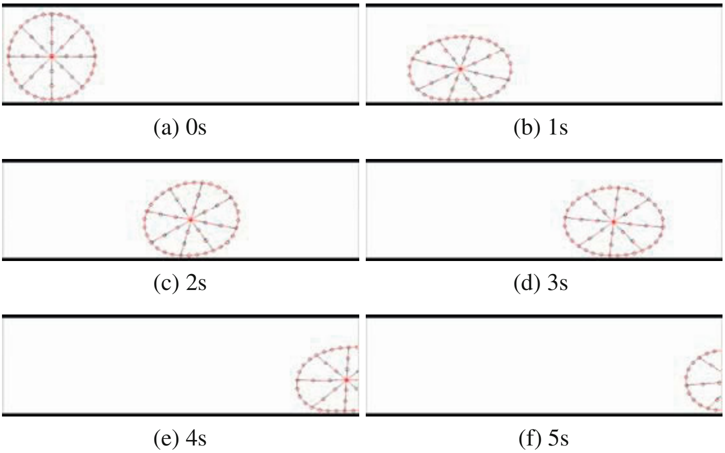
where f stands for the resultant external force applied to the element and x_v^{init} is the initial length of the Voigt model in the element.

An SMA coil actively generates a force that is determined by the voltage applied to it. Let us next formulate this actively generated force. Let $V(t)$ be the voltage applied to a coil. We apply open-loop PWM to the generation of a force by a coil. That is, voltage $V(t)$ alternates between V or 0. Let $F(t)$ be a force actively generated by an SMA coil at time t. Let D_{on} be the contraction force rate of the coil and D_{off} its relaxation force rate. Let F_{max} be the maximum force that can be generated by the coil. The force generated by the SMA coil can then be expressed as:

$$\frac{\mathrm{d}F}{\mathrm{d}t} = \begin{cases} D_{on} & V(t) = V \text{ and } F(t) < F_{max} \\ -D_{off} & V(t) = 0 \text{ and } F(t) > 0 \\ 0 & \text{otherwise} \end{cases} . \tag{5}$$

Integration of the above equation over the time interval $[0, t]$ yields the actively generated force at time t. Note that force $F(t)$ varies in the range $[0, F_{max}]$.

We have used SMA coils BMX100 to build a prototype of a circular soft robot. We have experimentally identified the model parameters to be used in eqs.(1)-(5):


(a) 0s (b) 1s

(c) 2s (d) 3s

(e) 4s (f) 5s

Fig. 5. Simulation of a circular soft robot crawling

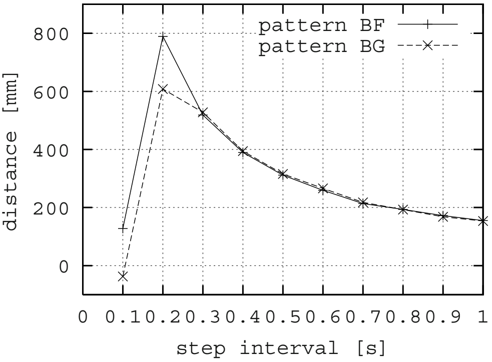


Fig. 6. Comparison of pattern BF and BG

$k = 50\text{N/m}$, $b = 0.1\text{N/(m/s)}$, $c = 10\text{N/(m/s)}$, $B = 0.016\text{N}$, $D_{on} = D_{off} = 150\text{mN/s}$, and $F_{max} = 150\text{mN}$.

Figure 5 shows the simulation results of the crawling of a circular soft robot. A periodic voltage pattern is applied to one or more of the SMA coils during the crawl. In this simulation, pattern BF illustrated in Figure 3-(a) was used to activate the coils. The figure shows that a circular robot can crawl on a flat terrain by open-loop PWM control of eight SMA coils. Thus, we can find an appropriate voltage pattern through simulation. Figure 6 shows locomotion distances covered over 10s at various step intervals. As shown in the figure, voltage pattern BF with a step interval of 0.2s yields the better result.

Figure 7 shows the simulation results for a jump performed by a circular soft robot. The results suggest that the robot can jump under gravity through PWM control of coils.

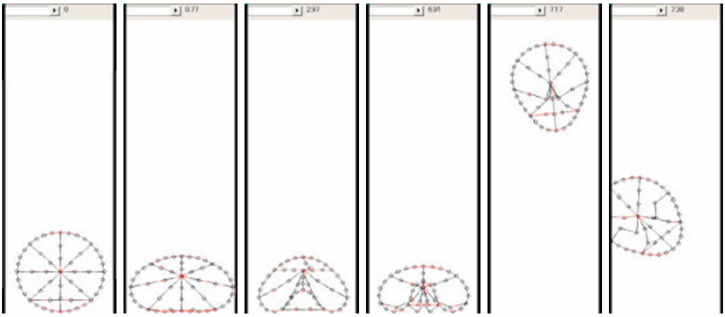
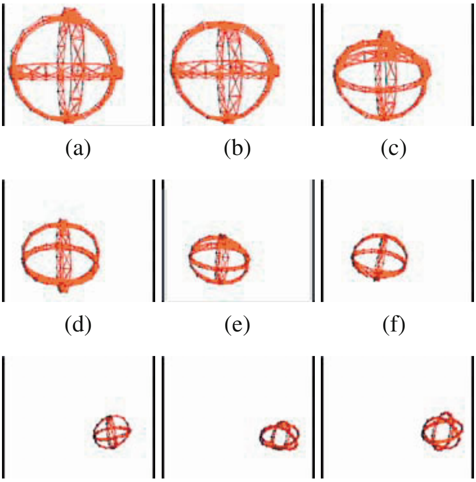

(a) 0s (b) 2s (c) 4s (d) 8s (e) 9s (f) 10s

Fig. 7. Simulation of a circular soft robot jumping


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 8. Crawling of a spherical soft robot

We can simulate the 3D motion of a deformable robot. Figure 8 describes the crawling of a spherical soft robot. The robot consists of three circular elastic shells, which are deformed by contraction of soft actuators located inside the robot.

4 Experimental Results

We build a prototype of a circular soft robot to assess experimentally the feasibility of a deformable robot crawling and jumping. The prototype shown in Figure 9 consists of eight BMX100 SMA coils, labelled A through H, attached to the inside of a circular rubber shell. The diameter of the circular body is 40mm and the robot weighs 3g. When voltage is applied to a coil, it contracts, resulting in the circular

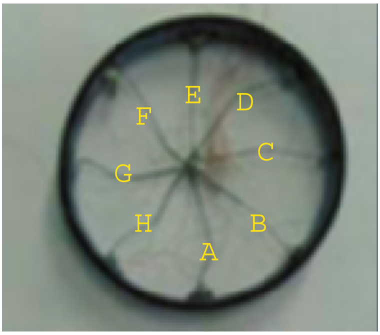
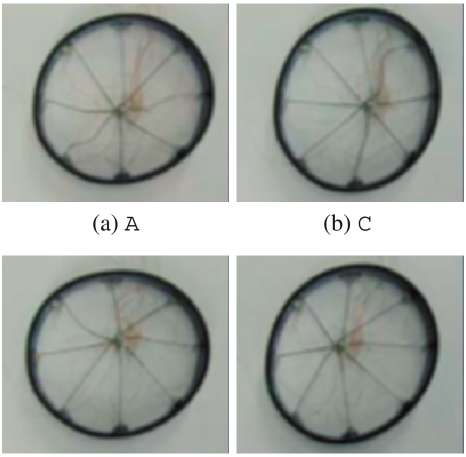


Fig. 9. Prototype of a circular soft robot



(a) A

(b) C

(c) E

(d) G

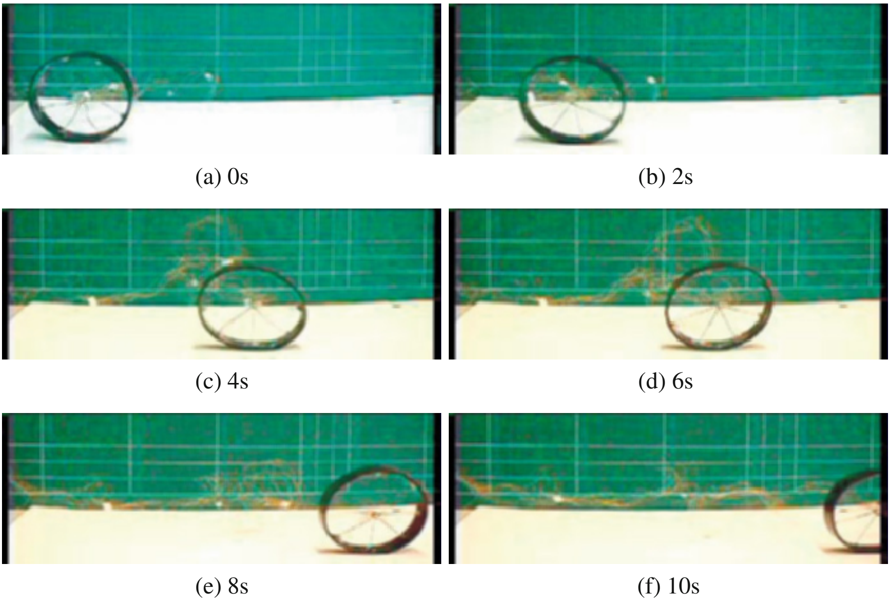
Fig. 10. Deformation of a circular soft robot

rubber deforming as shown in Figure 10. Each figure corresponds to the deformation caused by the contraction of an individual coil.

Figure 11 shows a sequence of snapshots of the prototype crawling. Voltage pattern BF is applied to SMA coils. As shown in the figure, the circular robot can crawl on a flat ground.

Let us compare the simulation and experimental results for crawling. Figure 12 describes locomotion distances covered over 10s at various step intervals. The prototype moves 260mm over 10s at a step interval of 0.6s. As shown in the figure, simulation results agree with experimental results. Reducing the step interval results in faster locomotion.

Figure 13 shows a sequence of snapshots of the prototype climbing a slope. The prototype can climb up a slope of 20° by applying pattern ABE illustrated in Figure 3-(b).


(a) 0s

(b) 2s

(c) 4s

(d) 6s

(e) 8s

(f) 10s

Fig. 11. Circular soft robot crawling

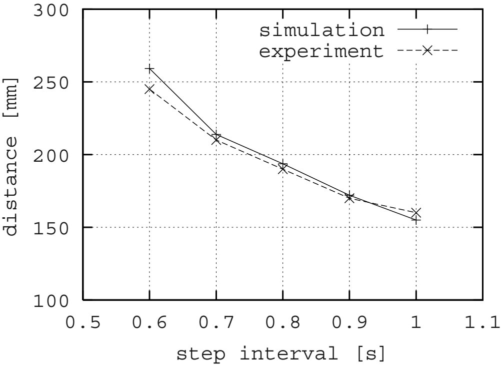
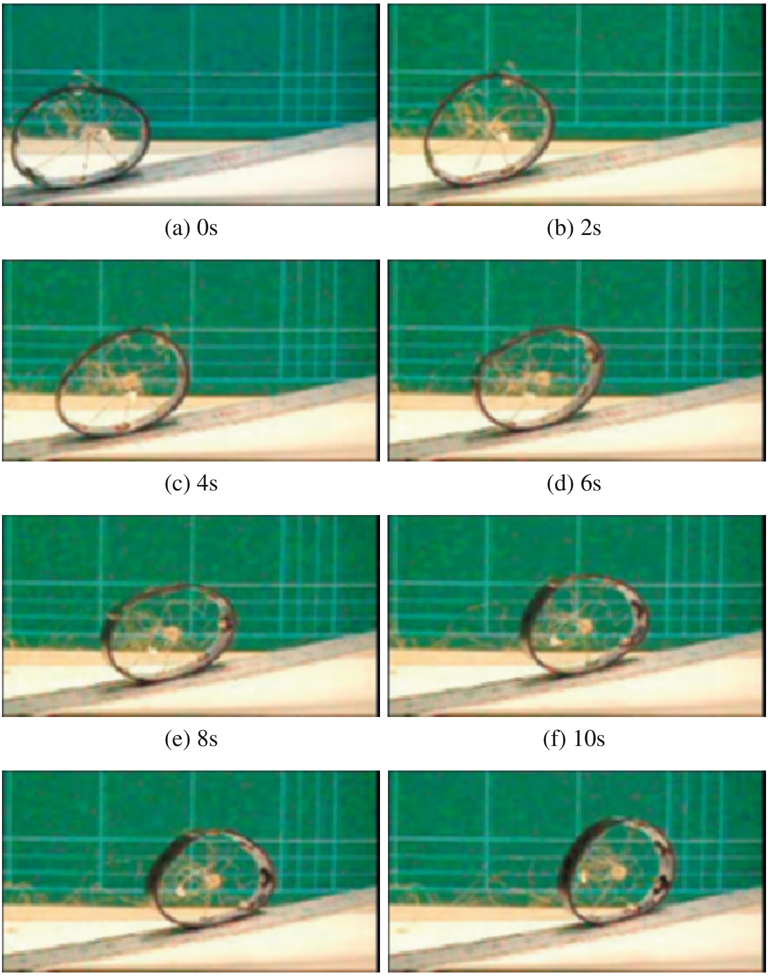


Fig. 12. Comparison between simulation and experimental results for voltage pattern BF

Figure 14 shows a sequence of snapshots of the prototype jumping. The prototype can jump a distance of 80mm, which is twice its diameter.

5 Conclusion and Research Perspective

In the present study, we proposed a deformable robot capable of crawling and jumping. First, we described the principle of crawling and jumping using the deformation of a robot body. Second, in a physical simulation using a three-element model



(a) 0s

(b) 2s

(c) 4s

(d) 6s

(e) 8s

(f) 10s

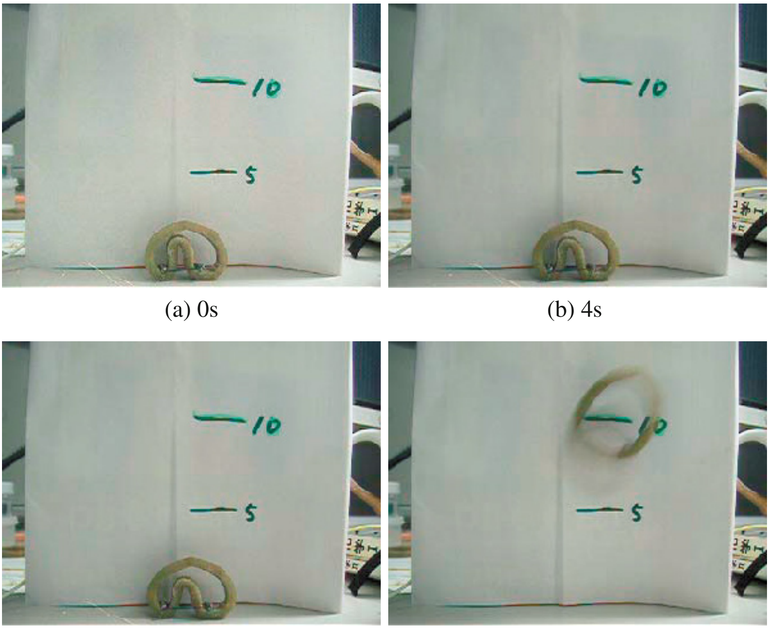
(g) 12s

(g) 16s

Fig. 13. Circular soft robot climbing a slope

with a slider, we showd the feasibility of the robot to crawl and jump. Finally, we experimentally verified that a prototype of a circular soft robot can crawl and jump.

To date, no analysis has been conducted on the motion of a circular soft robot. In further studies, we will apply linear object modeling to analyze and optimize the motion of a circular robot. We will evaluate the potential energy of a circular soft robot during crawling and jumping in order to get a better understanding of the system. We will also make a prototype of a spherical soft robot capable of performing 3D motion.



(a) 0s (b) 4s

(c) 5s (d) 6s

Fig. 14. Circular soft robot jumping

Acknowledgement

This research was supported in part by the Ritsumeikan University 21st Century COE program “Micro Nanoscience Integrated Systems”.

References

1. Saranli, U., Buehler, M., and Koditschek, D. E., *RHex: A Simple and Highly Mobile Hexapod Robot*, Int. J. of Robotics Research, Vol. 20, No. 7, pp.616–631, 2001.
2. Yim, M., Eldershaw, C., Zhang, Y., and Duff, D., *Limbless Conforming Gaits with Modular Robots*, Proc. Int. Symp. on Experimental Robotics, Singapore, June, 2004.
3. Pelrine, R., Kornbluh, R., Pei, Q., and Joseph, J., *High-speed Electrically Actuated Elastomers with Strain Greater Than 100%*, Science, Vol. 287, February, pp.836–839, 2000.
4. *The First Conference on Artificial Muscles*, December, 2001.
5. *Artificial Muscles*, Scientific American, October, pp.34–41, 2003.
6. Otake, M., Nakamura, Y., Inaba, M., and Inoue, H., *Wave-shape pattern control of electroactive polymer gel robots*, Proc. Int. Symp. on Experimental Robotics, Singapore, June, 2004.
7. Kimura, M., Sugiyama, Y., Tomokuni, S., and Hirai, S., *Constructing Rheologically Deformable Virtual Objects*, Proc. IEEE Int. Conf. on Robotics and Automation, Taipei, September, 2003.

The Berkeley Lower Extremity Exoskeleton Project

That which does not stabilize will only make us stronger

H. Kazerooni

University of California at Berkeley
Berkeley, CA 94720, USA
kazerooni@me.berkeley.edu

Abstract. In October 2003, the first functional load-bearing and energetically autonomous exoskeleton, called the Berkeley Lower Extremity Exoskeleton (BLEEX) was demonstrated, walking at the average speed of two miles per hour while carrying 75 pounds of load. The project, funded in 2000 by the Defense Advanced Research Project Agency (DARPA) tackled four fundamental technologies: the exoskeleton architectural design, a control algorithm, a body LAN to host the control algorithm, and an on-board power unit to power the actuators, sensors and the computers. This article gives an overview of the BLEEX project.

1 What is a Lower Extremity Exoskeleton?

The primary objective of this project at U.C. Berkeley is to develop the fundamental technologies associated with design and control of energetically autonomous Lower Extremity Exoskeletons that augment human strength and endurance during locomotion. The first field-operational lower extremity exoskeleton at Berkeley (commonly referred to as BLEEX) is comprised of two powered anthropomorphic legs, a power unit, and a backpack-like frame on which a variety of heavy loads can be mounted. This system provides its pilot (i.e., its wearer) the ability to carry significant loads on his/her back with minimal effort over any type of terrain. BLEEX allows the pilot to comfortably squat, bend, swing from side to side, twist, and walk on ascending and descending slopes, while also offering the ability to step over and under obstructions while carrying equipment and supplies. Because the pilot can carry significant loads for extended periods of time without reducing his/her agility, physical effectiveness increases significantly with the aid of this class of lower extremity exoskeletons. In order to address issues of field robustness and reliability, BLEEX is designed such that, in the case of power loss (e.g., from fuel exhaustion), the exoskeleton legs can be easily removed and the remainder of the device can be carried like a standard backpack.

BLEEX was first unveiled in October 2003, at U.C. Berkeley's Human Engineering and Robotics Laboratory. In this initial model, BLEEX offered a carrying capacity of seventy five pounds, with weight in excess of that allowance being supported by the pilot.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 291–301, 2006.

BLEEX's unique design offers an ergonomic, highly maneuverable, mechanically robust, lightweight, and durable outfit to surpass typical human limitations. BLEEX has numerous applications; it can provide soldiers, disaster relief workers, wildfire fighters, and other emergency personnel the ability to carry major loads such as food, rescue equipment, first-aid supplies, communications gear, and weaponry without the strain typically associated with demanding labor. It is our vision that BLEEX will provide a versatile transport platform for mission-critical equipment.

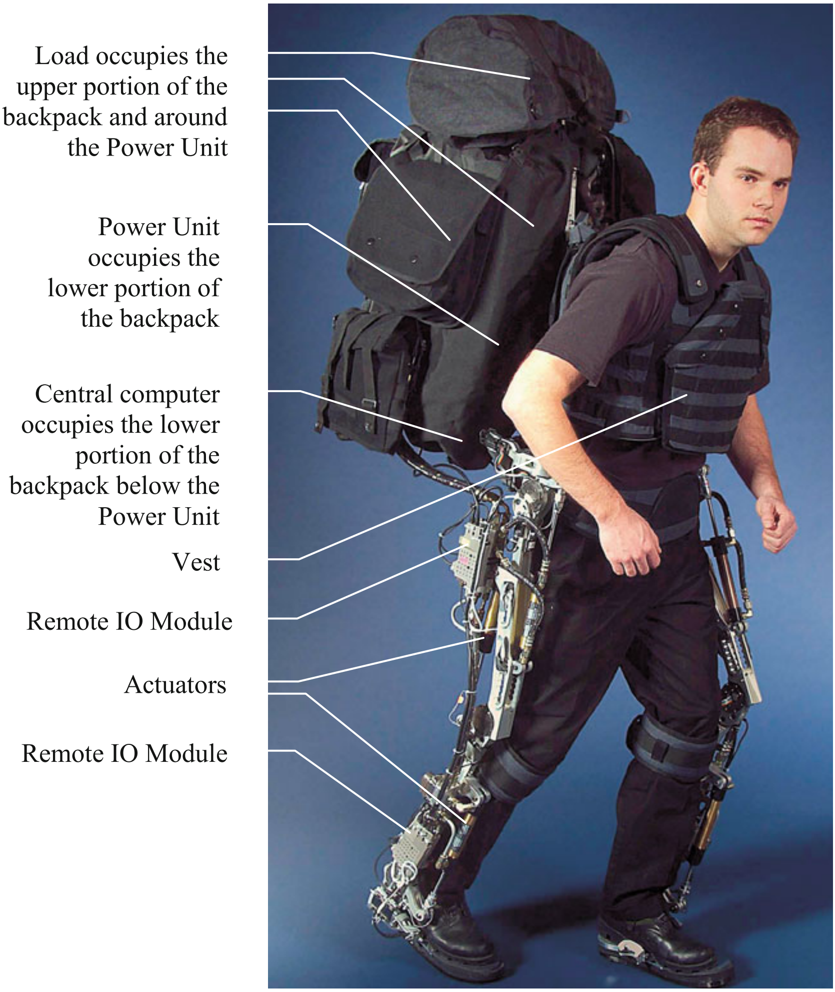

Load occupies the upper portion of the backpack and around the Power Unit
Power Unit occupies the lower portion of the backpack
Central computer occupies the lower portion of the backpack below the Power Unit
Vest
Remote IO Module
Actuators
Remote IO Module

Figure 1: Berkeley Lower Extremity Exoskeleton (BLEEX)

2 Disconnection of Lower Extremity and Upper Extremity Augmentation

Human-Robot integration, in particular human augmentation, outlines the future of robotics. Although autonomous robotic systems perform remarkably in structured environments (e.g. factories), integrated human-robotic systems are superior to any autonomous robotic systems in unstructured environments that demand significant adaptation. In our research work at Berkeley, we have separated the technology associated with human power augmentation into lower extremity exoskeletons and upper extremity exoskeletons. The reason for this was two-fold; firstly, we could envision a great many applications for either a stand-alone lower or upper extremity exoskeleton in the immediate future. Secondly, and more importantly for the division is that the exoskeletons are in their early stages, and further research still needs to be conducted to ensure that the upper extremity exoskeleton and lower extremity exoskeleton can function well independently before we can venture an attempt to integrate them. With this in mind, we proceeded with the designs of the lower and upper extremity exoskeleton separately, with little concern for the development of an integrated exoskeleton. We will first give a summary of the upper extremity exoskeleton efforts at Berkeley and then we will proceed with the description of the BLEEX project.

3 What is an Upper Extremity Exoskeleton?

In the mid-1980s, we initiated several research projects on upper extremity exoskeleton systems, billed as “human extenders [1-5].” The main function of an upper extremity exoskeleton is human power augmentation for manipulation of heavy and bulky objects. These systems, which are also known as assist devices or human power extenders, can simulate forces on a worker’s arms and torso. These forces differ from, and are usually much less than the forces needed to maneuver a load. When a worker uses an upper extremity exoskeleton to move a load, the device bears the bulk of the weight by itself, while transferring to the user as a natural feedback, a scaled-down value of the load's actual weight. For example, for every 40 pounds of weight from an object, a worker might support only 4 pounds while the device supports the remaining 36 pounds. In this fashion, the worker can still sense the load’s weight and judge his/her movements accordingly, but the force he/she feels is much smaller than what he/she would feel without the device. In another example, suppose the worker uses the device to maneuver a large, rigid, and bulky object, such as an exhaust pipe. The device will convey the force to the worker as if it was a light, single-point mass. This limits the cross-coupled and centrifugal forces that increase the difficulty of maneuvering a rigid body and can sometimes produce injurious forces on the wrist. In a third example, suppose a worker uses the device to handle a powered torque wrench. The device will decrease and filter the forces transferred from the wrench to the worker’s arm so the worker feels the low-frequency components of the wrench's vibratory forces instead of the high-frequency components that produce fatigue.

The assist devices we designed not only filter out unwanted forces on a worker, but can also be programmed to follow a particular trajectory regardless of the exact direction in which the worker attempts to manipulate the device. For example, suppose an auto-assembly worker is using an assist device to move a seat to its final destination inside a car. The assist device can bring the seat to its final destination, moving it along a preprogrammed path with a speed that is proportional to the magnitude of the worker’s force on the device. Although the worker might be paying very little attention to the final destination of the seat, the device can still bring the seat to its proper place without the worker’s guidance. The upper extremity exoskeleton reflects on the worker's arm forces that are limited and much smaller than the forces needed to maneuver loads.

This interaction between humans and machines benefits from the combination of human intellect and machine strength; the human provides a decision-making system for the assist device, while the device actuators provide most of the strength. With it, auto-assembly and warehouse workers can maneuver parts and boxes with greatly improved dexterity and precision, not to mention a marked decrease in muscle strain. Put into use, these devices will significantly reduce the incidence of back injury in the workplace, which will in turn greatly offset the annual cost of treating back injuries.

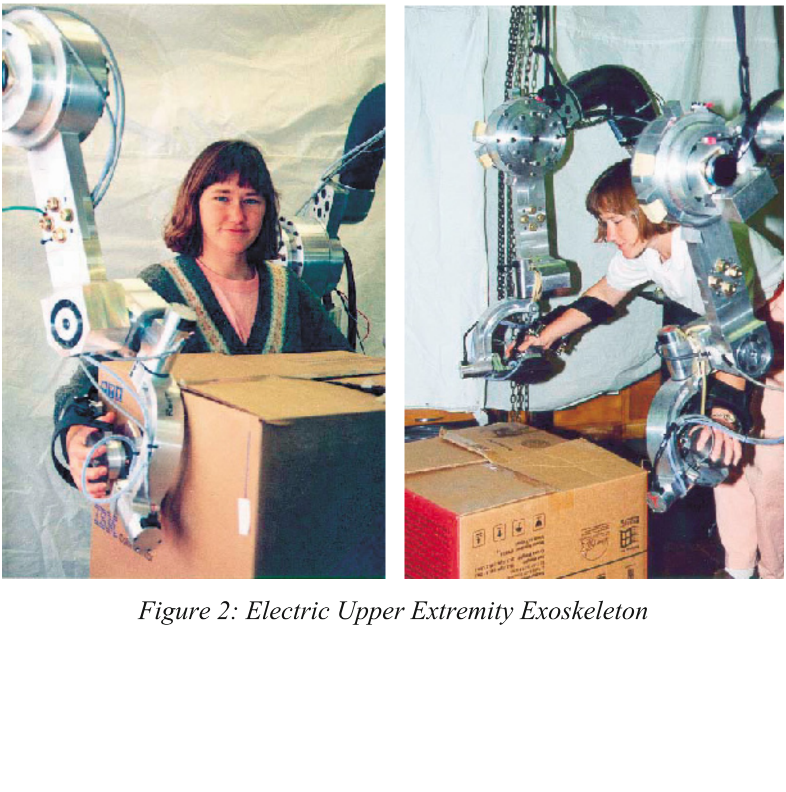


Figure 2: Electric Upper Extremity Exoskeleton

Upper extremity exoskeletons were designed based primarily on compliance control [6-8] schemes that relied on the measurement of interaction force between the human and the machine. Various experimental systems, including a hydraulic loader designed for loading aircrafts and an electric power extender built for two-handed operation, were designed to verify the theories. We also developed low-cost industrial upper extremity extenders (commonly referred to as intelligent assist devices) that are now widely used in the US and Europe. An Intelligent Assist Device includes a computer controlled electric actuator which is attached directly to a ceiling, or an overhead crane and precisely moves a wire rope with a controllable speed. Attached to the wire rope is a sensory end-effector where the operator hand, the IAD, and the load come in contact. In these assist devices, the operator force on the device is sensed and amplified electronically by use of a computer to drive the actuator. The end-effector includes a load interface subsystem which is designed to interface with a variety of loads and holding devices such as suction cups and hooks.

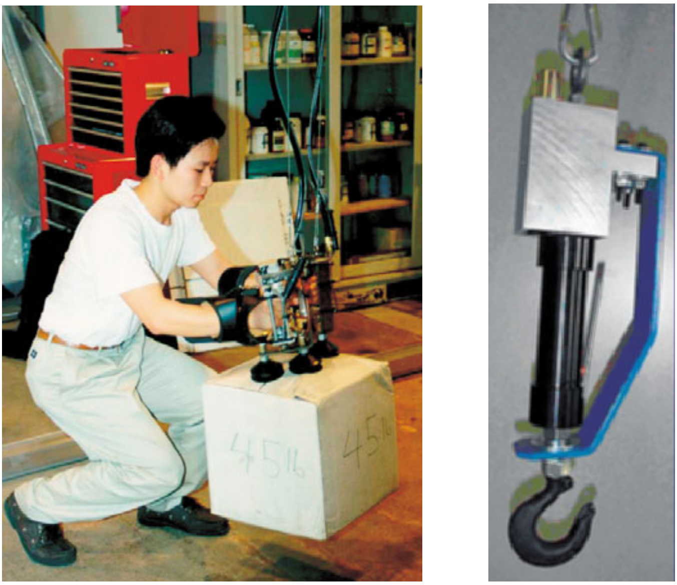


Figure 3: Intelligent Assist Devices (IAD): the simplest form of upper extremity enhancers for industrial applications

4 Highlights of the BLEEX Design

In designing BLEEX, several factors had to be considered: Firstly, the exoskeleton needed to exist in the same workspace of the pilot without interfering with his motion. Secondly, it had to be decided whether the exoskeleton should be anthropomorphic (i.e., kinematically matching), or non-anthropomorphic (i.e. kinematically matching the operator only at the connection points between human and machine). We ultimately selected the anthropomorphic architecture because of its transparency to the pilot. We also concluded that an exoskeleton that kinematically matches the wearer's legs gains the most psychological acceptance by the user and is therefore safer to wear. Consequently, the exoskeleton was designed to have the same degrees of freedom as the pilot: three degrees at the ankle and the hip, and one degree at the knee. This architecture also allowed the appropriately scaled clinical human walking data to be employed for design of the exoskeleton components, including the workspace, actuators and the power source.

A study of clinical gait analysis (CGA) data provides evidence that humans expend the most power through the sagittal plane joints of the ankle, knee, and hip while walking, squatting, climbing stairs, and most other common maneuvers. For this reason, the sagittal plane joints of the first prototype exoskeleton are powered. However, to save energy, the non-sagittal degrees of freedom at the ankle and hip remain unpowered. This compels the pilot to provide the force to maneuver the exoskeleton abduction and rotation, where the required operational forces are smaller. To further reduce the burden on the human operator, the unactuated degrees of freedom are spring-loaded to a neutral standing position. Using CGA data and human factors information, the ranges of motion for the exoskeleton were selected to be larger than those of the human while walking, and smaller than the physical limit of the human joints. This ensures sufficient flexibility for walking, while maintaining the safety of the pilot. In order to accommodate the largest number of pilots, the exoskeleton was designed to have adjustable shank and thigh sections. These sections will adjust from 5% to 95% of the shank and thigh length of men in the U.S. Army as determined from the human factors literature. At the foot, the exoskeleton rigidly attaches to the pilot's boot with a binding, but a flexible toe-section, ankle abduction, and vertical rotation axis allow the exoskeleton foot sufficient maneuverability to keep from encumbering the user.

The hips connect the legs to the torso through three degrees of freedom. Among these three degrees of freedom, only one degree in the sagital plane is powered. The exoskeleton torso is a structural member that rigidly connects to the pilot vest. The vest is designed from several hard Polycarbonate surfaces connected compliantly together to conform to the pilot's chest, shoulders and upper back, thereby preventing concentrated forces between the exoskeleton and the wearer. The torso also provides mounting points for the power supply, payload and computer.

The Clinical Gait Analysis (CGA) was used to provide not only the basis for the exoskeleton kinematics and dynamic architecture, but also for its actuation. The deployment of this CGA data was motivated by the assumption that BLEEX resembles a human's weight and volume. The BLEEX ankle actuators were designed to provide relatively large plantarflexion torques (~1000 lb-in) corresponding to those needed for propulsion at toe-off. The knee actuators were designed to provide both large extension torques (~800 lb-in) needed during heel strike, which occurs while walking, and large flexion torques (~-800 lb-in) needed during swing, which occurs with actions like climbing stairs. The hip actuators were designed to provide relatively symmetric flexion and extension torques (+/- 900 lb-in) corresponding to the symmetric nature of the torques required at the hip to walk. These critical design decisions were further reinforced by physiological observation. Since hydraulic linear actuators were employed to power the BLEEX joints, the imposed torque at each joint was a nonlinear function of the actuator location and its geometry. An optimization code was written to locate both the actuator endpoint locations and the area of the cross section of the actuators so as to yield the right amount of torque at each joint. If the actuators were designed to yield a large amount of torque more than what is prescribed by CGA data at each joint, then there would be a great amount of power loss during modulation (e.g. control).

CGA data, which provided torque and speed information at each joint of a 165 pound person, was also used to size the exoskeleton power source. The information suggested that a typical person uses about 0.25 HP (185 Watts) to walk at average speed of 3 mph. This figure, which represents the average product of speed and torque, is an expression of the purely mechanical power exhibited at the legs during walking. Since we assumed that the exoskeleton is similar to a human in terms of geometry and weight, one of the key design objectives turned out to be designing a power unit and actuation system to deliver about 0.25 HP at the exoskeleton joints.

5 BLEEX Electronics

BLEEX uses a multivariable nonlinear algorithm to robustly control its behavior. The system has many degrees of freedom with a great number of sensors and actuators. Since all computations required to implement the control are conducted on a single computer, we needed a control platform to minimize the number of signal wires in the system. The exoskeleton electronics system, EXOLINK, was designed to simplify and reduce the cabling task of all the sensors and actuators needed for exoskeleton control. It relies on a high-speed synchronous ring network topology where several electronic Remote Input Output Modules (RIOM) reside in a ring. Each RIOM is in communication with several sensors and one actuator in close proximity, and includes eight sixteen-bit Analog-to-Digital Converters (ADC), two quadrature counters, eight bits of digital input and output ports, two Digital-to-Analogue converters (DAC) and analog filters. Each RIOM also includes localized power regulation and isolation to minimize signal noise and system ground loops while a built-in FPGA manages all RIOM data transaction and filtering. The data gathered by each module is encoded and transmitted digitally to a central computer through the ring.

The EXOLINK has four rings, two of which are associated with the two legs and include three Remote Input Output Modules. A third ring is connected to a Graphical User Interface for debugging and data acquisition, and a fourth ring is used to accommodate other electronic and communication gears that are not related to the exoskeleton, but which the pilot must carry. Each ring is equipped to accommodate up to eight RIOMs.

The EXOLINK consists of a microcomputer and a Supervisor IO Module (SIOM). The SIOM includes a FPGA programmed to serve as the communication hub for all four rings. A transceiver chip residing in the SIOM and all the RIOMs allows for data transfer at a rate of 1500 Mb/s. Currently, a 650 MHz Pentium PC-104 microcomputer is used to implement the control algorithm, and the current Exoskeleton utilizes 75% of the I/O capability of the EXOLINK. The use of a high-speed synchronous network in place of the traditional parallel method enables the exoskeleton to reduce the over 200 sensor and actuator wires to only 24 communication and power wires. While the sensors are read at the rate of 10KHtz, the control is updated at 4KHtz (Control sampling time is 250 micro-seconds).

6 BLEEX Control

The effectiveness of the lower extremity exoskeleton stems from the combined benefit of the human intellect provided by the user and the strength advantage offered by the exoskeleton. The human provides an intelligent control system for the exoskeleton, while the exoskeleton actuators provide most of the strength necessary for walking. The control algorithm ensures that the exoskeleton moves in concert with the pilot with minimal interaction force between the two. The control scheme needs no direct measurements from the user or the human-machine interface (e.g. the force sensors between the two); instead, the controller estimates, based on measurements from the exoskeleton only, how to move so that the wearer feels very little force. This novel control scheme, which has never before been applied to any robotic system, is a complex but effective method of generating locomotion when the contact location between the wearer and the machine is unknown and unpredictable (i.e. the exoskeleton and the wearer are in contact in variety of places). This control method differs from compliance control methods employed for upper extremity exoskeletons because it requires no force sensor between the wearer and the exoskeleton.

The basic principle for the control of BLEEX rests on the notion that the exoskeleton needs to shadow the wearer's voluntary and involuntary movements quickly, and without delay. This means that the exoskeleton requires a high level of sensitivity in response to all forces and torques on the exoskeleton, in particular the forces imposed from the wearer. Addressing this need involves a direct conflict with control science's goal of minimizing system sensitivity in the design of a closed loop feedback system, but the exoskeleton is a system that requires considerable sensitivity to external forces and torques to perform its function properly; if fitted with a low sensitivity, the system will not move in concert with its wearer. We realize, however, that maximizing system sensitivity to external forces and torques leads to a loss of robustness in the system.

Taking into account this new approach, our goal was to develop a control system for BLEEX with high sensitivity. We were faced with two realistic concerns associated with this approach; the first was that an exoskeleton with high sensitivity to external forces would respond to other external forces that are not initiated by its pilot. For example if someone pushed against an exoskeleton that had high sensitivity, the exoskeleton would move just like the way it would move in response to the forces from its pilot. Although this may sound like a serious problem, the fact is we want to ensure that the exoskeleton does not stabilize its behavior on its own in response to other forces. If it did, the pilot would receive motion from the exoskeleton unexpectedly and would have to struggle with it to avoid unwanted movement. The key to stabilizing the exoskeleton and preventing it from falling in response to external forces depends on the pilot's ability to move quickly (e.g. step back or sideways) to create a stable situation for himself and the exoskeleton. For this, a very large control bandwidth is needed so the exoskeleton can respond to both voluntary and involuntary movements (reflexes).

The second concern is that systems with high sensitivity to external forces and torques are not robust to variations and therefore the precision of the system performance will be proportional to the precision of the exoskeleton dynamic model. This is quite a serious drawback and we have accepted it since various experimental systems in our laboratory proved the effectiveness of the control method in shadowing the pilot's movement.

One class of systems that has large sensitivity is marginally stable systems. We therefore designed a marginally stable closed loop controller system that uses the sensor information on the exoskeleton only. This is done by using the inverse of the dynamics of the exoskeleton as a positive feedback controller so the loop gain for the exoskeleton approaches unity (slightly less than 1). Obviously, to get this method working properly, one needs to understand the dynamics of the exoskeleton quite well, as the controller is heavily model based. Our experiments with BLEEX have shown that at this time, this control scheme— which does not stabilize BLEEX— forces it to follow human wide bandwidth maneuvers while carrying heavy loads although it requires a great deal of dynamic modeling. We have come to believe that that which does not stabilize, will only make us stronger.

7 BLEEX Power Unit

One of the greatest hurdles to overcome in achieving energetic autonomy with mobile robots is the power supply. Every effort has been made to ensure that BLEEX is energetically autonomous and field re-fuelable. Current mobile robotic devices typically use a tether attaching the robot to a large stationary power supply, or they carry large numbers of batteries for relatively short operation times. A significant challenge in designing the Berkeley lower extremity exoskeleton was the development of a power supply and actuation system that would satisfy its power and energy requirements for an extended amount of time.

We view the exoskeleton as a mobile fieldable platform like a vehicle or motorcycle, rather than an indoor industrial robot with a power cord to energize its limbs. This paradigm shift on exoskeleton design forces us to confront a set of design questions (including the power source) at the initial stages of design. Hydrocarbon fuels in the form of gasoline or diesel fuel are the most suitable form of energy for mobile platforms, due to their large specific energy (45 Mega Jules per Kilogram). Electric motor actuators, common in industrial robotics due to their simplicity and convenience, have a low power density, resulting in heavy and bulky actuation systems that require large power sources for long term missions.

BLEEX uses a specially-designed Hydraulic Power Unit (HPU), which delivers hydraulic power for locomotion and electrical power for the computing components and sensors. This hybrid power system is capable of fueling an otherwise unpowered human scale mobile robotic system for a period of hours, rather than minutes, and combines the high specific energy of hydrocarbon fuels with the high specific power of hydraulic actuator systems. According to CGA data, we found that a 165 pound person uses 0.25 HP of mechanical power to walk with the speed of 3 MPH. The key issue is to design a power and actuation system to deliver about 0.25 HP at the exoskeleton joints, assuming the exoskeleton resembles a 165 lb. person. The power unit should also provide about 150W of electrical power to fuel the electronic systems, including the computer, RIOMs and sensors. According to a detailed set of calculations, it was concluded that an exoskeleton that resembles a 165 lb. person and uses hydraulic for actuation needs a 1.6 HP hydraulic power supply (2.7GPM of hydraulic flow at 1000 psi) to walk the exoskeleton at the speed of 3MPH. We designed two HPUs to accommodate two different power sources for the exoskeleton.

The first design consists of an air-cooled four-stroke single cylinder engine coupled to a three-phase brushless generator and a hydraulic pump for powering electronics as well as hydraulic actuators. This HPU was capable of producing a maximum of 1.4 HP (1.04 KW) of hydraulic power at 1000 psi (6.9 MPa) operating pressure (i.e., 2.4 GPM of hydraulic flow), as well as approximately 200 W of electrical power. We carried out a set of experiments with the exo and we learned that this HPU produces 1.08 HP and 1.4 HP at walking speeds of 1.3 MPH and 1.9 MPH. The actual mechanical power used for two locomotion speeds were 0.24 HP and 0.31 HP, compared to 0.25 HP for a human walking at the speed of 3 MPH. We were never able to walk with the speed of 3 MPH with the BLEEX as the power unit saturated; the inadequate power (i.e. inadequate flow since pressure was regulated to 1000 psi) of this HPU limits the walking speed of the exoskeleton to 1.9 MPH. Although, the power draw from the exoskeleton power unit was in the same order of magnitude of the estimated one, the discrepancy was due to the fact that BLEEX's stature turned out not to be exactly like that of a person in term of weight and geometry. For example the load on the exoskeleton's back made it rather dissimilar to a person. We also realized that a one-gallon fuel tank provides approximately 4 hours of run time, which results in approximately 8 miles of travel distance. This long distance for a mission could not be achieved using batteries.

For faster walking (not longer mission times), we designed a HPU with more power. This HPU consists of a liquid-cooled, two-stroke opposed cylinder gasoline engine directly coupled to a three-phase brushless generator and a hydraulic gear pump for powering electronics as well as hydraulic actuators. In order to minimize mass and facilitate tight packaging with limited cooling airflow, the HPU is liquid-cooled with the same hydraulic fluid that the actuators utilize. This HPU successfully produces 3HP (2.24 KW) of hydraulic power at a 1000 psi (6.9 MPa) operating pressure (i.e., 5.2 GPM of hydraulic flow), in addition to 300 W of electrical power. The larger flow of this HPU theoretically allows the walking speed of the exoskeleton to exceed 4 MPH, but this speed is in practice limited by control bandwidth. A one-gallon fuel tank provides approximately 1 hour and 15 minutes of mission time at its maximum power (5.2 GPM at 1000 psi), which translates into 5 miles of travel in one hour. We concluded that this second HPU decreases the mission time by half for a given distance and payload.

The controllers of the HPUs regulate two variables: the hydraulic pressure and the engine speed employing two inputs: engine throttle and a bypass hydraulic valve. The hydraulic pressure was regulated at 1000 psi and the speed was regulated at a speed that the hydraulic pump or the engine had their highest efficiency and peak power (7000 RPM for the first HPU and 6500RPM for the second HPU). The constant engine speed also led to the design of an optimized muffler. The noise level produced by both HPUs is minimized using an exhaust system and integrated baffling incorporated in the system packaging.

References

[1] Kazerooni, H., "Human-Robot Interaction via the Transfer of Power and Information Signals," IEEE Trans. on Systems and Cybernetics, V. 20, No. 2, Mar. 1990.

[2] Kazerooni, H., and Mahoney, S., "Dynamics and Control of Robotic Systems Worn By Humans," ASME Journal of Dynamic Systems, Measurements, and Control, Vol. 113, No. 3, pp. 379-387, September 1991.

[3] Kazerooni, H., "The extender technology at the University of California, Berkeley," Journal of the Society of Instrument and Control Engineers in Japan, Vol. 34, 1995, pp. 291-298.

[4] Kazerooni, H., “The Human Power Amplifier Technology at the University of California, Berkeley”, *Journal of Robotics and Autonomous Systems*, Elsevier, Volume 19, 1996, pp. 179-187.

[5] Kazerooni, H., Guo, J., "Human Extenders," ASME Journal of Dynamic Systems, Measurements, and Control, Vol. 115, No. 2(B), June 1993.

[6] Kazerooni , H., Snyder, T. J., "A Case Study on Dynamics of Haptic Devices: Human Induced Instability in Powered Hand Controllers," AIAA J. of Guidance, Control, and Dynamics, V. 18, N1, 1995.

[7] Kazerooni, H., Houpt, P. K., and Sheridan, T. B., "A Design Method for Robust Compliant Motion of Manipulators," *IEEE Journal of Robotics and Automation*, Vol. 2, No. 2, June 1986.

[8] Kazerooni, H., and Waibel, B., "On the Stability of the Constrained Robotic Maneuvers" *IEEE Trans. on Robotics and Automation*, V7 No. 1. Feb.1991.

Toward Self-Replication of Robot Control Circuitry by Self-Inspection

Jackrit Suthakorn[1] and Gregory S. Chirikjian[2]

[1] Mahidol University
Department of Mechanical Engineering
25/25 Puttamonton 4 Road
Salaya, Nakorn Patom 73170 THAILAND
egjst@mahidol.ac.th

[2] Johns Hopkins University
Department of Mechanical Engineering
3400 North Charles Street
Baltimore, Maryland 21218 USA
gregc@jhu.edu

Abstract. The concept of man-made self-replicating machines was first proposed by John von Neumann more than 50 years ago. However, there has never been a physical implementation of his universal constructor architecture as a robotic system. Prior to our other recent work, an autonomous self-replicating mechanical system had not been developed. In this paper, we demonstrate a non-von-Neumann architecture for the replication of transistor circuits by active self-inspection. That is, there are no instructions stored about how to construct the circuit, but information observed about the spatial organization of the original circuit drives a larger electromechanical (robotic) system in which it is embedded to cause the production of a replica of the original circuit. In the work presented here, only replication of the control circuit is of interest. In the current context, the electromechanical hardware is viewed as a tool which is manipulated by the control circuit for its own reproduction. This architectural paradigm is demonstrated with prototypes that are reviewed here and compared with an implementation of the universal constructor concept.

1 Introduction

1.1 Motivation

The concept of man-made self-replicating machines was first proposed by John von Neumann more than fifty years ago [1], and this led to a flurry of related works [2, 3]. However, there has never been a physical implementation of his universal constructor architecture in a robotic system. In contrast there have been a number of implementations of self-assembling mechanical systems [4, 5]. Prior

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 303–309, 2006.

to our other recent work (see [6]), an autonomous self-replicating mechanical system had not been developed. In contrast to passive self-assembly, a self-replicating system actively utilizes an original unit to assemble a copy of itself from a collection of passive components. However, this does not require the use of von Neumann's universal constructor architecture. In the present work, we demonstrate a non-von-Neumann architecture for the replication of a transistor circuit by active self-inspection. That is, there are no instructions stored about how to construct the circuit, but information observed about the spatial organization of the original circuit is fed into the circuit itself to provide assembly commands. The circuit then drives a larger electromechanical (robotic) system in which it is embedded to cause the production of a replica of the original circuit. In the work presented here, only replication of the control circuit is of interest. In the current context, the electromechanical hardware is viewed as a tool which is manipulated by the control circuit for its own reproduction (much in the same way that deer living in a forest can reproduce without an associated reproduction of the forest itself). This architectural paradigm is demonstrated with prototypes that are reviewed here and compared with an implementation of the universal constructor concept.

1.2 Related Works

This work complements our recent work in which we have demonstrated various aspects of robotic self-replication with a series of prototypes including: (1) remote-controlled systems capable of assembling copies of an original robot from subsystems [7]; (2) a semi-autonomous system in which a remote-controlled robot builds fixtures which then autonomously assist in assembling a copy of the original robot [8]; (3) an autonomous self-replicating mechanical system (in which the computer program for the replica is pre-installed) which functions without human intervention [6]. The present work differs substantially from those works because it focuses on the replication of the “brains” of simple robotic systems from individual transistors rather than treating the microprocessor as a preconstructed subsystem in the assembly process.

Over the years, the concept of self-replicating robotic systems has been considered to be useful in many applications, especially, space applications. Many researchers have discussed the possibilities of using such a system in space and planetary exploration [9, 10, 11]. In order to make this vision realistic, one must show that self-replication of intelligence (SRI) is possible. We believe that the non-von-Neumann concept ofself-replication by self-inspection, first addressed theoretically by Burks [13], Arbib [12], and Liang [14] (and implemented in electromechanical systems for the first time here) is a paradigm which is very robust and worthy of consideration.

In the next section, our approach to self-replicating intelligence is described. Section 3 describes demonstrations developed by our students and their results, and Section 4 presents our discussions.

2 Our Approach on Self-Replicating Control Circuitry

Ideally, for a robotic system to be truly self-replicating, it would have to demonstrate the ability to assemble all of its own subsystems from the most fundamental components. In the case of the robot controller, we consider the most fundamental components to be transistors, resistors, capacitors, etc., whereas microcontrollers are too complex to be considered as basic elements.

Our approach is to build a circuit capable of controlling an electro-mechanical system to re-build replicas of the control circuit from the most fundamental electronic components. In the von Neumann universal constructor paradigm, an associated instruction code is also required. In contrast it is possible to replicate a particular system by self-inspection without invoking von Neumann's universal constructor. We illustrate both concepts in hardware designed and constructed by students in a Mechatronics course taught at Johns Hopkins University in 2003. Two prototypes illustrate replication by self-inspection, and one demonstrates the universal constructor. In all three cases, pre-built electro-mechanical systems (called the SRI-builders) use the transistorized circuit as its controller. While in the von Neumann paradigm, the controller follows instructions that are explicitly encoded (and hence must reproduce the code for the overall system to be self-replicating), in the self-inspection paradigm, actions are taken implicitly as a result of observing the spatial layout of components in the original and feeding that information into the circuit itself. Clever electromechanical design ensures that observations obtained during self-inspection are translated directly into actions without requiring the interpretive step of consulting a long sequence of encoded construction commands.

3 Experiments and Results

3.1 A von Neumann Universal Constructor Prototype

The robotic system is a two-degree of freedom gantry-style robot, consisting of two arms separated by a constant offset. One degree of freedom is the position along the entire system that includes the feeders and the assembly boards. The other is the vertical direction, perpendicular to the system foundation, used by the arm to pick up and drop off code and circuit pieces. The position and arm commands received by the control circuit are carried out and controlled by switches designed and placed to be operated at the completion of each movement. There are two boards being assembled at any given time, one for the circuit and one for the code. The circuit board is pre-wired so that when the circuit blocks are dropped into place, all of the chip connections are, in theory, instantly made. Each line of code consists of three bits fed simultaneously to a reader array consisting of three optical sensors that detect the value of the bits below – black being zero, white being one. These photo sensor cells are coupled with infrared LED emitters

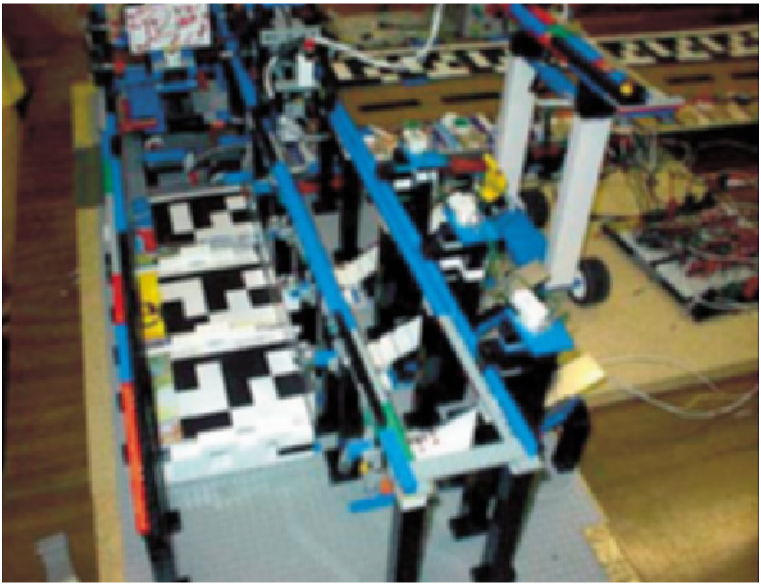


Fig. 1 Side View of the Replicating system (Von Neumann Universal Constructor Prototype.)

that provide IR light to be sensed by the sensors; ambient visible light has the potential of providing sufficient light energy to reflect off the code and be detected by the sensors, but the IR emitters guarantee the readability of the code. See Figure 1.

3.2 Non-universal self-replication by self-inspection (design 1)

This self-replicating control circuit has the ability to identify the proper electronic components required, translate information about its own constituent parts obtained from self-inspection into mechanical tasks that create a replica , and transfer all functions to the replica. There is no list of instructions in the form of a code. Each electronic component has a black-and-white color code. Parts are loaded into feeders, and as a reading head traverses the control circuit, the information about which part of the control circuit is being observed is fed into the circuit itself. This actuates the solenoid in the appropriate feeder to release the parts needed to form the replica. Parts then slide down an incline and form an orderly array. The reading head continues to move and creates replicas until resources are completely utilized or its track ends. The design is scalable and the components are modular, allowing many different levels of intelligence to be replicated. This concept is one of many which we are investigating to enable self-replicating robots to perform complex behaviors. See Figure 2.

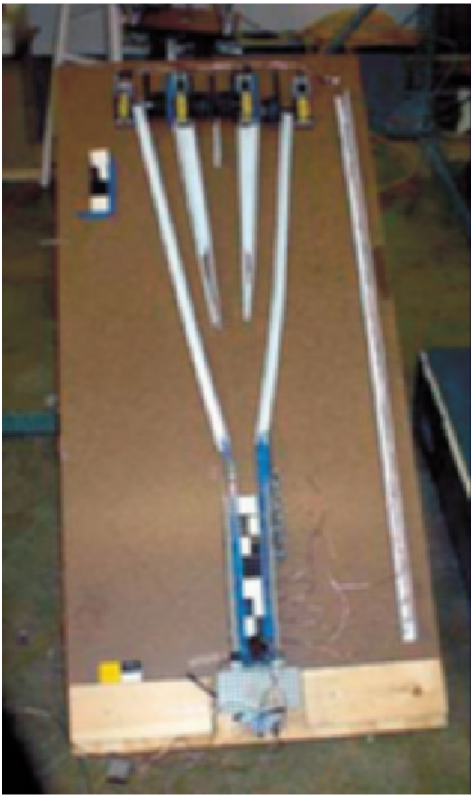


Fig. 2 Side View of the Replicating system (Non-Universal Self-Replication by Self-Inspection – Design 1.)

3.3 Non-universal self-replication by self-inspection (design 2)

This robotic system is an X-Y table constructed from modified LEGO components. A photo-transistor sensor system is attached to the end-effector of the X-Y system in order to inspect the control circuit (the components of which are each assigned a unique black and white code). On the top of the X-Y system, a set of component feeders is installed. The circuit converts the signal from the sensor system to control the component feeders to release the correct component to the parts assembler. The parts assembler then arranges all the components to create a new replica of the control circuit. See Figure 3.

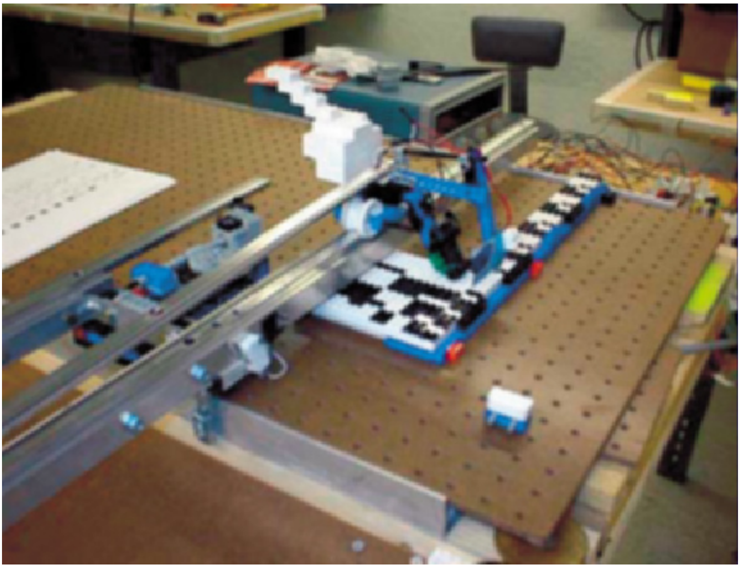


Fig. 3 Side View of the Replicating system (Non-Universal Self-Replication by Self-Inspection – Design 2.)

4 Discussion

Whereas von Neumann's architecture for self-replicating kinematic automata is the most widely known approach, it is not the only one. Self-reproduction by self-inspection in which a non-universal constructor ‘reads’ an original device and ‘writes’ a copy by executing a very small set of hardwired commands is an alternative. In our experience observing students attempting to build self-replicating devices, self-replication by self-inspection appears to be a more robust and less complicated alternative to the universal constructor.

Acknowledgements

The hardware described here was developed by the students in a Mechatronics course offered in the Department of Mechanical Engineering at The Johns Hopkins University. This work was funded by the department and a phase I NASA/NIAC award. The hard work and creative efforts of the following students are gratefully acknowledged: **Group 1:** Jin Seob Kim, Art Rivera, Danielle Soya, Danny Dokko, Landon Unninayar, Mohammed Ramadon, Robynn Denzene, Zain Syed; **Group 2:** Whitney A. Hastings, Mike Labarre, Anand Viswanathan, Stephen Lee, David Sparks, Tony Tran, Jason Nolin, Rob Curry, Michael David, Stanley Huang; **Group 3:** Keenan Wyrobek, Jesse Theiss, Jeff McDonald, Ezel

Baltali, Eric Dorflinger, Austin Moyer, Michael Comeau, Ryan Lavender, C.J. Pawlowski, Patrick Danaher

References

[1] von Neumann, J., "The General and Logical Theory of Automata," in Cerebral Mechanisms in Behavior, Proc. Hixon Symp., L.A. Jeffress, ed., pp. 1-31, Wiley, New York, 1951.

[2] Kemeny, J.G., "Man viewed as a machine," Scientific American, 192, pp. 58-67, 1955.

[3] Moore, E.F., "Artificial Living Plants," Scientific American, 195, pp. 118-126, 1956.

[4] Jacobson, H., "On Models of Reproduction," American Scientist, 46, pp. 255-281, 1958.

[5] Penrose, L.S., "Mechanics of Self-Reproduction," Annals of Human Genetics, 23, pp. 59-72, 1958.

[6] Suthakorn, J., Cushing, A.B., and Chirikjian, G.S., "An Autonomous Self-Replicating Robotic System, *Proc. of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronic (AIM)*, Kobe, Japan, 2003.

[7] Chirikjian, G.S., Suthakorn, J., "Toward Self-Replicating Robots," *Proceedings of the 8th International Symposium on Experimental Robotics (ISER) 2002*, Italy, 2002.

[8] Suthakorn, J., Kwon, Y.T., and Chirikjian, G.S., "A Semi-Autonomous Replication System," *Proceedings of the 2003 IEEE International Symposium on Computational Intelligent in Robotics and Automation (CIRA)*, Kobe, Japan, 2003.

[9] Freitas, R.A., Jr., "Report on the NASA/ASEE Summer Study on Advanced Automation for Space Missions", *J. British Interplanetary Society*, Vol. 34, 1980, pp 139-142.

[10] Chirikjian, G.S., Zhou, Y., Suthakorn, J., "Self Replicating Robots for Lunar Development," *IEEE/ASME Transactions on Mechatronics, Special Issue on Self-Reconfigurable Robots*, Vol. 7: Issue 4, 2002.

[11] Suthakorn, J., Zhou, Y., and Chirikjian, G.S., "Self-Replicating Robots for Space Resource Utilization," *Proceedings of the 2002 RoboSphere: Self-Sustaining Systems*, NASA Ames Research Center, California, 2002.

[12] Burks, A.W., "Computation, behavior and structure in fixed and growing automata," Behavioral Science, 6, 5-22, 1961.

[13] Arbib, M.A., "Simple Self-Reproducing Universal Automata," Information and Control, 9, 177-189, 1966.

[14] Laing, R.A., "Automaton Models of Reproduction by Self-Inspection," Journal of Theoretical Biology, 66, pp. 437-456, 1977.

[15] Suthakorn, J., "Paradigm for Service Robotics – Section 3: Self-Replicating Robots," *PhD Dissertation, Department of Mechanical Engineering, Johns Hopkins University*, 2003.

Optimal Mechanical/Control Design for Safe and Fast Robotics

Giovanni Tonietti, Riccardo Schiavi, and Antonio Bicchi

Interdepartmental Research Center “E. Piaggio”
Faculty of Engineering, University of Pisa
via Diotisalvi 2, 56100 Pisa, Italy
<g.tonietti, bicchi>@ing.unipi.it
http://www.piaggio.ccii.unipi.it

Abstract. The problem to ensure safety of performant robot arms during task execution was previously investigated by authors in [1], [2]. The problem can be approached by studying an optimal control policy, the “Safe Brachistocrone”, whose solutions are joint impedance trajectories coordinated with desired joint velocities. Transmission stiffness is chosen so as to achieve minimum–time task execution for the robot, while guaranteeing an intrinsic safety level in case of an unexpected collision between a link of the arm and a human operator. In this paper we extend this approach to more general classes of robot actuation systems, whereby other impedance parameters beside stiffness (such as e.g. joint damping and/or plasticity) can vary. We report on a rather extensive experimental campaign validating the proposed approach.

1 Introduction

In this paper we investigate the optimal design of mechanisms and controllers for safe and performant robotics, and propose an innovative solution based on mechanical actuator-transmission systems, that can vary their impedance parameters continuously during motion. In [1], [2] the authors introduced the idea of using a transmission system with varying stiffness, as a means of increasing the performance of the mechanism while satisfying safety constraints, and compared it with other existing approaches for guaranteeing safety and performance (e.g. [3]) by highligthing related potentialities and drawbacks. It should be pointed out that, while several mechanisms have been proposed in the robotics literature that can change transmission stiffness to adapt to different tasks (see e.g. [4], [5], [6]) the originality of our approach relies in dynamically controlling transmission characteristics within a single task.

The aim of this paper is to build upon the concept of variable stiffness and propose a more general class of Variable Impedance Actuators (VIA). Section 2 refers intuitively to the concept of Variable Impedance Approach. A brief highlight of Variable Damping and Variable Stiffness transmissions is reported respectively in section 3, and 4. Experimental results are reported showing the effectiveness of VIA in guaranteeing safety and performance during task execution.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 311–320, 2006.

2 Variable Impedance Approach for Guaranteed Safety and Performance

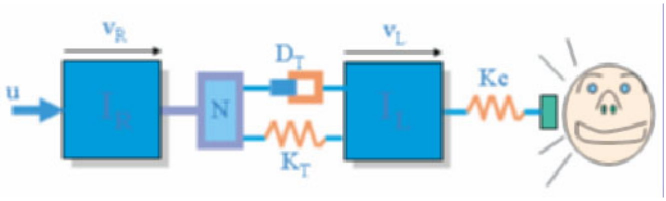


Fig. 1. General design of the coupling between torque source and link for a manipulator impacting with human. N denotes the reduction ratio, I_R the axial rotor inertia; I_L is the link inertia at the impact point; K_c is the effective cover stiffness.

Consider the simple model in fig.1, describing a robot arm impacting with an operator, where D_T, K_T are the damping and stiffness coefficients, respectively. Also let N denote the transmission gear ratio. The mechanical impedance from the impact force F to link velocity (relative to operator) v_L, $Z_m = \frac{F(s)}{v_L(s)}$ is given for this system by

$$Z_m = s\frac{(I_R N^2)I_L s^2 + [(I_R N^2) + I_L](D_T s + K_T)}{(I_R N^2)s^2 + D_T s + K_T}.$$

The positive effect of small values of impedance parameters on safety is illustrated in fig.2. On the other hand, it can be expected that small impedance values affect negatively performance, by reducing the mechanical bandwidth of the transmission (see [1]). The method we propose to overcome this limit consists in dynamically varying the impedance parameters allowing fast task executions without affecting the safety level. The optimization method adopted to obtain the shape in which these parameters are to be varied with respect to a desired motion profile comply with what reported in [1] in case of a compliant transmission (see also fig.3), therefore it is not discussed in this paper.

The variable impedance actuation approach can be implemented acting on three different parameters, i.e. effective inertia (by e.g. changing the reductiona ratio), damping, and compliance. Although the three parameters could in principle be varied simultaneously, we will explore in the next paragraphs only the variations of a single parameter at a time.

3 Variable Impedance Design

While in the previous section we introduced the concept of variable impedance as an effective means of dealing with the safety/performance trade-off, in this section we will review some examples of VIA mechanical implementations, so as to provide some background and directions to explore for the realization of novel intrinsically safe, efficient and compact actuation mechanisms for robotics.

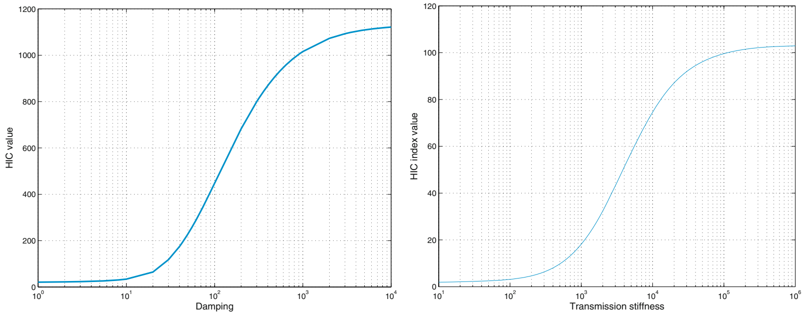


Fig. 2. Variation of the injury coefficient HIC (see e.g. [7], [1], [3]) for system in fig.1 with respect to impedance parameters, at constant link velocity $v = 2\,[m/s]$, and rotor and link inertias $M_{rot} = M_{link} = 1\,[Kg]$. As expected, in case of impact the injury risk increases with the coupling between the rotor and link.

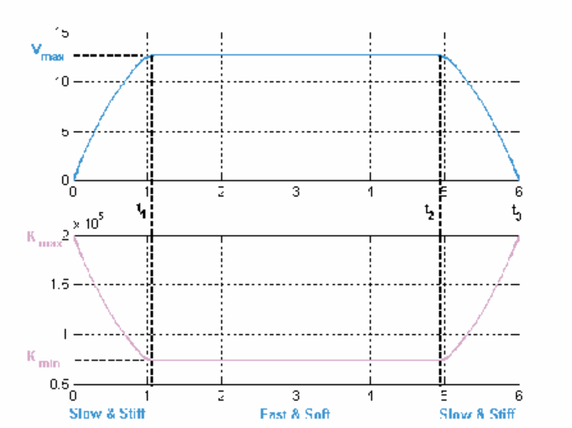


Fig. 3. Illustrating the intuitive behaviour of a Variable Stiffness Transmission in a 1DOF rest–to–rest task. High impedance is imposed at low velocities, while low impedance is used at high velocities to reduce potential impact injuries.

3.1 Variable Gear Ratio and Damping Transmissions

In order to change continuously the effective inertia $I_R N^2$, one could directly employ a variety of existing CVT (Continous Variable Transmission, [8]) mechanisms, many of which are readily commercially available. On the other hand, several possibilities for implementing actuators using the working principles of Magneto-Rheological (MR), Electro-Rheological (ER) Fluids, or Magnetic Particle Clutches (MPC's), are widely discussed in literature (see e.g. [9]). To implement a variable damping transmission, one could for example simply interpose a MPC, which is electrically controlled by a current I, between an actuator (such as an electric motor), and the

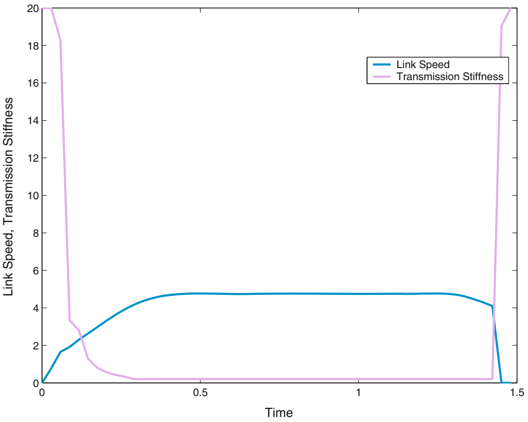


Fig. 4. The solution found with the Safe Brachistochrone algorithm for VDA. The minimum time optimal control sets high values of damping (i.e. impedance) for small values of speed and vice versa.

joint. In this manner, as the current increases, the damping factor $D(I)$ between the main shaft and joint velocities changes continuously (albeit nonlinearly) from a minimum to a maximum coupling. In the early stage of the VDT control design we adopted the Safe Brachistochrone to obtain results which can be used as a starting point for controlling the transmission damping in case of safety–oriented and perfomant manipulators (see fig. 4). As a matter of fact, results obtained by simulation comply with intuition in fig.3. Although actual implementation of VIA by either variable gear ratio or damping can be readily conceived by the above or other devices, preliminary experimentation has shown some limitations of commercially available devices (CVT's and MPC's), with particular regard to response time and nonlinearities, which hindered so far further experimenal validation. While the investigation is continuing on these solutions, in the following we provide more details on devices implementing continuous adaptation of impedance through variable stiffness transmissions (VST).

3.2 Variable Stiffness Transmission

A direct way to implement a Variable Stiffness Transmission (VST) is to design it with antagonistic configuration of nonlinear actuators, such as e.g. by interposing nonlinear spring–like mechanisms between at least two motors and the actuated joint [10]. In fact, a very important characteristic of these transmissions is that it is relatively simple to vary independently the equilibrium position q of joint shaft and transmission stiffness $\sigma(q, \theta_1, \theta_2)$ by suitably controlling the positions θ_1, θ_2 of motor shafts (see fig.5). Practically speaking, it can be highlighted that motions $\delta\theta_1 = -\delta\theta_2 = \delta\theta_\sigma$ of the motor shafts generate stiffness variations $\delta\sigma = \sigma(\delta\theta_\sigma)$,

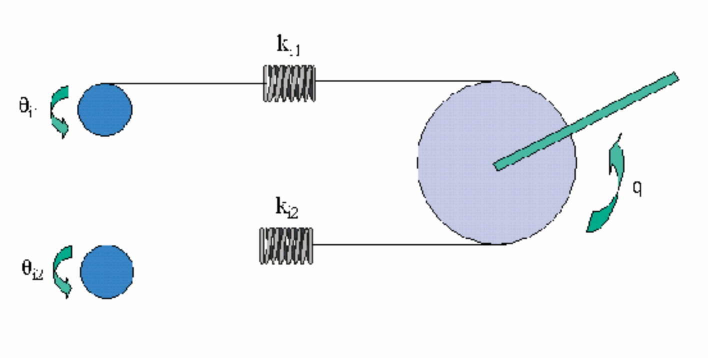


Fig. 5. Appearance of an antagonistic VST actuation system for the $i-th$ joint of a robot arm. The action of the two motors θ_{i1}, θ_{i2} generates a compression or a decompression of springs with nonlinear elasticities k_{i1}, k_{i2}, allowing the independent variation of joint positions q, and transmission stiffnesses $\sigma(q, \theta_{i1}, \theta_{i2})$.

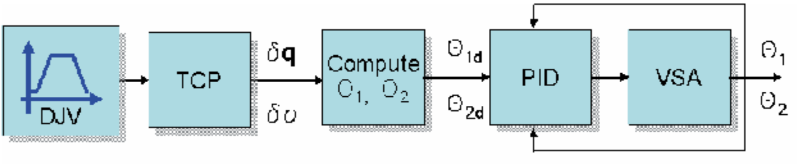


Fig. 6. VSA Control System. Desired main shaft positions θ_{1d}, θ_{2d} are chosen accordingly to (1) where $\delta\sigma$ is computed by the Trajectory Compliance Planner (TCP, [2]) so as to guarantee safety during motion with respect to a desired pre–planned joint velocity (DJV).

while motions $\delta\theta_1 = \delta\theta_2 = \delta\theta_q$ generate only joint angular displacements (in equlibrium configurations). This implies two equations

$$\begin{cases} \delta q = \frac{\delta\theta_1 + \delta\theta_2}{2} \\ \delta\sigma = \sigma(\frac{\delta\theta_1 - \delta\theta_2}{2}), \end{cases}$$

which can be solved, if σ is an invertible function, to find the desired angular displacements of the motors

$$\begin{cases} \delta\theta_1 = \delta q + \sigma^{-1}(\delta\sigma) \\ \delta\theta_2 = \delta q - \sigma^{-1}(\delta\sigma). \end{cases} \qquad (1)$$

In fig.6 it is reported a control scheme that can be implemented to control a Variable Stiffness Actuator (VSA), or a more general VST. A rotary VSA developed in our lab is described in fig.7. With the VSA we obtained experimental trajectory–tracking results which comply with the solutions of the Safe Brachistochrone applied in case of a compliant transmission (see fig.8). Experimental results are also reported in fig.9 showing the effective variation of transmission stiffness for VSA.

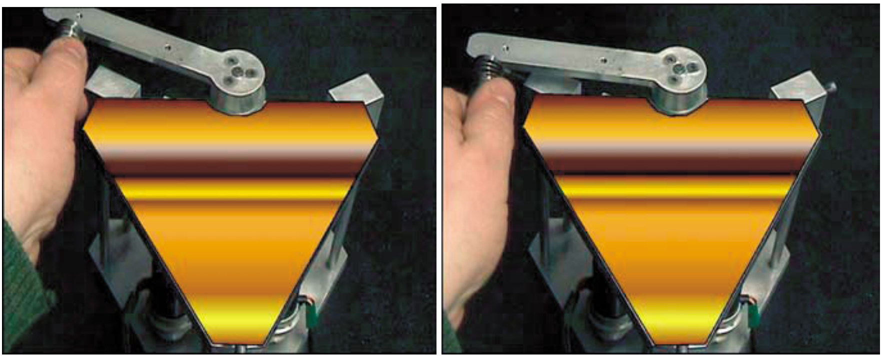


Fig. 7. The VSA developed at the Centro Interdipartimentale di Ricerca "E. Piaggio". The actuator is designed to allow the independent control of joint shaft position and stiffness (see the VSA in compliant (*left*) and stiff (*right*) configurations).

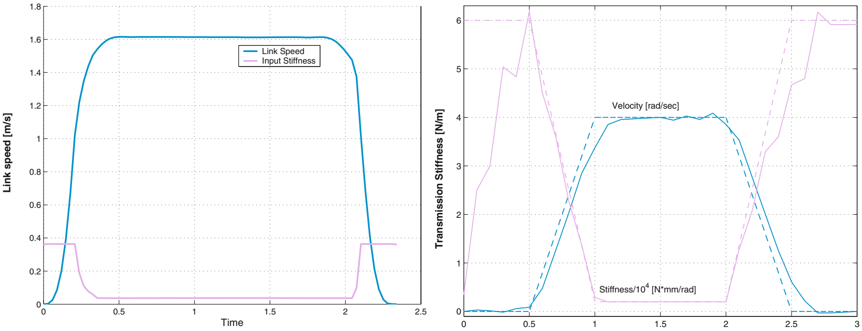


Fig. 8. Results of the Safe Brachistochrone applied to a variable compliant transmission (*left*), and joint speed and stiffness trackings results for the VSA during motion (*right*)

In the next paragraph we report some results of experiments we performed to test during motion the effectiveness of VSA, and of the VIA in general, in guaranteeing safety during a rest-to-rest motion task.

3.3 Safety guaranteed through VIA motion

The setup we realized to perform impact experiment with VSA is simply constituted by the VSA and a rotary accelerometer (see fig.10). A lightweight link, rigidly connected to the VSA joint shaft, impacts with the accelerometer during motion at different transmission stiffnesses σ. The measured acceleration $a(t)$ (see for instance fig.11) is then used to compute the relative HIC

$$HIC = \Delta T \left(\frac{1}{\Delta T} \int_{\Delta T} a(t)dt \right)^{2.5}$$

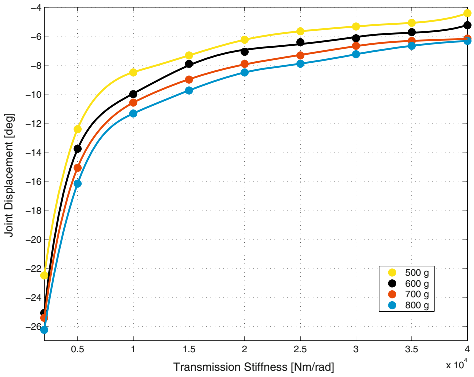


Fig. 9. Experimental displacement of the joint shaft of the VSA with respect to the transmission stiffness at increasing values of the applied axial load (from *yellow* to *blue*).

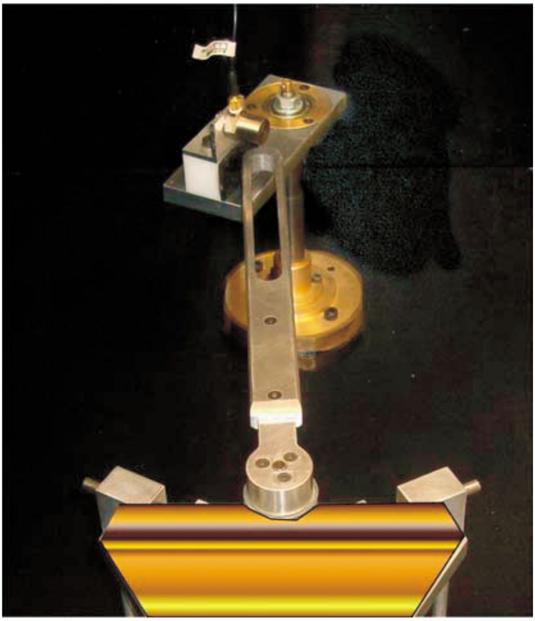


Fig. 10. Experimental setup for impact experiments with VSA.

where ΔT is the time duration of the impact. At this stage, it is be interesting to note how both ΔT and the maximum $a(t)$ *decrease* and *increase* with σ respectively. This behaviours generate variations of the injury risk. In fig.12 the experimental HIC curves for the VSA during motion at different transmission stiffness and velocity

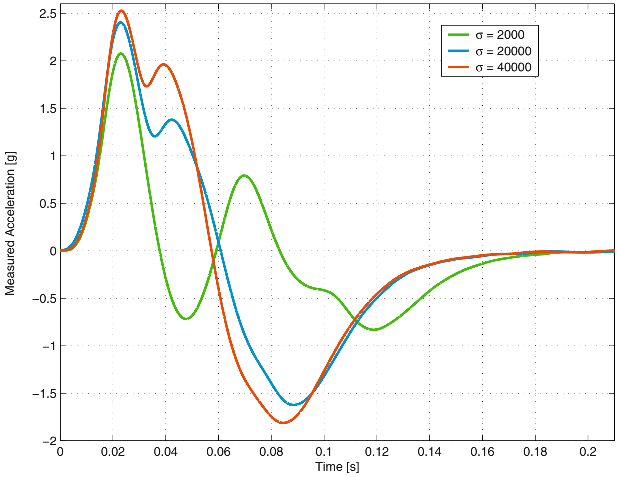


Fig. 11. Samples of the acceleration measured after collision at different stiffnesses of the VSA transmission, and at constant link velocity $v = 9\,[rad/sec]$.

(shaft velocity increases from *blue* to *green*) highlight how performance is limited due to safety constraints.

Such results suggest interesting hints for VIA control schemes. In particular, it shows that it is possible to set the higher impedance during motion until the safety bound is reached, and after that it is necessary to vary impedance parameters with respect to the link velocity, in a manner in which both maximum acceptable level of HIC and performance are preserved (see also the results of the Safe Brachistochrone problem reported in fig.8). In other words, the TCP in fig.6 chooses the trajectory $\sigma(HIC, DJV)$ to be followed by the VSA transmission stiffness, which ensures the desired HIC bound never be trespassed during the task execution. An example of experimental TCP output for a particular rest-to-rest velocity task, and safety bound $HIC = 75$, is reported in fig.13(*Left*). This output was used to control the VSA so as to perform impact experiments reported in fig.13(*Right*), from which the effectiveness of the proposed VSA in guaranteeing the safety bound during a trajectory traking task appears clearly.

4 Conclusion

The problem of achieving high performance with a mechanism which is safe to humans interacting directly with it poses many challenging technological problems. After a brief explanation of our concept of safety and performance for a robotic system, in this paper we have introduced, and experimentally validated, the concept of VIA that allow fast and accurate operation of a robot arm while guaranteeing a maximum suitable level of injury risk.

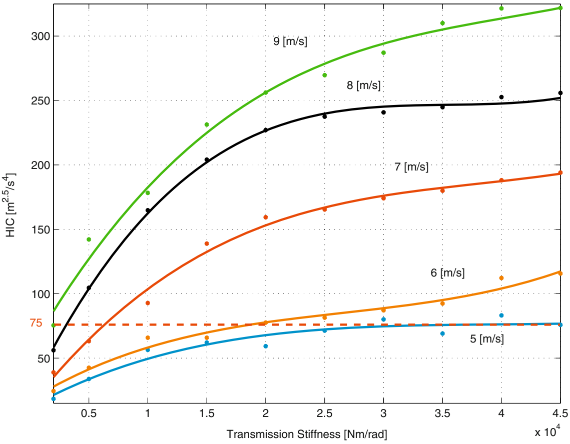


Fig. 12. Experimental results of the injury risk in case of impact at different tranmsission stiffnesses for VSA. As expected, the maximum allowable velocity v of the joint shaft decreases as the value of transmission stiffness σ increases, if an acceptable level of injury risk is chosen (as e.g the *red dashed* straight line, corresponding to $HIC = 75$). Continuous lines represent the fourth order minimum square interpolation of the experimental data (*dots*).

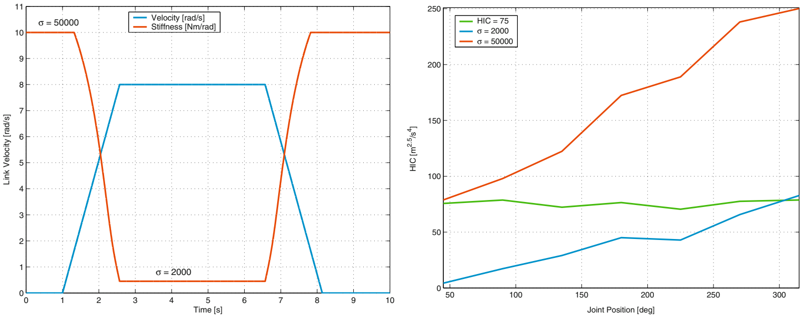


Fig. 13. (*Left*) Output of the TCP for a desired joint velocity at constant safety bound $HIC = 75$. The desired stiffness trajectory is obtained by interpolation of results in fig.12. (*Right*) Impact results in case of rigid (*red*), compliant (*blue*), and VIA (*green*) transmissions related to the accelerating phase of the task.

5 Acknowledgments

The work of undergraduate student Gianluca Boccadamo is gladly acknowledged. Work partially supported by EC contracts IST-2001-38040 (TOUCH-HAPSYS under F.E.T. Presence Initiative) and IST-2001-37170 (RECSYS).

References

1. A. Bicchi, G. Tonietti, M. Bavaro, and M. Piccigallo, "Variable stiffness actuators for fast and safe motion control," in *Preprints of Proc. Int. Symp. of Robotics Research*, R. Chatila and P. Dario, Eds., 2003.
2. A. Bicchi and G. Tonietti, "Dealing with the safety-performance trade-off in robot arms design and control," *IEEE Robotics and Automation Magazine, (in press)*, 2004.
3. Zinn M., Khatib O., Roth B., and Salisbury J.K., "A new actuation approach for human friendly robot design," in *Proc. of Int. Symp. on Experimental Robotics - ISER'02*, 2002.
4. J.K. Salisbury, *Kinematic and Force Analysis of Aticulated Hands*, Ph.D. thesis, Stanford University, Stanford, CA, May 1982.
5. T. Morita and S. Sugano, "Design and development of a new robot joint using a mechanical impedance adjuster," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1995, pp. 2469–2475.
6. M. Okada and Y. Nakamura, "Development of the cybernetic shoulder - a three-dof mechanism that imitates biological shoulder motion," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1999, pp. 453–548.
7. J. Versace, "A review of the severity index," in *Proc. of the Fiteenth Stapp Car Crash Conference*. Society of Automotive Engineers, 1971, number SAE Paper No. 710881, pp. 771–796.
8. Y. Kasai and Y. Morimoto, "Electronically controlled continuously variable transmission (ecvt-ii)," in *Proc. International Congress on Transportation Electronics*, Oct. 1988, pp. 33–42.
9. N. Takesue, G. Zang, J. Furusho, and M. Sakaguchi, "Precise position control of robot arms using a homogeneous er fluid," *IEEE Control Systems Magazine*, pp. 55–61, April 1999.
10. A. Bicchi and G. Tonietti, "Design, realization and control of a passively compliant robot for intrinsic safety," in *Proc. Second IARP/IEEE–RAS Joint Workshop on Techical Challenge for Dependable Robots in Human Environments*, 2002.

Actuation, Sensing, and Fabrication for In Vivo Magnetic Microrobots

K. Berk Yesin, Karl Vollmers, and Bradley J. Nelson

Institute of Robotics and Intelligent Systems
Swiss Federal Institute of Technology (ETH), Zurich
yesin@iris.mavt.ethz.ch, karl.vollmers@iris.mavt.ethz.ch, brad.nelson@iris.mavt.ethz.ch
http://www.iris.mavt.ethz.ch

Abstract. This paper investigates some of the fundamental design issues related to untethered biomedical microrobots guided inside the human body through external magnetic fields. Immediate application areas for these microrobots include cardiovascular, intraocular and inner-ear diagnosis and surgery. Issues investigated include the effects of magnetic actuation forces and viscous drag forces faced by magnetic microrobots, fabrication of MEMS devices with integrated hard magnetic materials and position sensing of the microrobot for intraocular applications. A new active defocused tracking method is proposed for visually servoing the microrobot in 3D using only a single microscope view.

1 Introduction

The state-of-the-art in MEMS technology is progressing from individual, chip-level microsensor and microactuator devices to complete integrated microrobot systems. These types of systems will impact future minimally invasive surgical techniques by providing sub-mm untethered biomedical microrobots capable of performing a number of procedures. The benefit will be even less injury to the patient resulting in correspondingly faster recovery times. Proven MEMS technologies such as micro-needles, micro-pumps and force and chemical sensors will be carried on-board for

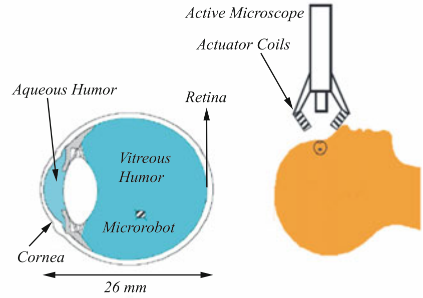


Fig. 1. The vitreous humor is a transparent, viscoelastic substance that occupies a large portion of the eyeball. Controlled magnetic fields are created by a set of coils to steer the microrobot inside the vitreous while an active microscope provides real-time 3D position feedback.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 321–330, 2006.

a variety of surgical and diagnostic tasks. Possible areas of application for these microrobots include cardiovascular, intraocular and inner-ear diagnosis and surgery.

Although electronic and mechanical systems have been miniaturized by VLSI and MEMS technologies, no counterpart to these exists for electro-chemical energy storage. Therefore, the only viable option for the actuation and steering of such a microrobot is external energy transfer. The use of magnetic fields generated ex-vivo for energy transfer and actuation can provide a solution to this problem.

This paper investigates some of the fundamental design issues related to magnetically actuated microrobotic systems. The nature of magnetic and viscous drag forces are analyzed. The advantages of using hard magnetic materials and fabrication methods for their integration into MEMS are discussed. Another major challenge is to create a suitable sensing technology to track the position of the microrobot inside the body. Different sensing methods will be better suited for application in different parts of the body, for example, magnetic or ultrasound sensing for cardiovascular microrobots and vision for intraocular applications. In this paper we also present a vision based sensing system to track the 3D position of a microrobot inside the eye. This system uses a single optical microscope to obtain a purposely defocused view of the microrobot to unambiguously resolve depth information.

2 Wireless Actuation Through Ex-Vivo Magnetic Fields

2.1 Nature of Magnetic Forces

The primary vectors that define the magnetostatic field in magnetized matter are (external) magnetic field strength, $\boldsymbol{H}$ (A/m), magnetization of the matter $\boldsymbol{M}$ (A/m) and magnetic flux density $\boldsymbol{B}$ (Tesla). The relationship between these vectors is

$$\boldsymbol{B} = \mu_0(\boldsymbol{H} + \boldsymbol{M}) \tag{1}$$

where μ_0 is the magnetic permeability of free space defined as $4\pi \times 10^{-7}$ Tm/A. For the idealized case of linear, isotropic and homogeneous media the following relationships simplify (1) as

$$\boldsymbol{M} = \chi \boldsymbol{H} \tag{2}$$

$$\boldsymbol{B} = \mu_0(1 + \chi)\boldsymbol{H} = \mu_0 \mu_r \boldsymbol{H} \tag{3}$$

where χ and μ_r are the susceptibility and relative permeability of the media, respectively. In general, these values are not constant but change with magnetization, approaching zero as the magnetization reaches a material dependent limit called the *saturation magnetization* M_s. Within the saturation limits, the permeability can be thought as an amplification factor that creates a net magnetic field inside the matter through an external field.

Ferromagnetic materials exhibit the largest relative permeability (on the order of $10^3 - 10^5$). The dependence of magnetization on external fields (i.e. the M-H curve) for a ferromagnetic material is shown in Figure 2. The non-linear response of the material also changes with its previous state of magnetization (indicating hysteresis). A ferromagnetic material magnetized to saturation will keep part of its magnetization (M_r, *remanent magnetization*) after the magnetizing field is taken away. It is necessary to apply a field H_{ci} (*intrinsic coercivity*) in the opposite direction of magnetization to cancel this magnetization. Soft magnetic materials (e.g. iron) are characterized by their high permeability and low coercivity ($H_{ci} < 10^3$ A/m) such that they can be easily magnetized and demagnetized by external fields. Hard magnetic materials (e.g. permanent magnets), on the other hand, have lower permeability but high coercivity ($H_{ci} > 10^4$ A/m), therefore, once magnetized, they retain their magnetization against external fields.

The magnetic force and torque that are exerted on an object with uniform magnetization $\boldsymbol{M}$ in a magnetic field with flux density $\boldsymbol{B}$ are defined as

$$\boldsymbol{F_m} = V_m(\boldsymbol{M} \cdot \nabla)\boldsymbol{B} \tag{4}$$

$$\boldsymbol{\tau_m} = V_m \boldsymbol{M} \times \boldsymbol{B} \tag{5}$$

where V_m is the volume of the magnetized object. Notice that the magnetic torque is dependent on $\boldsymbol{B}$ whereas the magnetic force is dependent on the gradient of $\boldsymbol{B}$. Equations (4) and (5) indicate that magnetic forces are volumetric. Therefore, the required fields and field gradients to exert a certain torque and force on a magnetized object increase rapidly as the object gets smaller.

2.2 Analysis of Required Magnetic Fields

Other than the magnetic forces analyzed above, gravitation/bouyancy and fluid drag forces are also acting on the microrobot. The drag force can be expressed as

$$F_d = 1/2 C_d \rho_f A \nu^2 \tag{6}$$

where ρ is the density of the body fluid, A is the cross sectional area of the robot, ν is the relative velocity of the robot with respect to the fluid media and C_d is the drag coefficient representing the overall effect of the robot's geometry on the drag force. A simplified analysis can be made assuming a spherical shape for the microrobot in which case (6) becomes

$$F_d = 3\pi\eta_f D\nu \tag{7}$$

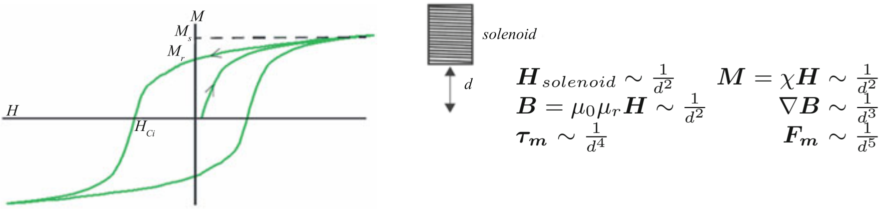

M
M_s
M_r
H
H_{Ci}
solenoid
d
$\boldsymbol{H}_{solenoid} \sim \frac{1}{d^2}$
$\boldsymbol{M} = \chi\boldsymbol{H} \sim \frac{1}{d^2}$
$\boldsymbol{B} = \mu_0\mu_r\boldsymbol{H} \sim \frac{1}{d^2}$
$\nabla\boldsymbol{B} \sim \frac{1}{d^3}$
$\boldsymbol{\tau_m} \sim \frac{1}{d^4}$
$\boldsymbol{F_m} \sim \frac{1}{d^5}$

Fig. 2. Typical M-H curve of a ferromagnetic material. The force and torque on a magnetic object created by a solenoid coil.

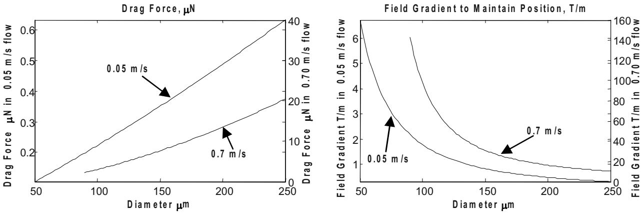


Fig. 3. Drag forces on a spherical magnet in the human blood and the required field gradient to resist the fluid drag for various sphere diameters. Plots for two different flow velocities (0.05 and 0.7 m/s) are shown. Note the different scales of the two plots.

where D is the diameter of the sphere and η_f is the fluid viscosity. In addition to the drag force, a net buoyancy force will be acting on the robot

$$F_b = V_r(\rho_r - \rho_f)g \tag{8}$$

where V_r and ρ_r are the density and volume of the robot respectively and g is the gravitational acceleration. The most important outcome of (4), (5), (6) and (8) is that, whereas magnetic and buoyancy forces are volumetric, the fluid drag forces are dependent on the area. For this reason, as the size of the robot gets smaller the required field gradient to move at a particular speed rapidly increases.

For cardiovascular applications, the fluid media that the microrobot moves in will be blood. The flow velocity of blood in the human body ranges between 0.05 m/s in the capillaries to 0.7 m/s at the exit of the aorta and viscosity of about 2.8 cP (centipoise) [11]. Figure 3 show the drag forces on a spherical permanent magnet with a diameter between 50 to 250 μm and magnetization $\boldsymbol{M} = 2.4 \times 10^4$ A/m (a reasonable value for deposited hard magnetic films on MEMS devices [9]) are plotted. The two extremes of blood flow velocity are considered in the two plots. In addition, the necessary field gradients to maintain position against the flow (i.e. balance the drag forces) are also shown.

The human eye, mostly filled by a transparent gel-like fluid called the vitreous humor, is another challenging workspace for a biomedical microrobot. The vitreous humor fills the posterior cavity of the eye between the lens and the retina (Figure 1) [12]. Although it is composed almost entirely of water (99%) it also contains a collagen fiber network, hyaluronic acid and soluble proteins, and, therefore, has the properties of a viscoelastic liquid with high viscosity. Viscoelastic materials can modeled by spring and dashpot elements representing the elastic and viscous characteristics of the material with shear modulus G and dynamic viscosity η. Two common models of viscoelastic behavior are the Maxwell and Kelvin elements that consist of a spring and a dashpot in series and in parallel, respectively (Figure 4).

A research study utilizing a specially built magnetic-bead rheometer device that performed non-destructive testing inside the eye has been reported by Lee et al [13][14]. Lee uses the Burgers model that consists of a Maxwell and Kelvin element in series to describe the properties of the vitreous (Figure 4a). Of the four parameters

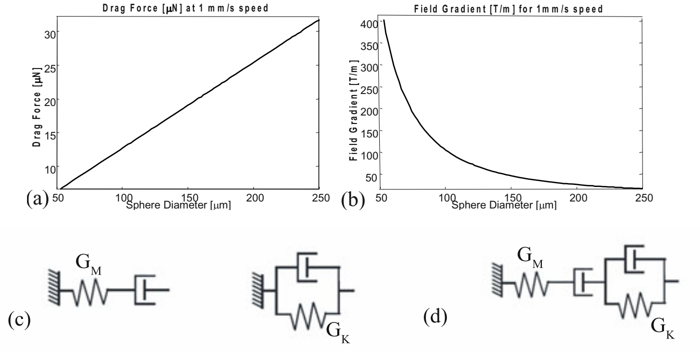


Fig. 4. a) and b) Drag forces on a spherical magnet moving at 1 mm/s in the human vitreous and the required field gradient for various sphere diameters. Magnetization is $M = 2.4 \times 10^4$ A/m and viscosity is 1398 cP. c) Maxwell and Kelvin elements for decribing the rheological properties of materials. d) Burgers model used by Lee et al. [13] to model the vitreous humor.

in this model, the viscosity of the Maxwell dashpot (η_M) represents the irreversible flow of the material under constant stress that is the main source of the viscous drag forces on the robot as it is steered inside the eye. The mean values of viscosity were 1398 cP at the anterior, 2179 cP at the central and 4862 cP at the posterior regions. The values for the Kelvin viscosity were 4 to 10 times lower, further indicating the dominance of Maxwell viscosity for the drag force. Figure 4 shows the drag forces on a spherical permanent magnet with a diameter between 50 to 250 μm and magnetization 2.4×10^4 A/m moving inside the vitreous at a speed of $1mm/sec$. The mean viscosity at the anterior region is used. In addition, the necessary field gradient to balance the drag forces are also shown.

2.3 Actuation Through Magnetic Fields

Equations (4) and (5) suggest that controlled external magnetic fields can be used to induce torques and forces on a magnetized object and control its orientation and position. At the micro scale, active control and steering of magnetic beads in two and three dimensions have been demonstrated for DNA micromanipulation [6] and for performing intracellular rheology and force measurements [4] [5]. Whereas ferromagnetic materials exhibit higher susceptibility and saturation magnetization, the control of the beads' magnetization vector by external fields is difficult due to the hysteresis effect. Also, undesired interactions between magnetized beads can occur even after the external field has vanished. Superparamagnetic beads, which do not show hysteresis, are often used instead. The susceptibility of these beads, however, is much lower than ferromagnetic materials (e.g. 10^{-2} for commonly used M280 beads from Dynal Inc. [4]). Solenoid electromagnets with soft magnetic poles are typically used to generate high field gradients as much as 100 T/m across a 20 mm air gap [6].

An important issue related to the control of a magnetic microrobot is the nonlinear nature of the field and gradients that are created by electromagnet coils. The field from an aircore solenoid coil along its axis is roughly proportional to the inverse square of the distance d to the solenoid. In this case the torque and force on the soft magnetic material are proportional to the fourth and fifth inverse power of the distance respectively, as shown in Figure 2.

One way of reducing the effect of such nonlinearities is to create uniform magnetic fields and field gradients using various coil configurations [10]. For example, the Helmholtz coil configuration consists of two identical coils that are placed on the same axis and separated by a distance equal to the radius of the coils. This arrangement generates a uniform field close to the center of the coil pair when current passes in the same direction in both coils. Similarly, the Maxwell coil configuration can generate a uniform gradient near the center when the coils are separated by $\sqrt{3}$ times the radius and the current passes in the opposite direction. Figure 5 shows the plot of the fields of Helmholtz and Maxwell coils. Both of these coils are commonly used in MRI systems. Although, the uniform fields of commercial MRI machines are quite high (0.5-2.0 T), the gradient fields they can generate continuously are in the 0.01 to 0.05 T/m range.

3 Fabrication of MEMS with Integrated Hard Magnetic Materials

Material choices can also have an effect on the non-linear effects mentioned above. The magnetization vector in a hard magnetic material is not as dependent on external magnetic fields as it is in soft materials. In a soft magnetic material the magnetization vector follows the direction of the external field (within the energy limits set by the shape anisotropy effects). Therefore, neither the directionality nor the magnetization of the device is fixed. However, soft magnetic materials with saturation magnetizations up to 19×10^5 A/m can generate larger forces when operated at saturation.

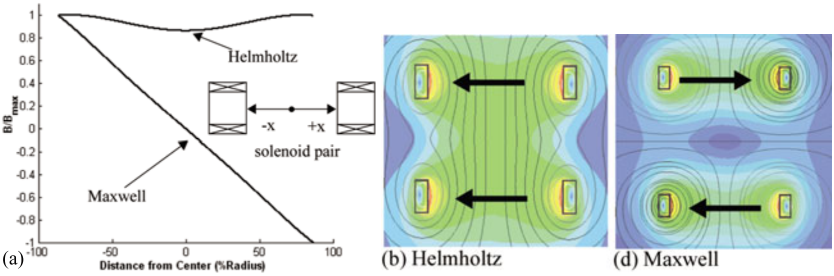


Fig. 5. a) Normalized magnetic field magnitudes along the central axes of Helmholtz and Maxwell coils. b) and c) Magnetic field and flux lines for Helmholtz and Maxwell coils, respectively. Arrows indicate the direction of current in the coils.

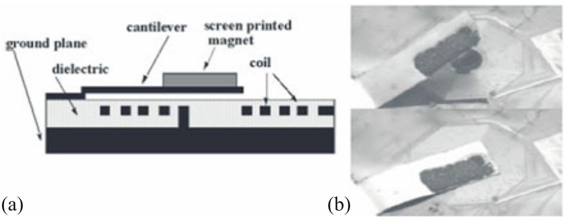


Fig. 6. Magnetically actuated cantilever device using screen printed NdFeB magnetic paste. a) Crossection of the device b) The cantilever is actuated by passing current through the 40 turn coil.

The ability to assign a direction to the robot is critical for accomplishing tasks with onboard tools and actuators.

A variety of techniques have been proposed for integrating hard magnetic materials into MEMS devices, including sputtering, thermal evaporation, assembly, electroplating, screen printing, spin coating and molding. Each method offers its own advantages and disadvantages. Our recent work [9] on integrating NdFeB magnetic powder into MEMS devices by wafer-level processing has resulted in magnets that have larger thickness (thus volume) and are stronger than thin film magnets. The screen printed magnets were fabricated using commercially available magnetic powder (Magnequench, MQP-S-9-8). The M_r value for the screen printed magnets was measured to be 2.4×10^5 A/m. A magnetically actuated thin film cantilever was built with screen printed magnets to demonstrate their integration with microfabrication processes. Figure 6 illustrates the device.

4 Microrobot Position Sensing Through Active Defocused Visual Tracking

Optical microscopes are commonly used to provide feedback for magnetic guidance of particles inside cells and other biological media [6][4]. These systems, however, operate only in 2D. In [5], a 3D magnetic bead steering system that used a laser light scattering technique to provide high bandwidth (10 kHz) position feedback was developed. Unfortunately, this technique is not suitable for tracking the intraocular microrobot since it requires a through path (i.e. not reflected) between the laser source and the photonic sensor element. The location of the eye inside the head does not allow such access. Although multi-camera methods with microscopes can provide precise 3D position information in real-time [15], these methods are also not suitable in this case since only a small window (i.e. the iris) is available for seeing inside the eye such a configuration is not possible.

Depth from focus/defocus methods are commonly used in microscopic vision to resolve the position of objects along the view-axis [16][17]. Depth from focus methods use a visual metric of image sharpness to detect whether regions of interest are in focus as the focus point of the microscope lens is moved along the optical axis in controlled increments. This way the relative distance of two points at different depths can be found by determining where along this focal scan the points are in best focus as illustrated in Figure 7. Due to the necessary scanning motion this method is not applicable in real-time. An alternative is the depth from defocus method which uses the focus measure response directly to make measurements of distance to the plane of best focus. However, due to the symmetry of the focus measure curve about the plane of best focus, there are two depths corresponding to the same focus measure except at optimum focus. This problem can be solved using two cameras which share the same view through image splitting optics but at different planes of focus [18].

For 3D position tracking of the magnetic microrobot a third strategy has been developed that combines the high depth sensitivity of the depth from defocus method with an increased range of measurement and uses only a single microscope. The *active defocused tracking* method uses a microscope system that is actively servoed along the view axis to keep the target object (i.e. the microrobot) in a mid-level of defocus *at one side of the focus measure curve* (e.g. point B in Figure 7). This way a calibrated focus measure curve can be used to resolve the distance of the target to the plane of best focus, E_C, in real-time (i.e. at camera frame rate). This position error is added to the position of the microscope along the view axis, P_C, which can be measured by a linear encoder on the microscope stage. The resulting position error is used in the feedback loop that controls the external magnetic field and steers the microrobot inside the eye. A second feedback loop uses the defocus error, E_B, to servo the microscope stage so that the target object is kept at a mid-level of focus (Figure 8d).

The defocus tracking method was successfully implemented. A Sutter MP-285 micropositioning stage was used to actuate a microscope lens and a digital video camera whereas an identical stage was used to carry a target object. The target is a small piece of retroreflective tape that contains microspheres that reflect incoming light parallel to the direction of incidence. Figure 8 shows screen shots of the visual

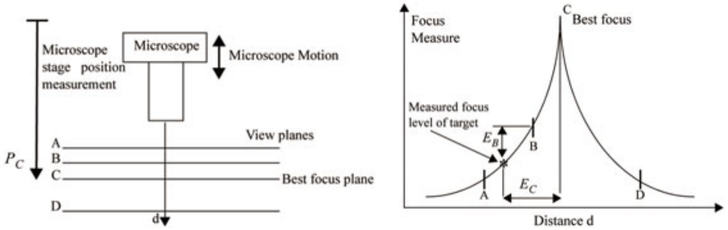
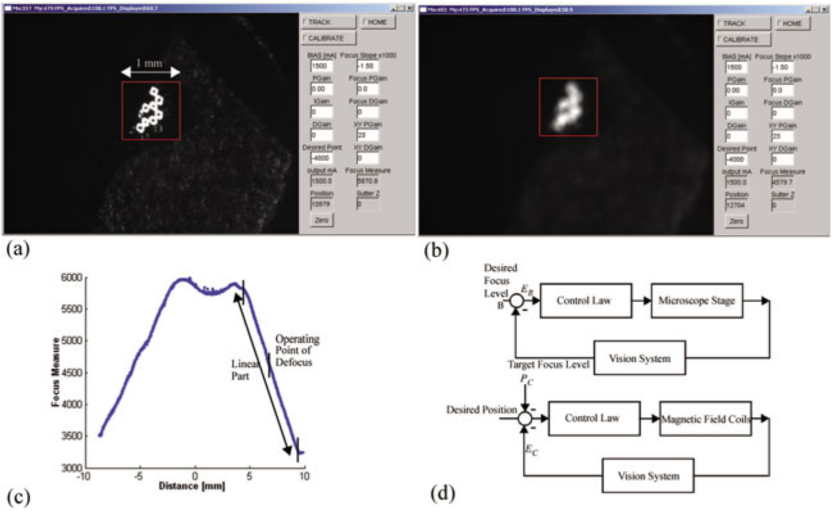


Fig. 7. A suitable measure of image focus reaches its maximum value when the reqion of interest is in best focus and decreases monotonically as the distance changes on either side of the best focus plane. The distance of a target to the best focus plane can be resolved from this curve if the target is always kept in defocus at one side of the curve by actively servoing the microscope stage.



(a)

BIAS [mA] 1500
Focus Slope x1000 -1.50
PGain 0.00
Focus PGain 0.0
IGain 0
Focus DGain 0
DGain 0
XY PGain 23
Desired Point -4000
XY DGain 0
output mA 1500.0
Focus Measure 5870.8
Position 10979
Sutter Z 0

(b)

BIAS [mA] 1500
Focus Slope x1000 -1.50
PGain 0.00
Focus PGain 0.0
IGain 0
Focus DGain 0
DGain 0
XY PGain 23
Desired Point -4000
XY DGain 0
output mA 1500.0
Focus Measure 4579.7
Position 12704
Sutter Z 0

Fig. 8. a) and b) Screen shots of the active defocused tracking program showing the retroreflective target object in best focus and in mid level focus. c) The variance of grayscale values as a focus measure. During defocused tracking the target is actively kept at the center of the linear portion of this curve. d) Two separate control loops are used for the tracking and guidance of the microrobot.

tracking program with the target object at the best focus and in the mid-focus plane. A 2D tracking algorithm that uses simple thresholding and centroid calculation was used to detect the location of the target in the image taking advantage of the high contrast presented by the retroreflective target.

Defocusing is essentially a low-pass filtering operation, therefore, the amount of high frequency components in the area of interest is an indicator of image focus. In this work, the variance of the grayscale values inside the target area was used as the focus measure which created a linear response away from the best focus plane, as shown in Figure 8c. Note that this particular measure and target did not create a sharp peak at the best focus plane, however, the active defocused tracking method is not effected by this.

5 Conclusions

Magnetic actuation is a suitable mode of energy transfer for untethered biomedical microrobots. The improportional scaling of magnetic and viscous drag forces dictates the need for higher gradient magnetic fields over larger areas than are currently available with commercial MRI systems. For intraocular microrobots, limited physical access to the eye creates additional challenges for actuation and sensing. A real-time 3D visual tracking method using an active microscope that is suitable for visually servoing the position of the microrobot inside the eye was presented.

References

1. U. Hafeli, W. Schutt, J. Teller, M. Zborowski (ed), *Scientific and Clinical Applications of Magnetic Carriers*, NewYork: Plenum, 1997.
2. D.L. Holligan, G.T. Gilles, J.P. Dailey, "Magnetic Guidance of Ferrofluidic Nanoparticles in an In Vitro Model of Intraocular Retinal Repair", *IOP Nanotechnology*, vol. 14, pp. 661–666, 2003.
3. M. Zborowski,"Physics of Magnetic Cell Sorting", in Hafeli, Schutt, Teller, Zborowski et al., *Scientific and Clinical Applications of Magnetic Carriers*, NewYork: Plenum, pp. 205–232, 1997.
4. F. Amblard, B.Yurke, A. Pargellis, S. Leibler, "A Magnetic Manipulator for Studying Local Rheology and Micromechanical Properties of Biological Systems", *Review of Scientic Instruments*, vol. 67, no. 3, pp. 818–827, 2000.
5. L. Vicci, *A 3D Magnetic Force Manipulator DC Prototype*, UNC Chapel Hill Dept. Of Computer Science Technical Report, no. TR01-031, 2001.
6. C. Haber, D. Wirtz, "Magnetic Tweezers for DNA Micromanipulation", *Review of Scientic Instruments*, vol. 7, no. 12, pp. 4561–4570, December 2000.
7. K. Ishiyama, K.I. Arai, M. Sendoh, A. Yamazaki, "Spiral-type Micromachine for Medical Applications", *Proc. 2000 International Sym›posium on Micromechatronics and Human Science*, pp. 65–69, 2000.
8. J. Mathieu, S. Martell, L.Yahia, G. Soulez, G. Beaudoin, "Preliminary Studies for Using Magnetic Resonance Imaging Systems as a Mean of Propulsion for Microrobots in Blood Vessels and Evaluation of Ferromagnetic Artefacts", *IEEE CCECE Conf. on Elec. and Comp. Eng.*, pp. 835–838, 2003.
9. K. Vollmers, R. Anderson, B.J. Nelson, J. Pepin, "High Strength Rare Earth-Iron-Boron Printed Magnets Used In A Long Throw-High Force Electromagnetic Actuator With Microfabricated Coil", *IEEE Int. Conf. on MEMS (MEMS2003)*, pp. 60–63, 2003.
10. J. Jin, *Electromagnetic Analysis and Design in Magnetic Resonance Imaging*, Florida: CRC Press, 1999.
11. A. Despopoulos, S. Silbernagl, *Color Atlas of Physiology*, NewYork: Thieme Medical Publishers, pp. 156, 1991.
12. T.V. Chirilla, Y. Young, "The Vitreous Humor", *Handbook of Biomaterial Properties*", J. Black and G. Hastings et al., Chapman& Hall, London, 1998.
13. B. Lee, M. Litt, G. Buchsbaum, "Rheology of the Vitreous Body. Part I: Viscoelasticity of Human Vitreous", *Biorheology*, vol. 29, pp. 521–533, 1992.
14. B. Lee, *Comparative Rheological Studies of the Vitreous Body of the Eye*, Ph.D. Thesis, University of Pennsylvania, 1992.
15. K.B. Yesin, B.J. Nelson, "Robust CAD Model Based Visual Tracking for 3D Microassembly Using Image Space Potentials", *IEEE International Conference on Robotics and Automation*, 2004.
16. S. Allegro, C. Chanel, J. Jacot, "Autofocus for Automated Microassembly Under a Microscope", *IEEE Int. Conf. on Image Processing*, vol. 1, pp. 677–680, 1996.
17. B. Vikramaditya, B.J. Nelson, "Visually Guided Microassembly Using Optical Microscopes and Active Vision Techniques", *IEEE Int. Conf. on Robotics and Automation*, pp. 3172–3177, 1997.
18. S.K. Nayar, M. Watanabe, M. Noguchi, "Real-Time Focus Range Sensor", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1186–1198, 1996.

IX. Flying Robots

The mobility of flying robots is often the main reason researchers select this mode of robot for a specific application. On top of that, it provides a bird's eye view of the environment, which is often very useful in solving certain problems.

The first article is the result of a collaborative effort between CSIRO (Australia), the University of Southern California and Dartmouth College. Corke and colleagues present a sensor network deployment method using autonomous aerial vehicles for a large-scale environmental monitoring task. The article presents the algorithm for network sensor deployment, maintenance of network connectivity and the results of field experiments.

The next article by Merz, Duranti and Conte studies the classic problem of autonomous landing of an unmanned helicopter, this time performed only using vision and inertial sensor. The algorithm is robust to weather conditions and results are provided from simulation and actual flight tests.

In the third article, Grocholsky and co-workers present the results of sensor fusion techniques for air-ground robotic operations. The technique exploits the complimentary capabilities of air and ground robots and is applied to a task of tracking multiple ground targets.

The last article by Ng and colleagues shows the effectiveness of reinforcement learning applied to the control of an autonomous helicopter platform. Helicopters have "highly stochastic" and "nonlinear" dynamics and autonomous helicopter flight is widely regarded to be a challenging control problem especially at low speed. Here, reinforcement learning was shown to answer this challenging problem and not only maintained the helicopter in the air, but also keeping it inverted. An "upside down" flying helicopter is not something we see everyday.

Deployment and Connectivity Repair of a Sensor Net with a Flying Robot

P. Corke[1], S. Hrabar[2], R. Peterson[3], D. Rus[4], S. Saripalli[2],
G. Sukhatme[2]

[1] CSIRO ICT Centre
Australia, peter.corke@csiro.au
[2] Center for Robotics and Embedded Systems
University of Southern California, Los Angeles, California, USA
shrabar@robotics.usc.edu, srik@robotics.usc.edu, gaurav@robotics.usc.edu
[3] Dartmouth Computer Science Department, Hanover, NH 03755 USA,
rapjr@cs.dartmouth.edu
[4] Computer Science and Artificial Intelligence Laboratory
MIT, Cambridge MA 02139, USA, rus@csail.dartmouth.edu

Abstract. We consider multi-robot systems that include sensor nodes and aerial or ground robots networked together. Such networks are suitable for tasks such as large-scale environmental monitoring or for command and control in emergency situations. We present a sensor network deployment method using autonomous aerial vehicles and describe in detail the algorithms used for deployment and for measuring network connectivity and provide experimental data collected from field trials. A particular focus is on determining gaps in connectivity of the deployed network and generating a plan for repair, to complete the connectivity. This project is the result of a collaboration between three robotics labs (CSIRO, USC, and Dartmouth.)

1 Introduction

We wish to develop distributed networks of sensors and robots that perceive their environment and respond to it. To perform such tasks there needs to exist a synergy between mobility and communication. Sensor networks provide robots with faster and cheaper access to data beyond their perceptual horizon. Conversely robots can assist a sensor network by deploying it, by localizing network elements post deployment [6], by making repairs or extensions as required, and acting as “data mules” to relay information between disconnected sensor clusters.

In this paper we describe our algorithms and experiments for deploying sensor networks using an autonomous helicopter. The static sensor nodes are Mica Motes and the mobile node is the autonomous helicopter. Once on the ground, the sensors establish an ad-hoc network and compute their connectivity map in a localized and distributed way. If the network is disconnected, a localized algorithm determines waypoints for the helicopter to drop additional nodes at.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 333–342, 2006.

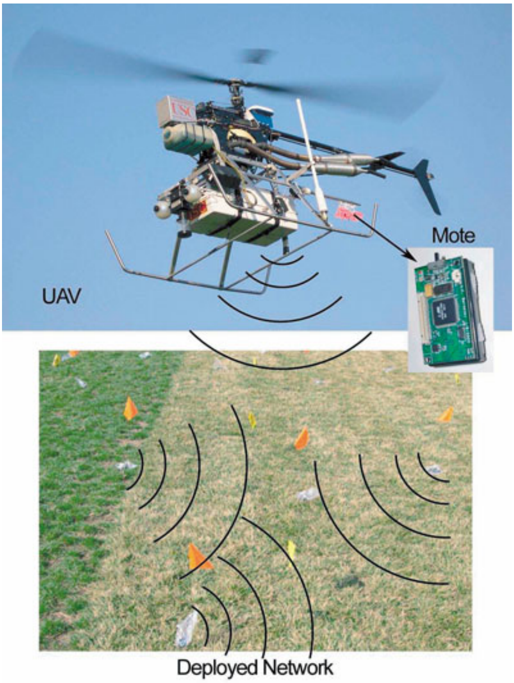


Fig. 1. AVATAR Autonomous Helicopter with a sensor interface for deploying sensors

2 Related Work

Our work builds on important previous work in sensor networks [8, 11, 14] and unmanned aerial vehicles [3, 16]. It bridges the two communities by integrating autonomous control of flying vehicles with multi-hop message routing in ad-hoc networks. Autonomous aerial vehicles have been an active area of research for several years. Autonomous model helicopters have been used as testbeds to investigate problems ranging from control, navigation, path planning to object tracking and following. Flying robot control is a very challenging problem and our work here builds on successes with hovering and control for two autonomous helicopters [3,17]. Several other teams are working on autonomous control and other varied problems with helicopters. A good overview of the various types of vehicles and the algorithms used for control of these vehicles can be found in [17] . Recent work has included autonomous landing [16,19], aggressive maneuvering of helicopters [9] and pursuit-evasion games [21].

Research in sensor networks has been very active in the recent past. An excellent general introduction on sensor networks can be found in [8]. An overview of hardware and software requirements for sensor networks can be found in [12] which describes the Berkeley Mica Motes. Algorithms for positioning a mobile sensor network includes even dispersal of sensors from a source point and redeployment for network rebuilding [2, 13]. Other important contributions include [1, 4, 10, 15, 18].

In [6] we describe a decentralized and localized algorithm called *robot-assisted localization* for localizing a sensor network with a robot helicopter. In [7] we describe an algorithm called *network-assisted navigation* in which a sensor network guides a robot helicopter. In [5] we describe an algorithm and preliminary experiments for deploying a sensor network with a robot helicopter. Here we extend this work to include deployment and connectivity repair and discuss our field experiments using a autonomous helicopter and a 55-node sensor network.

3 Approach

Our approach consists of three phases. In the first phase, an initial autonomous network deployment is executed. In the second phase, the entire network measures its connectivity topology. If this topology does not match the desired topology, a third phase is employed in which waypoints for the helicopter are computed at which additional sensors are deployed. The last two phases can be run at any point in time to detect the potential failure of sensor nodes and ensure sustained connectivity.

3.1 Deployment Algorithm

Given a desired network topology for the deployed network deployed, and a deployment scale (usually the inter-sensor distance between the nodes in the network), we embed the topology in the 3-dimensional hyper-plane at the given location and extract desired node locations from the resulting embedding. The resulting locations are the (x, y, z) co-ordinates where the sensors need to be deployed.These are given as way-point inputs to the helicopter controller. The helicopter then flies to each of these way-points autonomously, hovers at each of them and then deploys a sensor at the specified location.

3.2 Connectivity Measurement Algorithms

Two methods were used to measure network connectivity: a ping-based connectivity measure and a token-passing based measure. For the ping-based measure, a Mote sensor that has been specially modified to add physical user interface controls (a potentiometer and switch) is used to control and configure the sensor side of the ping connectivity tests prior to Algorithm 1 executing.

For the token based connectivity measure each node assumes its network ID as its token. All nodes broadcast and trade tokens as described in Algorithm 2. Tokens are only propagated amongst nodes in connected regions. Thus, disconnected regions will have differing token values. This algorithm is run automatically at 30 second intervals.

Slight differences in connectivity were observed when comparing the ping and token measurements of connectivity and were found to result from the differences

Algorithm 1 Ping connectivity algorithm for ground deployed motes.

```
Wait for experiment configuration/start message
Initialization: Set configuration mode = air-to-ground, ground-to-ground, or ground-to-air. Set count = number of ping iterations.
Send a multi-hop forwarding of start message to other motes.

Thread 1
for i=1 to count do
   if mode = ground-to-ground OR mode = ground-to-air then
      broadcast a ping message.
   Sleep a random interval

Thread 2
while Listen for messages do
   if message is a ping then
      if mode = air-to-ground OR mode = ground-to-ground then
         reply to ping.
   else if Message is a ping reply. then
      tabulate reply.

Termination: broadcast counts of replies per mote ID in response to download message.
```

in message length. Pings are very short messages (1 byte payload) while token messages are longer (10 byte payload). The longer message length increases the chance of collisions and reduces the probability of reception of token messages.

3.3 Connectivity Repair Algorithm

The token based connectivity algorithm is a localized and distributed algorithm for computing connected components in the deployed network. Each node ends up with one token that denotes the group to which it belongs. These tokens are collected by the helicopter during a sweep of the field. If more than one token is collected, the network is not connected and new sensor deployments are needed. The locations of the collected tokens can be used to determine the repair regions.

We have developed two algorithms for repairing network connectivity. In the first algorithm, the robot helicopter estimates the location of the gap between two

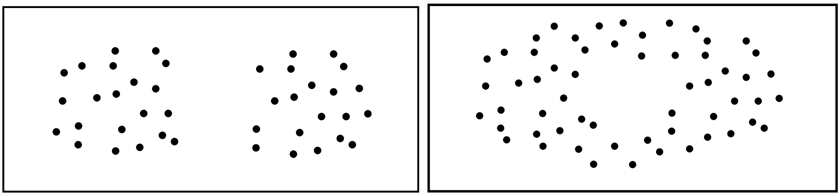


Fig. 2. (Left)Two disconnected components in a sensor network field. (Right) A single network which is not fully connected.

disconnected components by estimating the locations of the fringe nodes (see Figure 2(Left)). The repair locations are interpolated between the fringes, based on average sensor communication range, which is known.

Algorithm 2 Distributed algorithm for identifying the connected components in a sensor network. All the nodes in one connected component will have the same *component* value as a result of this protocol.

```
for each node in the sensor network do
  $component$ = id
for each node in the sensor network do
  broadcast node id.
  while listen for $new_{id}$ broadcasts do

    if received id $>$ $component$ then
      $component = new_{id}$
    broadcast $new_{id}$
Helicopter collects all $component$ values
Helicopter determines unique $component$ values as the number of connected components.
```

In the second algorithm the sensor field computes a potential field to regions of “dark” sensors (see Figure 2(Right)) discovered within it and guides the helicopter there using the potential field algorithm in [7]. This second algorithm handles both complete disconnections and holes in the middle of the sensor field.

For our field experiments we used a hand computed version of the first algorithm described above, averaging the fringe locations to determine a center and averaging the fringe gap distance to determine interpolated repair locations used in the autonomous repair deployment phase.

The general connectivity matching problem remains open. This problem reduces to computing subgraph embeddings which is intractable for the optimal case. We hope to identify a good approximation.

4 Experiments and Results

We have implemented the deployment algorithms on a hardware platform that integrates hardware and software from three labs: USC’s autonomous helicopter, Dartmouth’s sensor network, and CSIRO’s interface between a helicopter and a sensor network. Over January 23–25 the three groups met at USC and conducted joint experiments which demonstrate, for a desired network topology, (1) autonomous deployment of a 40 node sensor network with a robot helicopter, (2) autonomous and localized computation of connectivity maps (3) autonomous determination of disconnected network components and autonomous repair of the disconnections.

4.1 The Experimental Testbed

The experimental testbed consists of three parts (a) An autonomous helicopter (b) ”Mote” sensors and (c) Helicopter-sensor interface. The helicopter [20] is a gas-powered radio-controlled model helicopter fitted with a PC-104 stack augmented with sensors (Figure 1). Autonomous flight is achieved using a *behavior-based* control architecture [16]. Our sensor network platform is the Berkeley Mica Mote [12]. The operating system support for the Motes is provided by TinyOS, an event-based operating system. Our testbed consists of 50 Mote sensors deployed in the form of a regular 11 × 5 grid, see Figure 4.1. An extra Mote sensor is fitted to the helicopter to allow communications with the deployed sensor network and is connected to the helicopter's Linux-based computer. For further details the reader is referred to [5]. Several applications were run onboard the helicopter, depending on the experiment. The `ping` application sends a broadcast message with a unique id once per second and logs all replies along with the associated Mote identifier. This data allows us to measure air-ground connectivity. The `gps` application receives GPS coordinates via a network socket from the helicopter navigation software and broadcasts it. Simple algorithms in each Mote are able to use these position messages to refine an estimate of their location [6].

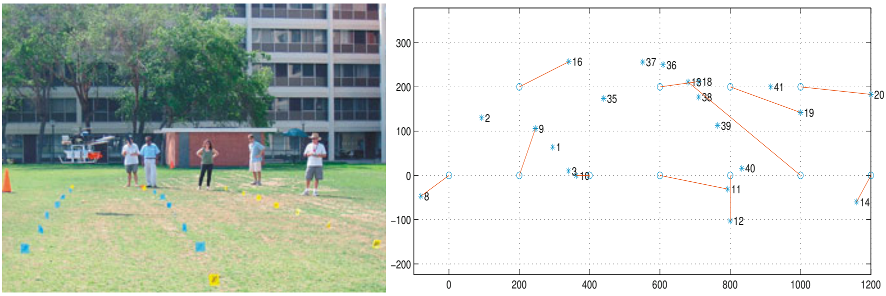


Fig. 3. (Left) The sensor network field with flags marking desired sensor locations. (Right) The locations of a sensor network deployed autonomously by the robot helicopter. The desired locations are denoted by $*$ and they are on a grid. The actual locations are denoted by *o*.

4.2 Experimental Results

Our field experiments have been performed on a grass field on the USC campus (see Figure 3(left)). We marked a 11 × 5 grid on the ground with flags. We used an empirical method to determine the spacing of the grid. We established that on that ground, the Mote transmission range was 2.5 meters. We selected the grid spacing at 2 meters so that we would guarantee communication between any neighbors in the field.

4.3 Deployment and Connectivity Results

Figure 3(Right) shows the desired and actual location of the deployed sensors. The deployment error has multiple causes: (a) error in release location compared to desired location (error due to inherent error present in GPS). (b)error in location of markers on ground compared to release location (wind, downwash, and bounce induced error.)

After being deployed the ground sensors establish autonomously an ad-hoc network whose connectivity topology is shown in Figure 4(Top Left). Although there was error in the deployed location, the resulting network is fully connected. We then manually removed 7 nodes down the center of the network to simulate node failure and create a disconnection in the network. The network automatically computed a new token connectivity map as described in Algorithm 2. Figure 4(Top Right) shows the disconnected components as computed by the token algorithm. Finally the robot helicopter autonomously deployed new nodes to repair connectivity resulting in the connectivity map shown in Figure 4(Bottom). Note that some network links were lost in the final graph. Besides some nodes failing due to being out in the hot sun for a day, the introduction of new nodes results in changes in message timing which changes collision rates and hence overall connectivity, even for nodes remote from the area of repair. Mote communication is inherently unreliable as well. The communication range is dependent on relative antenna orientation, shielding (eg. obstacle between two Motes), ground moisture, current receiver autogain levels, etc. The communication links are asymmetric and congestion is a significant concern. We believe that error, uncertainty, and asymmetry are significant factors that should be explicitly included in any model and approach for networked robotics.

4.4 Localization Results

During localization the flying robot followed a preprogrammed path, see Fig.6(Left). The computer onboard the helicopter obtained its current coordinates and broadcast this via the mote attached to the helicopter once every 100ms. Each ground mote recorded all the X,Y broadcasts it received and used them to compute a centroid based location for itself. Figure 6(Right) shows the helicopter height. Figure 5 shows the location of each of the the motes broadcasts received. It is clear that the motes do not receive messages uniformly from all directions. We speculate that this is due to the non-spherical antenna patterns for transmitter and receiver motes, as well as non-uniform height of the helicopter itself during flights.

5 Conclusion

We have described control algorithms and experimental results from sensor network deployment, localization and subsequent repair of the sensor network with an autonomous helicopter. By sprinkling sensor nodes, we can reach remote or dangerous environments such as rugged mountain slopes, burning forests, etc. We believe that

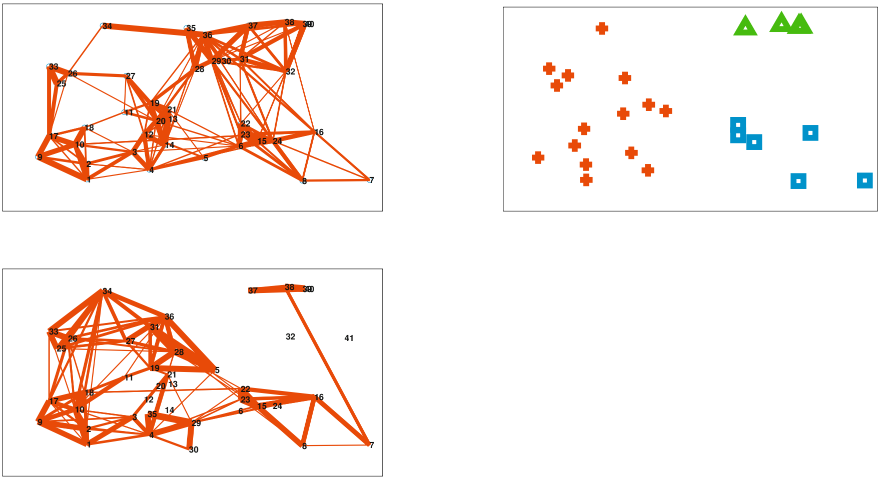


Fig. 4. (Top Left) Connectivity of the initial deployment. (Top Right) Token groups showing connected components after several nodes were removed from the field. (Bottom) Connectivity after the deployment of additional sensor nodes to repair connectivity.

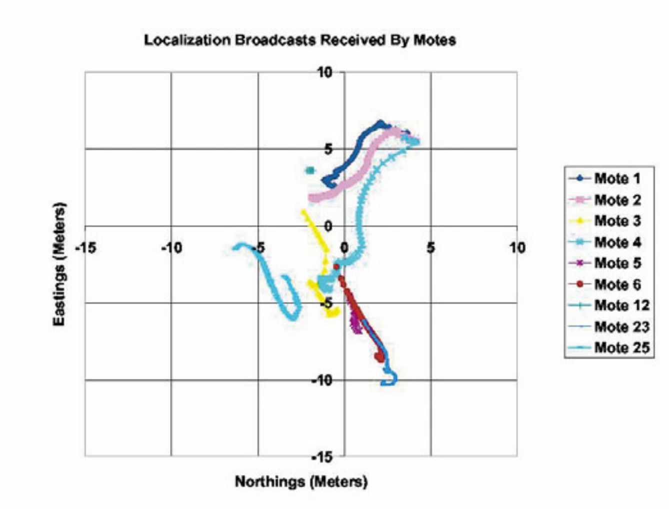


Fig. 5. Location broadcasts heard by some of the motes in the network.

this kind of autonomous approach will enable the instrumentation of remote sites with communication, sensing, and computation infrastructure, which in turn will support navigation and monitoring applications. From what we've learned in these experiments we plan to develop systems for automatic network repair. This will

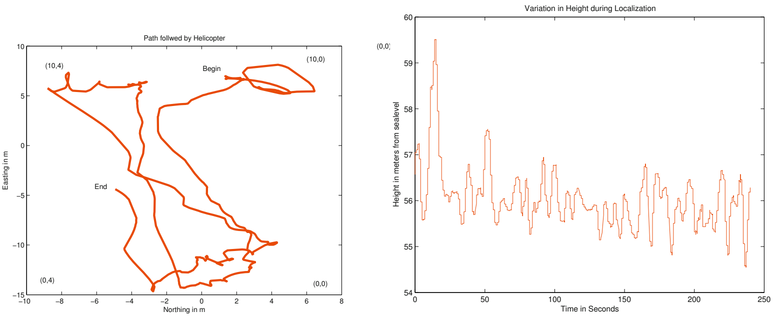


Fig. 6. (Left) The path taken by the helicopter while broadcasting location messages. (Right) The height of the helicopter during the process.

require the ground sensors and helicopter to cooperate to identify network disconnections and guide the helicopter to appropriate locations for autonomous sensor deployment.

Acknowledgment

Support for this work was provided through the Institute for Security Technology Studies, NSF awards EIA-9901589, IIS-9818299, IIS-9912193, EIA-0202789 and 0225446, ONR award N00014-01-1-0675 and DARPA Task Grant F-30602-00-2-0585. This work is also supported in part by NASA under JPL/caltech contract 1231521, by DARPA under grants DABT63-99-1-0015 and 5-39509-A (via UPenn) as part of the Mobile Autonomous Robot Software (MARS) program. Our thanks to our safety pilot Doug Wilson for keeping our computers from (literally) crashing.

References

1. J. Agre and L. Clare. An integrated architeture for cooperative sensing networks. *Computer*, pages 106 – 108, May 2000.
2. M.A. Batalin and G.S. Sukhatme. Spreading out: A local approach to multi-robot coverage. In *Distributed Autonomous Robotic Systems 5*, pages 373–382, 2002.
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. Y. Chen and T. C. Henderson. S-NETS: Smart sensor networks. In *Seventh International Symposium on Experiemental Robotics*, Hawaii, Dec. 2000.
5. P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 1143–8, 2004.
6. P. Corke, R. Peterson, and D. Rus. Networked robots: Flying robot navigation with a sensor network. In *ISRR*, 2003.
7. P. Corke, R. Peterson, and D. Rus. Coordinating aerial and ground robots for navigation and localization. In *Submitted to Distributed Autonomous Robotic Systems*, 2004.

8. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *ACM MobiCom 99*, Seattle, USA, August 1999.
9. V. Gavrilets, I. Martinos, B. Mettler, and E. Feron. Control logic for automated aerobatic flight of miniature helicopter. In *AIAA Guidance, Navigation and Control Conference*, Monterey, CA, USA, Aug 2002.
10. P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, March 2000.
11. J. Hill, P. Bounadonna, and D. Culler. Active message communication for tiny network sensors. In *INFOCOM*, 2001.
12. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In *ASPLOS*, 2000.
13. A. Howard, M.J. Mataric, and G.S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable so lution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5*, pages 299–308, 2002.
14. Q. Li, M. DeRosa, and D. Rus. Distributed algorithms for guiding navigation across sensor networks. In *MOBICOM*, 2003.
15. G. J. Pottie. Wireless sensor networks. In *IEEE Information Theory Workshop*, pages 139–140, 1998.
16. S. Saripalli, J. F. Montgomery, and G. S. Sukhatme. Visually-guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–381, June 2003.
17. S. Saripalli, J. M. Roberts, P. I. Corke, G. Buskey, and G. S. Sukhatme. A tale of two helicopters. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, USA, Oct 2003. (To appear).
18. A. Scaglione and S. Servetto. On the interdependence of routing and data compression in multi-hop sensor networks. In *ACM Mobicom*, Atlanta, GA, 2002.
19. O. Shakernia, 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*, volume 1, pages 128–145, September 1999.
20. University of Southern California Autonomous Flying Vehicle Homepage. http://www-robotics.usc.edu/˜avatar.
21. R. Vidal, O. Shakernia, H. J. Kim, D. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation. *IEEE Transactions on Robotics and Automation*, Oct 2002.

Autonomous Landing of an Unmanned Helicopter based on Vision and Inertial Sensing

Torsten Merz, Simone Duranti, and Gianpaolo Conte

Department of Computer and Information Science
Linköping University, SE-58183 Linköping, Sweden

Abstract. In this paper, we propose an autonomous precision landing method for an unmanned helicopter based on an on-board visual navigation system consisting of a single pan-tilting camera, off-the-shelf computer hardware and inertial sensors. Compared to existing methods, the system doesn't depend on additional sensors (in particular not on GPS), offers a wide envelope of starting points for the autonomous approach, and is robust to different weather conditions. Helicopter position and attitude is estimated from images of a specially designed landing pad. We provide results from both simulations and flight tests, showing the performance of the vision system and the overall quality of the landing.

1 Introduction

Many autonomous landing systems for Unmanned Aerial Vehicles (UAVs) are based on GPS and a dedicated close range sensor for accurate altitude measurement (radar altimeter, sonar, infrared or theodolites). However, in urban environments buildings and other obstacles disturb the GPS signal and can even cause loss of signal (multipath effects, EM noise due to active emitters). Once the GPS signal is lost, the dead reckoning capability of affordable on-board inertial navigation systems does not allow precision navigation for more than few seconds, before diverging. Hence the need of a robust observation of the position: a vision system is self-contained, not jammable, and provides in the proposed implementation a position measurement one order of magnitude more accurate than standard GPS (cm accuracy or better). Estimating velocity from vision is difficult due to limited image frame rate and sensor resolution. In the proposed method velocity is estimated accurately and robustly by fusing vision position with the measurements of inertial sensors that usually belong to the standard instrumentation of an UAV for stabilization. The problem is to develop: (a) a vision system with a sufficient operating range to allow robust pose estimation from a reasonable distance at a sufficient rate with low latency using a landing pad of minimal size; (b) a method to fuse these data with inertial measurements; (c) a suitable flight controller. In an autonomous landing system all components have to match each other. For instance, for calculating vision estimates a proper trade-off between accuracy, range, latency, and rate has to be found optimizing the overall performance of the system.

Our method requires a special landing pad (Fig. 2). As unmanned helicopters usually operate from a designated home base this is not a real constraint. A precise and fast pan/tilt camera is used to extend the range of the vision system and decouple

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 343–352, 2006.

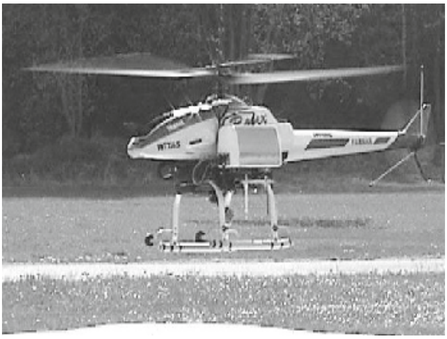


Fig. 1. The WITAS helicopter descending to the landing pad.

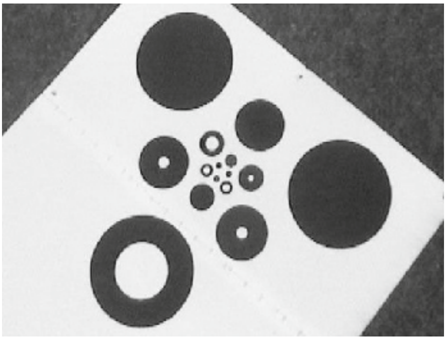


Fig. 2. Landing pad with reference pattern seen from the on-board camera.

the helicopter attitude from the vision field. We developed a single camera solution, as multi-camera systems make the system more complex and expensive and don't offer significant advantages when using known landmarks. For the experimentation we used a helicopter platform (Fig. 1) which had been developed in the WITAS project [2,1].

Vision-based control of small size autonomous helicopters is an active area of research. A good overview of the state-of-the-art can be found in [7]. Our contribution to the landing problem consists of: (a) many demonstrated landings with a system only based on images from a single camera and inertial data using off-the-shelf computer hardware; (b) a wide envelope of starting points for the autonomous approach; (c) robustness to different weather conditions (wind, ambient light); (d) a quantitative evaluation of the vision system and the landing performance.

2 The vision system

The vision system consists of a camera mounted on a pan/tilt unit (PTU), a computer for image processing, and a landing pad (a foldable plate) with a reference pattern on its surface. In this section, we explain the design of the reference pattern, describe the image formation, and present the image processing algorithm.

The reference pattern is designed to fulfill the following criteria: fast recognition, accurate pose estimation for close and distant range, minimum size, and minimal asymmetry. We have chosen black circles on white background as they are fast to detect and provide accurate image features (Fig. 2). From the projection of three circles lying on the corner points of an equilateral triangle the pose of an object is uniquely determined, assuming all intrinsic camera parameters are known. Circles are projected as ellipses, described by the center point $\boldsymbol{u}_e$, the semi-major axis l_a, the semi-minor axis l_b, and the semi-major axis angle θ_e The pose of the landing pad with respect to the camera coordinate system is estimated by minimizing the reprojection error of the extracted center points and semi-axes of the three ellipses. We use five circle triplets of different size (radius 2 to 32 cm, distance 8 to 128 cm) with common center point to achieve a wide range of possible camera positions. Each triplet is uniquely determined by a combination of differently sized inner circles.

A point ${}^p\tilde{\boldsymbol{x}}$ in the landing pad frame is projected on the image plane as follows:

$$\tilde{\boldsymbol{u}} = \boldsymbol{P}\,{}^p\tilde{\boldsymbol{x}} = \begin{pmatrix}\alpha_u & 0 & u_0 & 0\\ 0 & \alpha_v & v_0 & 0\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}{}^c_p\boldsymbol{R} & {}^c\boldsymbol{t}_p\\ \boldsymbol{0}_3^T & 1\end{pmatrix}{}^p\tilde{\boldsymbol{x}} \quad \tilde{\boldsymbol{u}} \in \mathcal{P}^2 \quad {}^p\tilde{\boldsymbol{x}} \in \mathcal{P}^3 \tag{1}$$

The extrinsic camera parameters are given by the three Euler angles of the rotation matrix ${}^c_p\boldsymbol{R}$ and the three components of the translation vector ${}^c\boldsymbol{t}_p$. We use a camera model with the following intrinsic parameters: "focal lengths" α_u and α_v in pixels, principal point (u_0, v_0), and four lens distortion coefficients . All intrinsic parameters are calibrated using Bouguet's calibration toolbox [3]. A conic in $\mathcal{P}^2$ is the locus of all points $\tilde{\boldsymbol{u}}$ satisfying the homogeneous quadratic equation $\tilde{\boldsymbol{u}}^T\boldsymbol{C}\,\tilde{\boldsymbol{u}} = 0$. The transformation of a circle $\boldsymbol{C}_p$ on the landing pad into an ellipse $\boldsymbol{C}_i$ in the image plane is given by[4]:

$$\boldsymbol{C}_i = (\boldsymbol{H}^{-1})^T\boldsymbol{C}_p\,\boldsymbol{H}^{-1} \tag{2}$$

The homography matrix $\boldsymbol{H}$ is the projection matrix $\boldsymbol{P}$ without third column ($z = 0$). We calculate the ellipse center and axes from $\boldsymbol{C}_i$ and represent the parameters in a common feature vector $\boldsymbol{c}$.

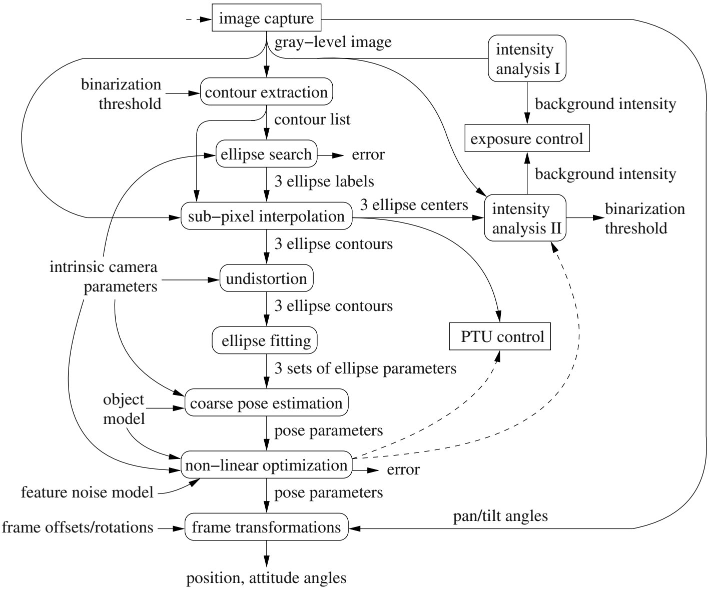


Fig. 3. Dataflow in the vision system.

Fig. 3 shows a data flow diagram of the vision system. Round-edged boxes represent image processing functions, sharp-edged boxes indicate independent processes, and dashed lines show trigger connections. Closed contours are extracted from gray-level images using a fast contour following algorithm with two parameters: edge strength and binarization threshold. The latter is calculated from the intensity distribution of the reference pattern. In the contour list we search for the three biggest ellipses belonging to a circle triplet. Ellipse parameters are estimated by minimizing the algebraic distance of undistorted contour points to the conic using SVD [4,6]. After having found three ellipses, the corresponding contours are resampled with sub-pixel accuracy. A coarse pose is estimated based on the ratio of semi-major axes and circle radii. The estimation is optimized by minimizing the reprojection error:

$$\min_{\boldsymbol{a}} \sum_{i=1}^{12} \left(\frac{d_i - c_{i \bmod 4}}{\sigma_{i \bmod 4}} \right)^2 \quad \begin{aligned} \boldsymbol{d} &= (u_{e1}, v_{e1}, l_{a1}, l_{b1}, \cdots, u_{e3}, v_{e3}, l_{a3}, l_{b3}) \\ \boldsymbol{\sigma} &= (\sigma_c, \sigma_c, \sigma_l, \sigma_l) \\ \boldsymbol{c} &= \left(\hat{u}_e(\boldsymbol{a}), \hat{v}_e(\boldsymbol{a}), \hat{l}_a(\boldsymbol{a}), \hat{l}_b(\boldsymbol{a}) \right) \end{aligned} \tag{3}$$

This function is non-linear and minimized iteratively using the fast-converging Levenberg-Marquardt method [6]. It's initialized with the pose parameters from the first estimate. The uncertainties of the ellipse centers σ_c and axes σ_l are known from separate noise measurements. Finally, the pose parameters are converted to helicopter position and attitude using angles from the PTU and known frame offsets and rotations. The PTU control runs in parallel to the image processing using pixel coordinates of the pattern center as input, aiming at centering the landing pad in the frame as soon as it's localized.

Two methods for analyzing image intensities are implemented. The first estimates the background intensity of the reference pattern based on the assumption being the brightest surface in the image. When the landing pad is detected, the second method is applied. It computes the background intensity and the binarization threshold based on the intensity distribution of the pattern. The exposure controller controls the camera shutter time and iris aiming at keeping the background intensity in a certain range.

3 Sensor Fusion

The position and attitude estimates delivered by the vision system can not be fed directly into the controller due to their intrinsic lack of robustness: the field of view can be temporarily occluded (for example by the landing gear), the illumination conditions can change dramatically just by moving few meters (sun reflections, shades, etc.). On the other hand, vision readings are very accurate, when available.

Hence, a navigation filter based on a Kalman filter (KF) has been developed, fusing highly accurate 3D position estimates from the vision system with inertial data provided by the on-board accelerometers and angular rate gyros. Besides filtering out a large part of the noise and outliers, the filters provides a satisfying dead reckoning capability, sufficient to complete the landing even when the vision system is "blind"[1], see Fig. 8.

[1] During the last 50 cm before touch down the vision system is often "blind" due to two factors: (a) the shade of the helicopter covers part of the pattern at touch down, and (b) when

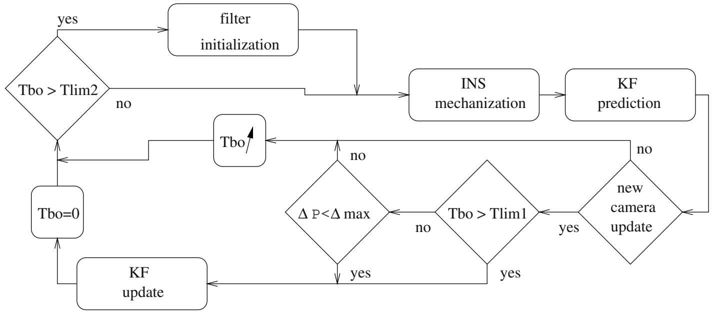


Fig. 4. The filter architecture.

The implementation of the KF is done using the *error state space* or *indirect* formulation with *feedback mechanization* (Fig. 4). The states of the filter are the estimated inertial navigation system (INS) errors. The three observations are given by the difference between the INS position and the position from the vision system (lateral, longitudinal and vertical position relative to the pattern). The advantage of the indirect formulation versus the direct formulation (position, velocity and attitude are among the state variables in the filter) lies in the fact that the INS errors have much slower dynamics than the navigation variables and are very well described by linear equations. In this particular application we have to deal with black-out camera periods in the order of seconds. The indirect KF is quite robust in this respect, in fact an effective indirect filter can be developed with a sample periods of the order of half a minute [5]. The estimated errors are fed back into the mechanization algorithm to avoid unbounded growth of the INS errors.

The inertial measuring unit (IMU) used in this application is integrated in the Yamaha Attitude Sensor (YAS) and it is composed of three accelerometers and three rate gyros. The output rate is 200 Hz for the gyros and 66 Hz for the accelerometers. The filter runs at 50 Hz and the inertial sensors are sampled at the same frequency. Both gyro and accelerometer outputs are prefiltered at 25 Hz to take into account the information available between the filter samples.

The filtered IMU outputs are used in the INS mechanization step to calculate the position, velocity and attitude by solving the inertial navigation equations (4) where $\boldsymbol{r}^n$ and $\boldsymbol{v}^n$ are the position and velocity vectors, $\boldsymbol{C}_b^n$ is the direction cosine matrix of the attitude angles. $\boldsymbol{f}^b$ and $\boldsymbol{\Omega}_{ib}^b$ are the accelerometers and gyros outputs, $\boldsymbol{g}^n$ is the gravity vector and $\boldsymbol{\omega}_{ie}^n$ the Earth rotation rate.

$$\begin{aligned}
\dot{\boldsymbol{r}}^n &= \boldsymbol{v}^n \\
\dot{\boldsymbol{v}}^n &= \boldsymbol{C}_b^n \boldsymbol{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \boldsymbol{v}^n + \boldsymbol{g}^n \quad (4) \\
\dot{\boldsymbol{C}}_b^n &= \boldsymbol{C}_b^n (\boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{in}^b)
\end{aligned}$$

$$\begin{aligned}
\delta\dot{\boldsymbol{r}} &= -\boldsymbol{\omega}_{en} \times \delta\boldsymbol{r} + \delta\boldsymbol{v} \\
\delta\dot{\boldsymbol{v}} &= -(\boldsymbol{\omega}_{ie} + \boldsymbol{\omega}_{in}) \times \delta\boldsymbol{v} - \boldsymbol{\psi} \times f + \delta\boldsymbol{a} \quad (5) \\
\dot{\boldsymbol{\psi}} &= -\boldsymbol{\omega}_{in} \times \boldsymbol{\psi} \\
\delta\dot{\boldsymbol{a}} &= -\beta\delta\boldsymbol{a}
\end{aligned}$$

the distance of the camera to the pattern is very small it is very hard for the controller of the pan/tilt unit to keep the pattern in the picture.

Several filter configurations have been tested, the final implementation is a 12-state KF with 9 navigation error states and 3 accelerometer biases. The dynamic model of the KF is based on the error dynamics equations (5) where $\delta\boldsymbol{r}$, $\delta\boldsymbol{v}$ and $\boldsymbol{\psi}$ are the position, velocity and attitude error vectors and $\delta\boldsymbol{a}$ are the accelerometer biases.

In Fig. 4 the filter architecture is shown. The black-out time (T_{bo}) is the time elapsed since the last valid update. When a new update from the vision system is available, it is first compared with the predicted position and if the difference ($\varDelta P$) is smaller than the maximum allowed tolerance ($\varDelta_{max}$) it is passed to the KF as a new observation. This consistency check is active only if $T_{bo} < T_{lim1}$ because the uncertainty of the system increases each time the update is not made. The KF prediction equations are applied at each time step. When there is a black-out from the vision system the uncertainty of the INS error system represented by its covariance grows unbounded until it's not possible to believe in it anymore. T_{lim1} is the maximum black-out time after that the new vision update is passed directly to the filter. The covariance update equation reduces the covariance of the error system and the decrease is large when the covariance of the measurement is small. The covariance of the vision system measurement is quite small, this means that immediately after the update the uncertainty of the INS error system is low and the consistency check can be activated again. Of course, it can happen that the first update after the black-out is an outlier and in this case it can't be detected. For stability reasons the landing is aborted after a maximum black-out time T_{lim2} and the filter is reinitialized.

4 Flight Controls

The requirements set on the flight control system during landing are the following:

1. The landing mode should be engaged from any point where the landing pad is visible, that means approximately within a 20 m radius hemisphere, centered on the pattern.
2. Once the landing mode is engaged, the helicopter state should be compatible with the proper functionality of the vision system, until touchdown, this means that during the approach phase the following should be considered : (a) the helicopter's position and attitude should not be such as to cause physical occlusion of the visual field; this may happen due to the landing gear skids or the mechanical limitations of the pan/tilt unit; (b) the regions where the accuracy of the vision system is worst should be avoided, if possible; (c) the helicopter velocity and angular rates should not saturate the pan/tilt unit capability for compensation: too high angular rates of the visual beam may result in blurred images; (d) the position of the dominant light source (sun) should be considered, to avoid full reflections.
3. The wind direction has to be taken into account: tailwind landings should be avoided.
4. The control system should be dimensioned for wind levels up to 10 m/s.

Mode	Horizontal control	Yaw control	Altitude control	Logical condition for Mode Transition
READY				
AIM	hold	$\psi \to \psi_{\mathrm{AIM}}$	hold	$\lvert\psi - \psi_{\mathrm{AIM}}\rvert < 5°$
APPROACH	linearly moving to P_D @ $V_{\mathrm{HOR}} < 0.5$ m/s	hold	Descending to PD $h \to h_{D1}$ (5 m) @ VZ = -50 cm/s	$\lvert P - P_{D1}\rvert < 2$ m $\lvert h - h_{D1}\rvert < 0.4$ m $\lvert V_Z\rvert < 10$ cm/s $\lvert V_{\mathrm{HOR}}\rvert < 0.3$ m $\lvert\psi - \psi_{\mathrm{AIM}}\rvert < 3°$
ALIGN	hold	$\psi \to \psi_{\mathrm{TOUCHDOWN}}$	Hold	$(\psi - \psi_{\mathrm{TOUCHDOWN}}) < 3°$
DESCEND	hold	hold	$h \to h_{D2}$ (1 m) @ VZ = -20 cm/s	$\lvert P - P_{D2}\rvert < 0.25$ m $\lvert h - h_{D2}\rvert < 0.1$ m $\lvert V_{\mathrm{HOR}}\rvert < 0.3$ m $\lvert V_Z\rvert < 0.1$ m/s vision data VALID
TOUCH DOWN	hold	hold	VZ = -20 cm/s	h < 0.1 m
SHUT OFF	hold	hold	Throttle back descending ramp	

Fig. 5. Mode sequence leading to touch down. Note that all transitions are one-directional: once a mode is exited it can not be re-entered.

5. The engine should be shut down autonomously, once touch-down is detected. The detection should be timely, since early detections cause high touch down loads and late detections can cause ground resonance.
6. The vertical velocity at touch down should be of the same order of magnitude as a proper manual landing.

In the following, the landing procedure is described. Fig. 5 shows the sequence of control modes and the triggering conditions. As soon as the navigation filter provides state estimates, the helicopter turns towards the pattern to guarantee occlusion-free view of the pattern and flies to a point located 5 meters on the vertical of the desired touch down point (P_{TD}). Once the helicopter is on top of P_{TD}, the heading is changed for landing, taking into consideration the illumination conditions (sun from the side is optimal) and the wind conditions (optimum with head-wind). The final descent is conducted at a constant sink rate of 20 cm/s. At 8 cm from the ground, the throttle command is ramped down, inducing loss of lift and touch down.

The control laws of the helicopter consist of an inner loop (pitch, roll and yaw angle control, and vertical velocity control) and an outer loop (position and velocity control). The inner loop consists of the Yamaha Attitude Control System (YACS), the properties of which have been identified with a dedicated system identification session. The control equations of the outer loop can be summarized as following:

$$
\begin{aligned}
\theta_{\mathrm{C}} &= K_{px}\delta X + K_{pvx}\delta V_X + K_{ivx}\delta V_{X\mathrm{sum}} \\
\Delta\phi_{\mathrm{C}} &= K_{py}\delta Y + K_{pvy}\delta V_Y + K_{ivy}\delta V_{Y\mathrm{sum}} \\
V_{Z\mathrm{C}} &= K_{ivz}\delta V_{Z\mathrm{sum}} + K_{pvz}(V_{Z\mathrm{target}} - V_Z) \\
&\quad V_{Z\mathrm{target}} = \mathrm{limit}(0.75\delta Z, V_{Z\mathrm{min}}, V_{Z\mathrm{max}}) \\
\omega_{\mathrm{C}} &= \mathrm{limit}(K_{pw}\delta\psi, -26\,\mathrm{deg/s}, 26\,\mathrm{deg/s})
\end{aligned}
\tag{6}
$$

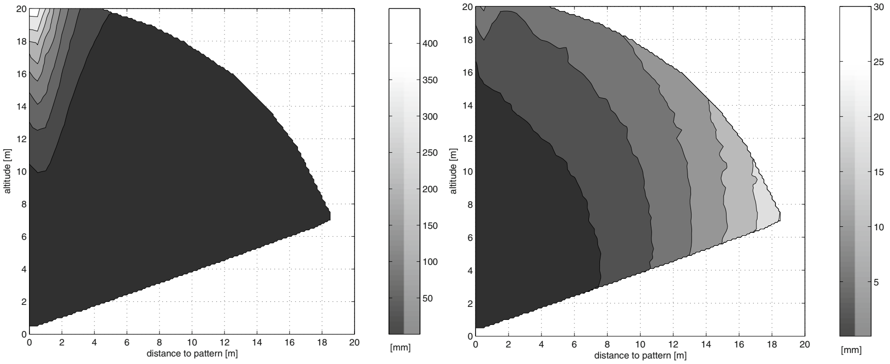


Fig. 6. RMS error in horizontal (left) and vertical position (right) from simulation.

where the subscripted K are control gains, the δ are control errors, the subscript *sum* indicates the integral terms, θ_C is the commanded pitch angle, $\Delta\phi_\mathrm{C}$ is the commanded roll angle variation, ω_C is the commanded yaw rate and $V_{Z\mathrm{C}}$ is the commanded vertical velocity[2].

5 Experimental Results

The helicopter used for experimentation is a slightly modified Yamaha RMAX (Fig. 1). It has a total length of 3.6 m (incl. main rotor) and a take-off weight of 95 kg, including 30 kg available for payload. The vision navigation system consists of two PC104 stacks with PIII 700 MHz processors, the inertial sensors of the YAS, and a single standard CCD camera with approx. 45 degrees horizontal angle of view which is mounted on an off-the-shelf pan/tilt unit (PTU). One of the two computers is dedicated to sensor management and low level control of the helicopter, the other one for image processing and control of the camera and the PTU. The two computers communicate over a RS232C serial link. They are built inside a shock and vibration isolated box, which also includes a precision GPS, a barometric altitude sensor, a compass, a video recorder, a video transmitter, and a wireless Ethernet bridge. The PTU is mechanically limited to 111 degrees tilt and ±180 degrees pan, the max. angular rate is 300 degrees/s and the resolution 0.051 degrees. It is mounted on a vibration isolating platform on the underside of the helicopter body.

We estimated the RMS error of the vision system in position and attitude depending on the relative position to the pattern in leveled flight. For each position 1000 samples were generated using a feature noise model that included noise from image formation, digitization, and segmentation. We developed a method to analyze noise in ellipse center position and semi-axis length. Errors introduced by the

[2] During the descent and touch down phases, the gains of the velocity terms (K_{pvx} and K_{pvx}) are increased by one fifth and the integral terms in the horizontal control are activated, for faster and more precise position control.

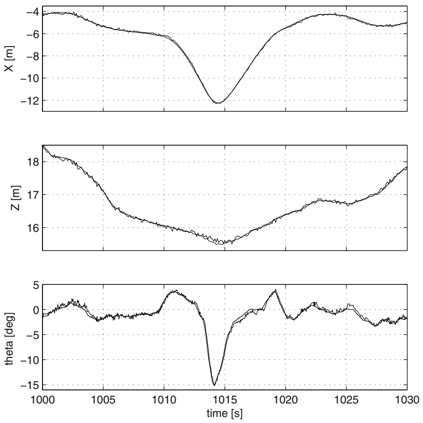


Fig. 7. Position and heading estimates from the vision system vs. estimates based on YAS and precision RTK GPS observations.

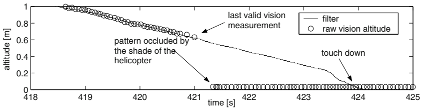


Fig. 8. Altitude estimates from the navigation filter when losing vision.

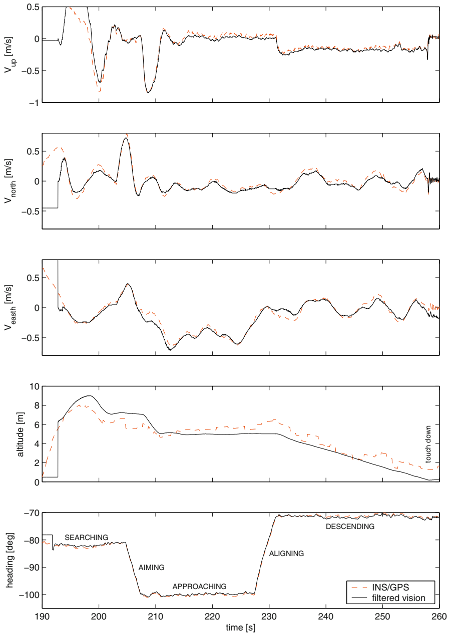


Fig. 9. Time histories and control modes during autonomous landing. The dashed lines show readings from a standard INS/DGPS unit.

transformation from the camera frame into the body frame were not considered in simulation. Fig. 6 shows the RMS errors (1σ) in horizontal and vertical position. The error doesn't change much in the simulated envelope (a 20 m radius hemisphere, with a "blind" sector from azimuth 0 to 15 degrees) due to different triplet sizes and sub-pixel feature extraction. For pitch and roll the error behaves similar to the horizontal error, with a maximum RMS value of $\approx$1 degree, the error in heading is negligible.

The actual accuracy of the vision system was evaluated through an in-flight comparison with a navigation solution based on the YAS and a precision RTK GPS which supplied horizontal/vertical position with 10 mm/15 mm uncertainty (1σ). We found good agreement between measured and simulated errors. Fig. 7 shows timeplots for distance, altitude, and pitch angle at typical starting points for autonomous landing. Position and attitude estimates were provided with an average rate of 20 Hz (using 384×288 pixels images) and an average latency of 110 ms (including delays from capturing PAL signals).

Some tens of autonomous landings were conducted from different relative positions to the landing pad within the specified envelope, on grass and snow fields, with different wind and illumination conditions. A sample of the results is available in Fig. 10 . The vertical velocity at touch down ranged between 18 and 35 cm/s, this corresponds to load factors of about 1.4 g on grass fields. The horizontal velocity

Test ID	Wind Speed [km/h]	V_Z [cm/s]	V_{NORTH} [cm/s]	V_{EAST} [cm/s]	V_{HOR} [cm/s]	P_{TD} MEASURED (X_{TD}, Y_{TD}) [cm]	P_{TD} TARGET (X_{TD}, Y_{TD}) [cm]	$P_{TD\text{-}ERROPR}$ [cm]	ψ_{TD} [°]	$\psi_{TD\text{-}TARGET}$ [°]
128/2/1	15	**-31**	-5	-24	**25**	(51,-17)	(94,-34)	**46**	161	160
128/3/1	14	**-37**	-20	17	**26**	(49,-37)	(94,-34)	**45**	162	160
128/4/1	13	**-35**	0	-1	**1**	(43,-31)	(87,-50)	**48**	151	150
128/4/2	13	**-24**	-23	-13	**26**	(91,-28)	(87,-50)	**22**	152	150
506/5/1	30	**-28**	-7	0	**7**	(41,-33)	(50,-87)	**54**	122	120
506/6/1	22	**-25**	-5	-11	**12**	(47,-53)	(50,-87)	**34**	123	120
506/6/2	26	**-27**	1	0	**1**	(28,-41)	(50,-87)	**50**	121	120
506/8/1	30	**-18**	-7	-17	**18**	(63,-47)	(50,-87)	**42**	121	120

Fig. 10. Flight test results from several autonomous landings.

at touch down was in the order of magnitude of 15 cm/s. The average touch down point precision was about 42 cm (13 % of rotor diameter). Thanks to the pan/tilt camera and a robust controller, considerable wind levels can be handled. Successful landings have been performed with wind levels on ground up to 30 km/h (2 min average), with gusts of 45 km/h.

6 Acknowledgements

We thank Piotr Rudol and Mariusz Wzorek for providing the pan-tilt controller and Anke Knöppler for work on camera calibration. This work was supported by the Wallenberg Foundation, Sweden.

References

1. P. Doherty. Advanced research with autonomous unmanned aerial vehicles. In *Proc. of the 9th International Conference on the Principles of Knowledge Representation and Reasoning*, pages 731–732, June 2004.
2. P. Doherty, P. Haslum, F. Heintz, T. Merz, T. Persson, and B. Wingman. A distributed architecture for autonomous unmanned aerial vehicle experimentation. In *Proc. of the 7th International Symposium on Distributed Autonomous Robotic Systems*, pages 221–230, June 2004.
3. Jean-Yves Bouguet. Camera Calibration Toolbox for Matlab . http://www.vision.caltech.edu/bouguetj/calib_doc.
4. K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1995.
5. P.S. Maybeck. Stochastic models, estimation, and control. In *Mathematics in Science and Engineering*, volume 141-1, pages 289–367. Academic Press, 1979.
6. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.
7. S. Saripalli, J.F. Montgomery, and G. Sukhatme. Visually-guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–380, June 2003.

Synergies in Feature Localization by Air-Ground Robot Teams

Ben Grocholsky, Selcuk Bayraktar, Vijay Kumar, Camillo J. Taylor, and George Pappas

GRASP Laboratory
University of Pennsylvania
{bpg,selcuk,kumar,cjtaylor,pappasg}@grasp.upenn.edu
http://www.grasp.upenn.edu

Abstract. This paper describes the implementation of a decentralized architecture for autonomous teams of aerial and ground vehicles engaged in active perception. We provide a theoretical framework based on an established approach to the underlying sensor fusion problem [3]. This provides transparent integration of information from heterogeneous sources. The approach is extended to include an information-theoretic utility measure that captures the task objective and robot inter-dependencies. A distributed solution mechanism is employed to determine *information maximizing* trajectories and assignments subject to the constraints of individual vehicle and sensor sub-systems. This architecture enables the benefit of the complementary aerial and ground based vehicle and sensor capabilities to be realized. The approach is applied to missions involving searching for and tracking multiple ground targets. Experimental results for vehicles equipped with cameras are presented. These illustrate the impact of the team configuration on overall system performance.

1 Introduction

Aerial and ground vehicles exhibit complementary capabilities and characteristics as robotic sensor platforms. Fixed wing aircraft offer broad field of view and rapid coverage of search areas. However, minimum limits on operating airspeed and altitude, combined with attitude uncertainty, place a lower limit on their ability to resolve and localize ground features. Ground vehicles on the other hand offer high resolution sensing over relatively short ranges with the disadvantage of slow coverage. This paper presents a decentralized architecture and solution methodology for seamlessly realizing the collaborative potential of air and ground robots. Experimental results using robot with complementary capabilities detailed in Table 1, demonstrate rapid localization of ground features alleviating the requirement for a time consuming extensive search by ground vehicles.

This paper is organized as follows. Following a review of related work, Section 3 details the technical approach taken and the system architecture. The experimental setup and hardware along with sensor modelling and controller implementation for a collaborative feature localization task are described in Section 4. Experimental results are presented and discussed in Section 5 followed by concluding remarks.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 352–362, 2006.

2 Related Work

This work builds on previous endeavors in decentralized data fusion [6,5] and active sensor networks [3]. The established architecture and methodology is used here. Approaches to active sensing that implement alternative system architectures and techniques for estimation and control include [7,12,9]. The use of aerial and ground based sensor platforms is closely related to other efforts to exploit the truly complementary capabilities of air and ground robots. Examples of such initiatives include the DARPA PerceptOR program [8] and Fly Spy project [11]. While exceptional results have been obtained with advanced airborne sensors such as aerial laser scanning [10], the combined use of air and ground active sensing offers high resolution awareness from relatively low cost visual sensors. The approach presented enables this collaborative potential to be realized through seamless integration and refinement of information from heterogeneous robot sensor systems.

3 System Architecture and Approach

The approach taken builds on established methods in decentralized data fusion (DDF): the information form of the Kalman filter. This methodology has previous been applied to localization of ground features by aerial sensor platforms [6]. Ground targets are modeled as point features with corresponding position estimated by decentralized information filter implementation. The underlying equations are presented briefly. See [5] for a full derivation.

Consider a system described by the discrete time state and observation processes

$$\mathbf{x}(k) = \mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{G}(k)\mathbf{w}(k), \quad \mathbf{z}(k) = \mathbf{h}(k, \mathbf{x}(k)) + \mathbf{v}(k) \tag{1}$$

where the process noise $\mathbf{w}(k)$ and observation noise $\mathbf{v}(k)$ are uncorrelated white sequences $\mathbf{w} \sim \mathcal{N}(0, Q)$ and $\mathbf{v} \sim \mathcal{N}(0, R)$. The information filter is obtained by replacing the representation of the state estimate $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$ with the information state $\hat{\mathbf{y}}$ and Fisher information $\mathbf{Y}$. Notation $(i \mid j)$ indicates a value at time i, conditional on observation information obtained up to time j. The information state and information matrix are defined as

$$\hat{\mathbf{y}}(i \mid j) \triangleq \mathbf{P}^{-1}(i \mid j)\hat{\mathbf{x}}(i \mid j), \quad \mathbf{Y}(i \mid j) \triangleq \mathbf{P}^{-1}(i \mid j). \tag{2}$$

The information vector and matrix contributions associated with an observation are

$$\begin{aligned} \mathbf{i}(k) &\triangleq \mathbf{H}^T(k)\mathbf{R}^{-1}(k)(\mathbf{z}(k) - \mathbf{h}(\hat{\mathbf{x}}(k \mid k-1)) + \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1), \\ \mathbf{I}(k) &\triangleq \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k). \end{aligned} \tag{3}$$

where $\mathbf{H}^T(\cdot)$ is the Jacobian $\nabla_{\mathbf{x}}\mathbf{h}(\cdot)$. With these definitions, the information filter can be summarized in two stages as:

Prediction:

$$\begin{aligned} \mathbf{Y}(k \mid k-1) &= \left[\mathbf{F}(k)\mathbf{Y}^{-1}(k-1 \mid k-1)\mathbf{F}^T(k) + \mathbf{Q}(k)\right]^{-1}, \\ \hat{\mathbf{y}}(k \mid k-1) &= \mathbf{Y}(k \mid k-1)\mathbf{F}(k)\mathbf{Y}^{-1}(k-1 \mid k-1)\hat{\mathbf{y}}(k-1 \mid k-1). \end{aligned} \tag{4}$$

Estimation:

$$\begin{aligned}\mathbf{Y}(k \mid k) &= \mathbf{Y}(k \mid k-1) + \textstyle\sum_{i=1}^{N} \mathbf{I}_i(k),\\ \hat{\mathbf{y}}(k \mid k) &= \hat{\mathbf{y}}(k \mid k-1) + \textstyle\sum_{i=1}^{N} \mathbf{i}_i(k).\end{aligned} \tag{5}$$

where $\mathbf{I}_i(k)$ and $\mathbf{i}_i(k)$ are the information matrix and information state contributions of the sensors $i = 1, \dots, N$. The posterior state estimate may be obtained from

$$\hat{\mathbf{x}}(k \mid k) = \mathbf{Y}^{-1}(k \mid k)\hat{\mathbf{y}}(k \mid k). \tag{6}$$

The additive structure of the estimation Equation 5 results in a remarkably simple decentralized architecture. As in [3], a control layer is implemented above the DDF framework. Figure 1 details the structure of the DDF and control implementation in each active sensing node. Mutual information gain is used as a control objective in order to generate robot sensing trajectories that seek to maximize the reduction of estimate uncertainty. This utility measure applied to a ground feature localization task is depicted in Figure 2.

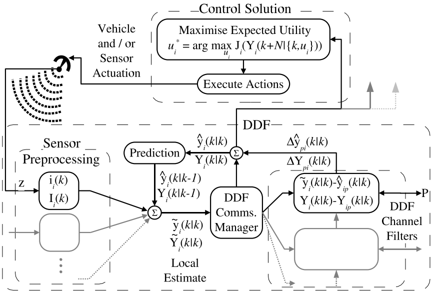


Fig. 1. Active sensing node structure.

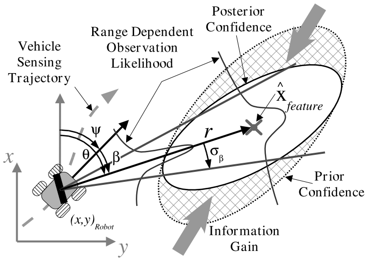


Fig. 2. Ground vehicle mutual information gain utility measure

Instances of the active sensing node may be composed to form proactive teams of networked robotic sensors. This is the basis of the approach taken to active ground feature localization by collaborative aerial and ground robots. All air and ground vehicles execute an instance of the DDF node detailed in Figure 1. The search area specified by system operator is divided into search patterns to be executed by the available aerial vehicles. Upon sighting potential ground features, associated new filters are pushed onto the network of DDF nodes. This exchange provides cues to the ground vehicles actively seeking to reduce the estimate uncertainty. Ground vehicles see these uncertain features influencing their utility and the resulting trajectories.

4 Experimental Setup

This section describes the elements involved in the collaborative air-ground feature localization experiments. The robot platforms, feature localization filter, sensor modeling and control implementation are detailed.

4.1 Robot Platforms

The approach detailed in Section 3 has been implemented on the aerial and ground robot test-beds shown in Figure 3. The Ground vehicles are a commercial 4WD model truck modified and augmented with an on-board computer, stereo firewire camera, GPS and inertial sensors as described in [2]. The aerial vehicles are quarter scale Piper Cub model aircraft equipped with the Piccolo autopilot by Cloud Cap Technology (see [1] for further details). In addition to the sensors within the autopilot, the air vehicles carry a sensor pod containing a high resolution firewire camera, inertial sensors and a 10Hz GPS receiver. A spread-spectrum radio modem is used for Communications between air vehicles and the operator base station. The ground vehicles and base station communicate through an Ad-Hoc 802.11b network.

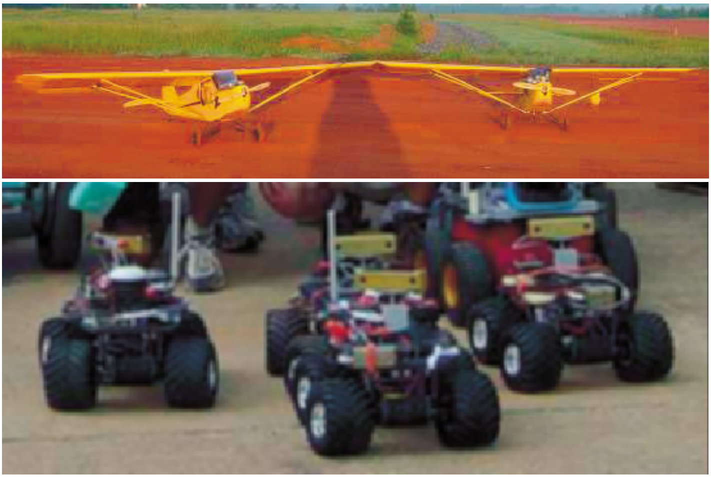


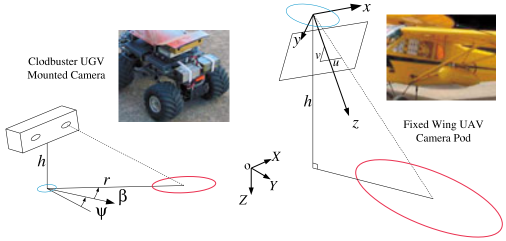
Fig. 3. Fixed wing UAVs (top) and ground robot platforms (bottom).

4.2 Feature Localization Filter

The ground features are modeled as two-dimensional stationary points in a plane at known altitude represented by Gaussian random variables as depicted in Figure 2. There is no process dynamics, *i.e.*, $\mathbf{F}(k) = \mathbf{I}_{2\times 2}$ and no process noise, *i.e.*, $\mathbf{Q}(k) = \mathbf{0}_{2\times 2}$. Each filter node maintains a list of active and potential features. Detection is based on extracting indistinguishable colored features from camera images. A data association process using Chi-square testing is performed on incoming feature observations. A new potential filter is created for observations that fail to match existing filters. Potential filters that receive a set number of associated observations are promoted to active status and propagated throughout the DDF network.

4.3 Sensor Modeling and Platform Capabilities

Projective geometry is used to determine the observed location of ground features from measurements obtained using the camera installations shown in figure 4. The projection matrix $\mathcal{P}$ for each camera is obtained from pre-determined intrinsic parameters and current estimates of the camera rotation and translation.



$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathcal{P} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \tag{7}$$

Fig. 4. Modelling aerial and ground based vision sensors.

Ground features are considered to be at a common known altitude Z. Equation 7 is solved to determine the location of feature observations in a global XY plane from the measurements u and v in the camera coordinates. Estimates of the camera measurement noise and attitude uncertainty are propagated through this relationship and degraded by the platform position uncertainty to determine an approximate feature observation covariance.

Fusion of GPS, inertial and encoder measurements allows the ground vehicles to determine their position with greater certainty than the aerial platforms using GPS alone. A summary of the system capabilities for the air and ground vehicles is presented in Table 4.3. This illustrates the complementary attributes of these sensor platforms. Airborne sensors can cover large areas and distances to make a small number of relatively uncertain observations. While significantly more accurate, ground platforms travel slowly and offer limited field of view.

Table 1. Summary of sensor and vehicle capabilities

	Clodbuster UGV	Fixed Wing UAV
Sensor Height	0.3m	65m
Sensor Range	5m	50m
Observation Uncertainty	0.2 $\sim$ 0.5m	6 $\sim$ 8m
Vehicle Velocity	0.5 $\sim$ 1m/s	15m/s

4.4 UGV Controller Implementation

A controller is implemented on the ground vehicles to generate sensing trajectories that actively reduce the uncertainty in feature estimates. A gradient control law is obtained by considering action utility with zero look-ahead rather than planning actions over time. The instantaneous mutual information rate for the estimation process is [4]

$$\mathcal{I}(t) = \frac{1}{2}\frac{d}{dt}\log\mid \mathbf{Y}(t) \mid = \frac{1}{2}\text{trace}\left(\mathbf{Y}^{-1}(t)\dot{\mathbf{Y}}(t)\right) \tag{8}$$

where $\mathbf{Y}$ is the block diagonal information matrix corresponding active features with uncertainty above a desired threshold. For the process model considered, $\dot{\mathbf{Y}}(t)$ is equal to the sensor observation information $\mathbf{I}(t)$ in Equation 3. Given that this observation information depends on the system configuration through the nonlinear observation model, Equation 8 relates the sensor system state $\mathbf{x_R} = [x_R, y_R, \psi_R]^T$ as indicated in Figure 5, to a time varying field equal to the instantaneous rate of change of feature uncertainty. The gradient of this field can be evaluated in terms of the current Fisher information and the partial derivatives of the observation information with respect to $\mathbf{x_R}$ by

$$\nabla_{\mathbf{x_R}} \mathcal{I}(t) = \frac{1}{2} \text{trace} \left(\mathbf{Y}^{-1}(t) \nabla_{\mathbf{x_R}} \mathbf{I}(t) \right). \tag{9}$$

Analytic expressions are available for the models considered here. Control actions can be scheduled according to the direction and magnitude of the local gradient. A simple control solution is implemented by driving at a fixed velocity while heading in the direction of steepest gradient $\psi^\star(t)$

$$\psi^\star(t) = \arctan \left(\frac{\nabla_{y_R} \mathcal{I}(t)}{\nabla_{x_R} \mathcal{I}(t)} \right). \tag{10}$$

When implemented on a nonholonomic robot with constraints imposed on the vehicle turn rate and sensor field of view, this controller may result in the robot circling a feature while unable to make observations. To resolve this, the controller is disengaged when the expected feature location is within the turn constraint and outside the field of view as illustrated in Figure 6.

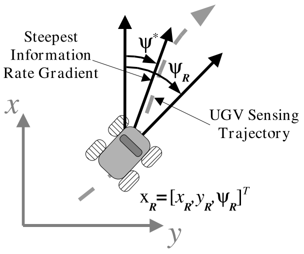


Fig. 5. The ground vehicle turn controller tracks the gradient of feature information gain.

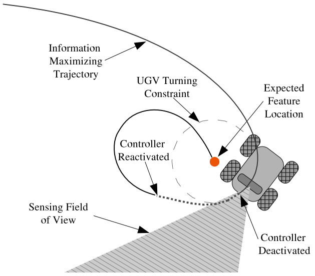
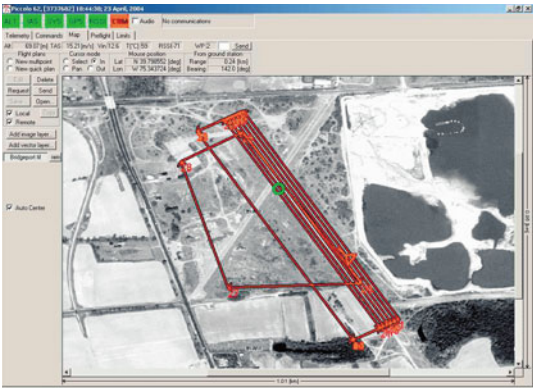


Fig. 6. Handling ground vehicle sensing field of view and control constraints.

5 Results

Results are presented for an experimental investigation of a collaborative feature localization scenario. Three rectangular orange features each measuring 1.1×1.4 meters were placed in a 50×200 meter search area. Figure 7 details the search trajectory generated for the aerial vehicle to cover this area in multiple passes. The elapsed time for each pass was approximately 100 seconds. A sequence of images


Piccolo 62, [3737682] 18:44:38; 23 April, 2004

Alt: 69.07 [m] TAS 15.21 [m/s] Vin 12.6 T[°C] 58 RSSI 71 WP 2

Mouse position: Lat N 39.798552 [deg] Lon W 75.343724 [deg]

From ground station: Range 0.24 [km] Bearing 142.0 [deg]

Fig. 7. Example single UAV search pattern at the Bridgeport Airport, New Jersey test site.

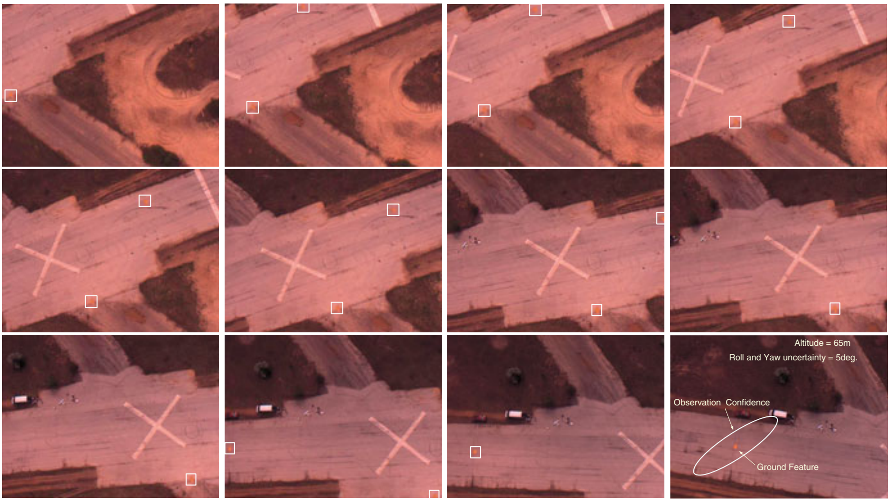


Fig. 8. Aerial images of the test site captured during a typical UAV flyover at 65 meters altitude. Three orange ground features highlighted by white boxes are visible during the four second pass. The confidence ellipse associated with a feature observation is indicated in the last image. This represents the compounded uncertainty due to errors in UAV attitude, UAV position and camera calibration.

captured from an altitude of 65 meters is shown in Figure 8. The feature estimates are made available to the ground vehicles seamlessly thought the DDF network.

Figure 9 illustrates the initial feature uncertainty and the trajectory taken by the ground vehicle to refine the quality of these estimates. Detailed snapshots of the active sensing process are shown in Figure 10. These indicate the proposed control scheme successfully positioning the ground vehicle to take advantage of the on-board sensor characteristics.

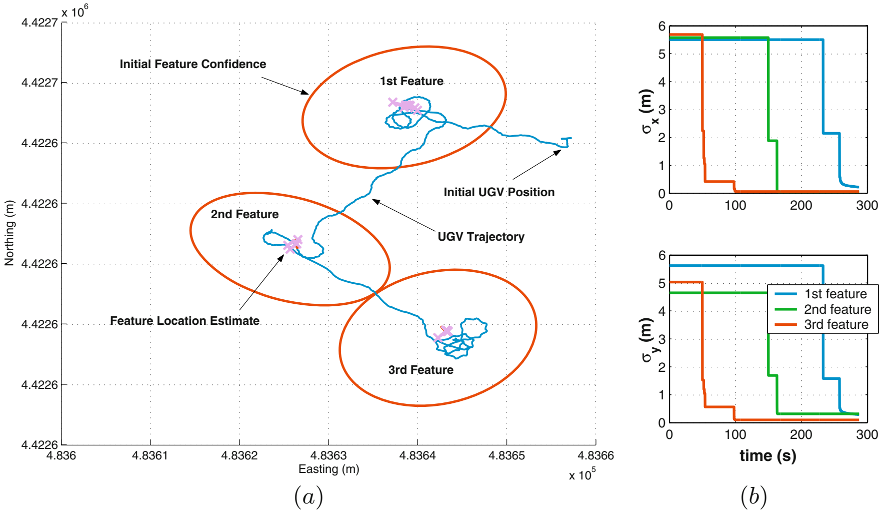


Fig. 9. Figures indicating (a) Initial feature confidence and UGV active sensing trajectory, (b) σ_x and σ_y components of feature estimate standard deviation over time.

It is important to note the performance benefit obtained through collaboration. Assuming independent measurements, in excess of fifty passes (about eighty minutes flight time) are required by the UAV to achieve this feature estimate certainty. It would take in excess of half an hour for the ground vehicle with this speed and sensing range to cover the designated search area and achieve a high probability of detecting the features. The collaborative approach using aerial cues to active ground sensing completes this task in under 10 minutes. A performance level well in excess of the individual system capabilities.

6 Future Work

The architecture presented could be applied to teams of multiple ground and air vehicles without change. However, the simple controller implemented here is not expected to achieve the full potential of multi UGV teams. The application of predictive cooperative control strategies that account for the nonholonomic, control and sensing constraints, promises to address this concern. More sophisticated estimation, detection and association schemes should also be considered. This work investigated tasking ground vehicles from cues provided by aerial robots executing predetermined fixed search trajectories. Actively controlling the UAV sensing trajectories will be the subject of future research.

7 Conclusion

This paper presented a consistent architecture and approach for enabling proactive collaboration among aerial and ground based sensor platforms. The architecture

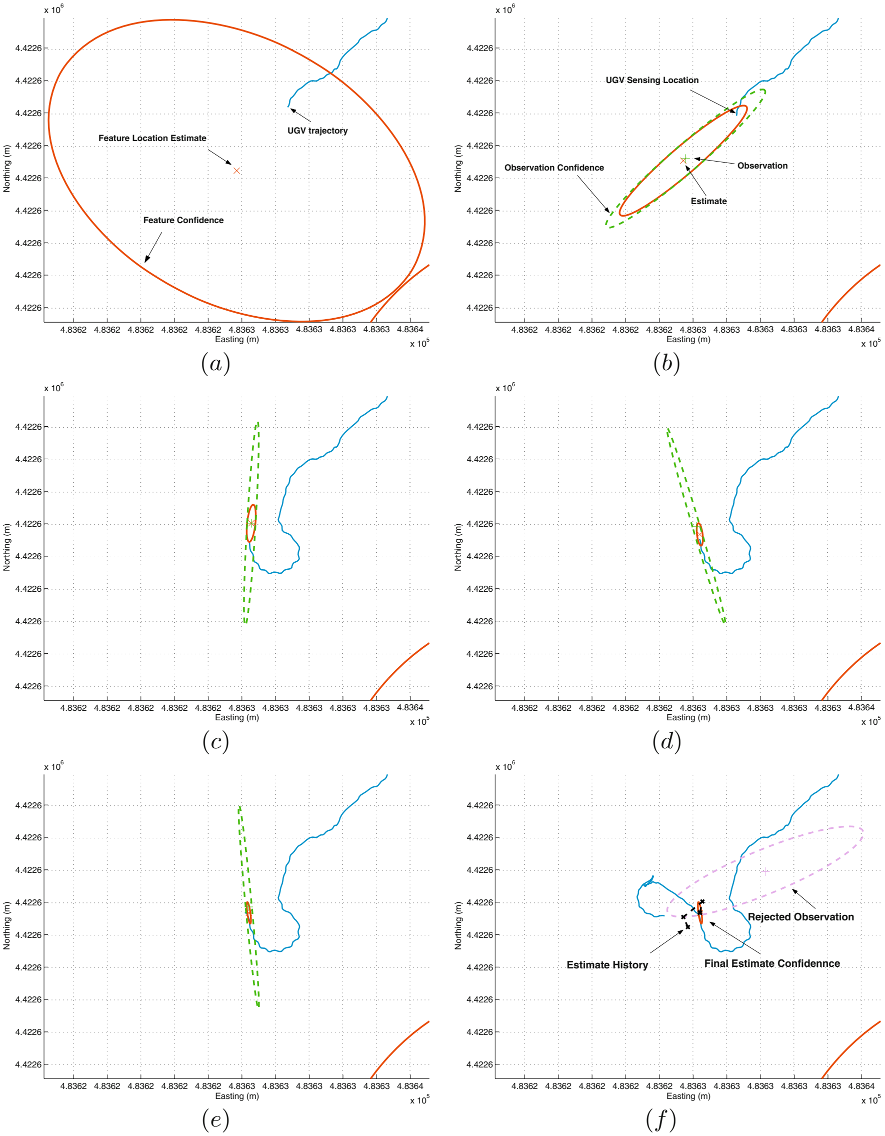


Fig. 10. Snapshots of the active feature location estimate refinement by an autonomous ground robot equipped with vision, GPS, inertial and odometric sensors. This corresponds to the second feature indicated in Figure 9(a). Figure (a) indicates the initial confidence region obtained through aerial sensing alone. Any need for an extensive search by the ground vehicle is alleviated since this confidence region is slightly smaller than the ground vehicle on-board camera effective field of view. Compounded error sources in the ground vehicle sensor system result in feature observations that provide predominantly bearing information as shown in Figures (b) to (e). The controller successfully drives the ground robot to sensing locations orthogonal to the confidence ellipse major axis that maximize the expected reduction in estimate uncertainty. False feature detections are rejected as indicated in Figure (f).

provides seamless integration of air and ground sensor platforms, allowing system elements to be transparently aware of and exploit the collective knowledge and resources. This methodology was applied to a ground feature search and localization problem. Experiments were conducted using fixed wing aerial and ground robot platforms with truly complementary capabilities in terms of sensing accuracy, coverage and speed. Results indicate significant performance benefits are obtained through collaborative air and ground sensing in search and reconnaissance missions.

Acknowledgement

This research is supported by the ARO MURI Grant DAAD 19-02-01-0383, DARPA MARS NBCH1020012 and NSF CCR02-05336. The authors wish to thank Jim Keller, Daniel Gomez-Ibanez, Anthony Cowley, Hwa Chow Hsu, Luiz Chaimowicz, Pranav Srivastava, Eglantine Dang and Rahul Swaminathan for their valuable contributions.

References

1. Selcuk Bayraktar, Georgios E. Fainekos, and George J. Pappas. Experimental cooperative control of fixed-wing unmanned aerial vehicles. In *IEEE Conf. Decision and Control (submitted)*, 2004.
2. A. Das, J. Spletzer, , V. Kumar, and C. Taylor. Ad hoc networks for localization and control. In *Proc. of the 41st IEEE Conf. on Decision and Control, Las Vegas, NV*, 2002.
3. B. Grocholsky, A. Makarenko, T. Kaupp, and H.F. Durrant-Whyte. Scalable control of decentralised sensor platforms. In *Information Processing in Sensor Networks: 2nd Int Workshop, IPSN03*, pages 96–112, 2003.
4. J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with applications in statistics and econometrics*. John Wiley and Sons, New York, 1988.
5. J. Manyika and H.F. Durrant-Whyte. *Data Fusion and Sensor Management: An Information-Theoretic Approach*. Prentice Hall, 1994.
6. M. Ridley, E. Nettleton, S. Sukkarieh, and H. Durrant-Whyte. Tracking in decentralised air-ground sensing networks. In *Proc. of the Fifth Int. Conference on Information Fusion*, volume 1, pages 616–623, July 2002.
7. J. Spletzer and C.J. Taylor. Dynamic sensor planning and control for optimally tracking targets. *International Journal of Robotics Research*, 22:7–20, 2003.
8. T. Stentz, A. Kelly, H. Herman, P. Rander, and Mandelbaum R. Integrated air/ground vehicle system for semi-autonomous off-road navigation. In *Proceedings of AUVSI Symposium on Unmanned Systems*, 2002.
9. Ashley W. Stroupe, Ramprasad Ravichandran, and Tucker Balch. Value-based action selection for exploration and dynamic target observation with robot teams. In *Proc. of the IEEE Int. Conf. on Robotics & Automation*, pages 4190–4197, April 2004.
10. S. Thrun, Mark Diel, and D. Hähnel. Scan alignment and 3d surface modeling with a helicopter platform. In *Proceedings of the International Conference on Field and Service Robotics*, Lake Yamanaka, Japan, 2003.
11. R. Vaughan, G. Sukhatme, J. Mesa-Martinez, and J. Montgomery. Fly spy: lightweight localization and target tracking for cooperating ground and air robots. In *Proc. Int. Symp. Distributed Autonomous Robot Systems*, 2000.
12. F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing Magazine*, 19(2):61–72, 2002.

Autonomous Inverted Helicopter Flight via Reinforcement Learning

Andrew Y. Ng[1], Adam Coates[1], Mark Diel[2], Varun Ganapathi[1], Jamie Schulte[1], Ben Tse[2], Eric Berger[1], and Eric Liang[1]

[1] Computer Science Department, Stanford University, Stanford, CA 94305
[2] Whirled Air Helicopters, Menlo Park, CA 94025

Abstract. Helicopters have highly stochastic, nonlinear, dynamics, and autonomous helicopter flight is widely regarded to be a challenging control problem. As helicopters are highly unstable at low speeds, it is particularly difficult to design controllers for low speed aerobatic maneuvers. In this paper, we describe a successful application of reinforcement learning to designing a controller for sustained inverted flight on an autonomous helicopter. Using data collected from the helicopter in flight, we began by learning a stochastic, nonlinear model of the helicopter's dynamics. Then, a reinforcement learning algorithm was applied to automatically learn a controller for autonomous inverted hovering. Finally, the resulting controller was successfully tested on our autonomous helicopter platform.

1 Introduction

Autonomous helicopter flight represents a challenging control problem with high dimensional, asymmetric, noisy, nonlinear, non-minimum phase dynamics, and helicopters are widely regarded to be significantly harder to control than fixed-wing aircraft. [3,10] But helicopters are uniquely suited to many applications requiring either low-speed flight or stable hovering. The control of autonomous helicopters thus provides an important and challenging testbed for learning and control algorithms.

Some recent examples of successful autonomous helicopter flight are given in [7,2,9,8]. Because helicopter flight is usually open-loop stable at high speeds but unstable at low speeds, we believe low-speed helicopter maneuvers are particularly interesting and challenging. In previous work, (Ng et al.,2004) considered the problem of learning to fly low-speed maneuvers very accurately. In this paper, we describe a successful application of machine learning to performing a simple low-speed aerobatic maneuver—autonomous *sustained* inverted hovering.

2 Helicopter platform

To carry out flight experiments, we began by instrumenting a Bergen industrial twin helicopter (length 59", height 22") for autonomous flight. This

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 363–372, 2006.

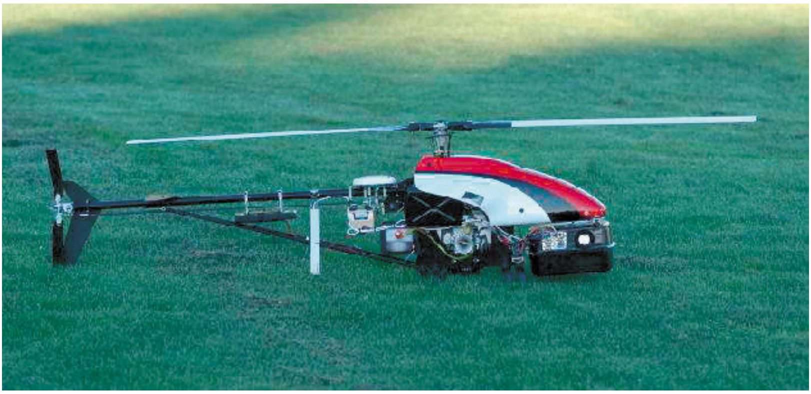


Fig. 1. Helicopter in configuration for upright-only flight (single GPS antenna).

helicopter is powered by a twin cylinder 46cc engine, and has an unloaded weight of 18 lbs.

Our initial flight tests indicated that the Bergen industrial twin's original rotor-head was unlikely to be sufficiently strong to withstand the forces encountered in aerobatic maneuvers. We therefore replaced the rotor-head with one from an X-Cell 60 helicopter. We also instrumented the helicopter with a PC104 flight computer, an Inertial Science ISIS-IMU (accelerometers and turning-rate gyroscopes), a Novatel GPS unit, and a MicroStrain 3d magnetic compass. The PC104 was mounted in a plastic enclosure at the nose of the helicopter, and the GPS antenna, IMU, and magnetic compass were mounted on the tail boom. The IMU in particular was mounted fairly close to the fuselage, to minimize measurement noise arising from tail-boom vibrations. The fuel tank, originally mounted at the nose, was also moved to the rear. Figure 1 shows our helicopter in this initial instrumented configuration.

Readings from all the sensors are fed to the onboard PC104 flight computer, which runs a Kalman filter to obtain position and orientation estimates for the helicopter at 100Hz. A custom takeover board also allows the computer either to read the human pilot's commands that are being sent to the helicopter control surfaces, or to send its own commands to the helicopter. The onboard computer also communicates with a ground station via 802.11b wireless.

Most GPS antenna (particularly differential, L1/L2 ones) are directional, and a single antenna pointing upwards relative to the helicopter would be unable to see any satellites if the helicopter is inverted. Thus, a single, upward-pointing antenna cannot be used to localize the helicopter in inverted flight. We therefore added to our system a second antenna facing downwards, and used a computer-controlled relay for switching between them. By examining the Kalman filter output, our onboard computer automatically selects the upward-facing antenna. (See Figure 2a.) We also tried a system in which

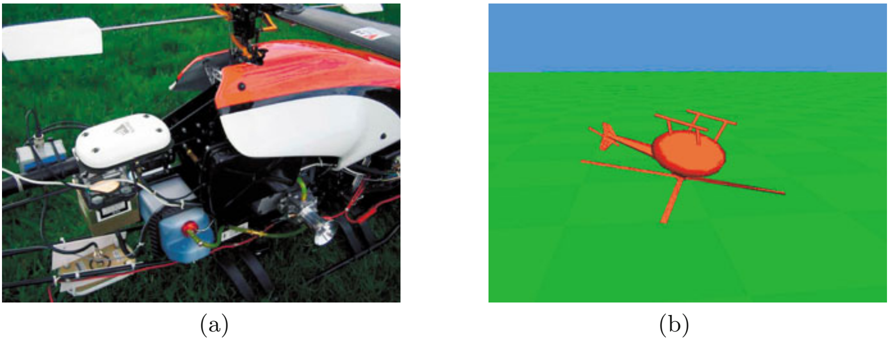


Fig. 2. (a) Dual GPS antenna configuration (one antenna is mounted on the tail-boom facing up; the other is shown facing down in the lower-left corner of the picture). The small box on the left side of the picture (mounted on the left side of the tail-boom) is a computer-controlled relay. (b) Graphical simulator of helicopter, built using the learned helicopter dynamics.

the two antenna were simultaneously connected to the receiver via a Y-cable (without a relay). In our experiments, this suffered from significant GPS multipath problems and was not usable.

3 Machine learning for controller design

A helicopter such as ours has a high center of gravity when in inverted hover, making inverted flight significantly less stable than upright flight (which is also unstable at low speeds). Indeed, there are far more human RC pilots who can perform high-speed aerobatic maneuvers than can keep a helicopter in sustained inverted hover. Thus, designing a stable controller for sustained inverted flight appears to be a difficult control problem.

Most helicopters are flown using four controls:

- $a[1]$ and $a[2]$: The longitudinal (front-back) and latitudinal (left-right) cyclic pitch controls cause the helicopter to pitch forward/backwards or sideways, and can thereby also be used to affect acceleration in the longitudinal and latitudinal directions.
- $a[3]$: The main rotor collective pitch control causes the main rotor blades to rotate along an axis that runs along the length of the rotor blade, and thereby affects the angle at which the main rotor's blades are tilted relative to the plane of rotation. As the main rotor blades sweep through the air, they generate an amount of upward thrust that (generally) increases with this angle. By varying the collective pitch angle, we can affect the main rotor's thrust. For inverted flight, by setting a negative collective pitch angle, we can cause the helicopter to produce negative thrust.
- $a[4]$: The tail rotor collective pitch control affects tail rotor thrust, and can be used to yaw (turn) the helicopter.

A fifth control, the throttle, is commanded as pre-set function of the main rotor collective pitch, and can safely be ignored for the rest of this paper.

To design the controller for our helicopter, we began by learning a stochastic, nonlinear, model of the helicopter dynamics. Then, a reinforcement learning/policy search algorithm was used to automatically design a controller.

3.1 Model identification

We applied supervised learning to identify a model of the helicopter's dynamics. We began by asking a human pilot to fly the helicopter upside-down, and logged the pilot commands and helicopter state s comprising its position (x, y, z), orientation (roll ϕ, pitch θ, yaw ω), velocity $(\dot{x}, \dot{y}, \dot{z})$ and angular velocities $(\dot{\phi}, \dot{\theta}, \dot{\omega})$. A total of 391s of flight data was collected for model identification. Our goal was to learn a model that, given the state s_t and the action a_t commanded by the pilot at time t, would give a good estimate of the probability distribution $P_{s_t a_t}(s_{t+1})$ of the resulting state of the helicopter s_{t+1} one time step later.

Following standard practice in system identification [4], we converted the original 12-dimensional helicopter state into a reduced 8-dimensional state represented in body coordinates $s^b = [\phi, \theta, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\omega}]$. Where there is risk of confusion, we will use superscript s and b to distinguish between spatial (world) coordinates and body coordinates. The body coordinate representation specifies the helicopter state using a coordinate frame in which the x, y, and z axes are forwards, sideways, and down relative to the current orientation of the helicopter, instead of north, east and down. Thus, $\dot{x}^b$ is the forward velocity, whereas $\dot{x}^s$ is the velocity in the northern direction. (ϕ and θ are always expressed in world coordinates, because roll and pitch relative to the body coordinate frame is always zero.) By using a body coordinate representation, we encode into our model certain "symmetries" of helicopter flight, such as that the helicopter's dynamics are the same regardless of its absolute position and orientation (assuming the absence of obstacles).[1]

Even in the reduced coordinate representation, only a subset of the state variables need to be modeled explicitly using learning. Specifically, the roll ϕ and pitch θ (and yaw ω) angles of the helicopter over time can be computed exactly as a function of the roll rate $\dot{\phi}$, pitch rate $\dot{\theta}$ and yaw rate $\dot{\omega}$. Thus, given a model that predicts only the angular velocities, we can numerically integrate the velocities over time to obtain orientations.

We identified our model at 10Hz, so that the difference in time between s_t and s_{t+1} was 0.1 seconds. We used linear regression to learn to predict, given

[1] Actually, by handling the effects of gravity explicitly, it is possible to obtain an even better model that uses a further reduced, 6-dimensional, state, by eliminating the state variables ϕ and θ. We found this additional reduction useful and included it in the final version of our model; however, a full discussion is beyond the scope of this paper.

$s_t^b \in \mathbb{R}^8$ and $a_t \in \mathbb{R}^4$, a sub-vector of the state variables at the next timestep $[\dot{x}_{t+1}^b, \dot{y}_{t+1}^b, \dot{z}_{t+1}^b, \dot{\phi}_{t+1}^b, \dot{\theta}_{t+1}^b, \dot{\omega}_{t+1}^b]$. This body coordinate model is then converted back into a world coordinates model, for example by integrating angular velocities to obtain world coordinate angles. Note that because the process of integrating angular velocities expressed in body coordinates to obtain angles expressed in world coordinates is nonlinear, the final model resulting from this process is also necessarily nonlinear. After recovering the world coordinate orientations via integration, it is also straightforward to obtain the rest of the world coordinates state. (For example, the mapping from body coordinate velocity to world coordinate velocity is simply a rotation.)

Lastly, because helicopter dynamics are inherently stochastic, a deterministic model would be unlikely to fully capture a helicopter's range of possible behaviors. We modeled the errors in the one-step predictions of our model as Gaussian, and estimated the magnitude of the noise variance via maximum likelihood.

The result of this procedure is a stochastic, nonlinear model of our helicopter's dynamics. To verify the learned model, we also implemented a graphical simulator (see Figure 2b) with a joystick control interface similar to that on the real helicopter. This allows the pilot to fly the helicopter in simulation and verify the simulator's modeled dynamics. The same graphical simulator was subsequently also used for controller visualization and testing.

3.2 Controller design via reinforcement learning

Having built a model/simulator of the helicopter, we then applied reinforcement learning to learn a good controller.

Reinforcement learning [11] gives a set of tools for solving control problems posed in the Markov decision process (MDP) formalism. An MDP is a tuple $(S, s_0, A, \{P_{sa}\}, \gamma, R)$. In our problem, S is the set of states (expressed in world coordinates) comprising all possible helicopter positions, orientations, velocities and angular velocities; $s_0 \in S$ is the initial state; $A = [-1, 1]^4$ is the set of all possible control actions; $P_{sa}(\cdot)$ are the state transition probabilities for taking action a in state s; $\gamma \in [0, 1)$ is a discount factor; and $R : S \mapsto \mathbb{R}$ is a reward function. The dynamics of an MDP proceed as follows: The system is first initialized in state s_0. Based on the initial state, we get to choose some control action $a_0 \in A$. As a result of our choice, the system transitions randomly to some new state s_1 according to the state transition probabilities $P_{s_0 a_0}(\cdot)$. We then get to pick a new action a_1, as a result of which the system transitions to $s_2 \sim P_{s_1 a_1}$, and so on.

A function $\pi : S \mapsto A$ is called a policy (or controller). It we take action $\pi(s)$ whenever we are in state s, then we say that we are acting according to π. The reward function R indicates how well we are doing at any particular time, and the goal of the reinforcement learning algorithm is to find a policy

π so as to maximize

$$U(\pi) \doteq \mathrm{E}_{s_0, s_1, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi \right], \tag{1}$$

where the expectation is over the random sequence of states visited by acting according to π, starting from state s_0. Because $\gamma < 1$, rewards in the distant future are automatically given less weight in the sum above.

For the problem of autonomous hovering, we used a quadratic reward function

$$\begin{aligned} R(s^s) = -(&\alpha_x (x - x^*)^2 + \alpha_y (y - y^*)^2 + \alpha_z (z - z^*)^2 \\ &+ \alpha_{\dot{x}} \dot{x}^2 + \alpha_{\dot{y}} \dot{y}^2 + \alpha_{\dot{z}} \dot{z}^2 + \alpha_\omega (\omega - \omega^*)^2), \end{aligned} \tag{2}$$

where the position (x^*, y^*, z^*) and orientation ω^* specifies where we want the helicopter to hover. (The term $\omega - \omega^*$, which is a difference between two angles, is computed with appropriate wrapping around 2π.) The coefficients α_i were chosen to roughly scale each of the terms in (2) to the same order of magnitude (a standard heuristic in LQR control [1]). Note that our reward function did not penalize deviations from zero roll and pitch, because a helicopter hovering stably in place typically has to be tilted slightly.[2]

For the policy π, we chose as our representation a simplified version of the neural network used in [7]. Specifically, the longitudinal cyclic pitch $a[1]$ was commanded as a function of $x^b - x^{*b}$ (error in position in the x direction, expressed in body coordinates), $\dot{x}^b$, and pitch θ; the latitudinal cyclic pitch $a[2]$ was commanded as a function of $y^b - y^{*b}$, $\dot{y}^b$ and roll ϕ; the main rotor collective pitch $a[3]$ was commanded as a function of $z^b - z^{*b}$ and $\dot{z}^b$; and the tail rotor collective pitch $a[4]$ was commanded as a function of $\omega - \omega^*$.[3] Thus, the learning problem was to choose the gains for the controller so that we obtain a policy π with large $U(\pi)$.

Given a particular policy π, computing $U(\pi)$ exactly would require taking an expectation over a complex distribution over state sequences (Equation 1). For nonlinear, stochastic, MDPs, it is in general intractable to exactly compute this expectation. However, given a simulator for the MDP, we can approximate this expectation via Monte Carlo. Specifically, in our application, the learned model described in Section 3.1 can be used to sample $s_{t+1} \sim P_{s_t a_t}$

[2] For example, the tail rotor generates a sideways force that would tend to cause the helicopter to drift sideways if the helicopter were perfectly level. This sideways force is counteracted by having the helicopter tilted slightly in the opposite direction, so that the main rotor generates a slight sideways force in an opposite direction to that generated by the tail rotor, in addition to an upwards force.

[3] Actually, we found that a refinement of this representation worked slightly better. Specifically, rather than expressing the position and velocity errors in the body coordinate frame, we instead expressed them in a coordinate frame whose x and y axes lie in the horizontal plane/parallel to the ground, and whose x axis has the same yaw angle as the helicopter.

for any state action pair s_t, a_t. Thus, by sampling $s_1 \sim P_{s_0\pi(s_0)}$, $s_2 \sim P_{s_1\pi(s_1)}$, $\ldots$, we obtain a random state sequence $s_0, s_1, s_2, \ldots$ drawn from the distribution resulting from flying the helicopter (in simulation) using controller π. By summing up $\sum_{t=0}^{\infty} \gamma^t R(s_t)$, we obtain one "sample" with which to estimate $U(\pi)$.[4] More generally, we can repeat this entire process m times, and average to obtain an estimate $\hat{U}(\pi)$ of $U(\pi)$.

One can now try to search for π that optimizes $\hat{U}(\pi)$. Unfortunately, optimizing $\hat{U}(\pi)$ represents a difficult stochastic optimization problem. Each evaluation of $\hat{U}(\pi)$ is defined via a random Monte Carlo procedure, so multiple evaluations of $\hat{U}(\pi)$ for even the same π will in general give back slightly different, noisy, answers. This makes it difficult to find "$\arg\max_\pi \hat{U}(\pi)$" using standard search algorithms. But using the Pegasus method (Ng and Jordan, 2000), we can turn this stochastic optimization problem into an ordinary deterministic problem, so that any standard search algorithm can now be applied. Specifically, the computation of $\hat{U}(\pi)$ makes multiple calls to the helicopter dynamical simulator, which in turn makes multiple calls to a random number generator to generate the samples $s_{t+1} \sim P_{s_t a_t}$. If we fix in advance the sequence of random numbers used by the simulator, then there is no longer any randomness in the evaluation of $\hat{U}(\pi)$, and in particular finding $\max_\pi \hat{U}(\pi)$ involves only solving a standard, deterministic, optimization problem. (For more details, see [6], which also proves that the "sample complexity"—i.e., the number of Monte Carlo samples m we need to average over in order to obtain an accurate approximation—is at most polynomial in all quantities of interest.) To find a good controller, we therefore applied a greedy hillclimbing algorithm (coordinate ascent) to search for a policy π with large $\hat{U}(\pi)$.

We note that in earlier work, (Ng et al., 2004) also used a similar approach to learn to fly expert-league RC helicopter competition maneuvers, including a nose-in circle (where the helicopter is flown in a circle, but with the nose of the helicopter continuously pointed at the center of rotation) and other maneuvers.

4 Experimental Results

Using the reinforcement learning approach described in Section 3, we found that we were able to extremely quickly design new controllers for the helicopter. We first completed the inverted flight hardware and collected (human pilot) flight data on 3rd Dec 2003. Using reinforcement learning, we completed our controller design by 5th Dec. In our flight experiment on 6th Dec, we successfully demonstrated our controller on the hardware platform by having a human pilot first take off and flip the helicopter upside down, immediately

[4] In practice, we truncate the state sequence after a large but finite number of steps. Because of discounting, this introduces at most a small error into the approximation.

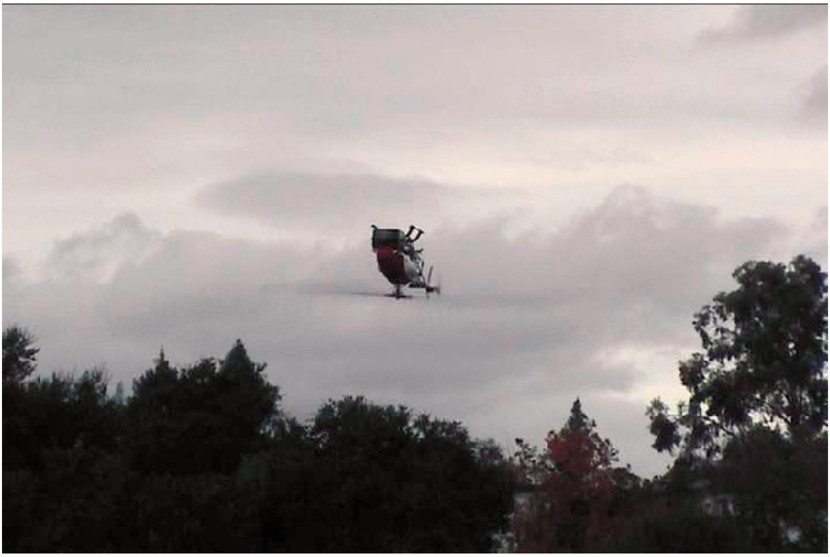


Fig. 3. Helicopter in autonomous sustained inverted hover.

after which our controller took over and was able to keep the helicopter in stable, sustained inverted flight. Once the helicopter hardware for inverted flight was completed, building on our pre-existing software (implemented for upright flight only), the total time to design and demonstrate a stable inverted flight controller was less than 72 hours, including the time needed to write new learning software.

A picture of the helicopter in sustained autonomous hover is shown in Figure 3. To our knowledge, this is the first helicopter capable of sustained inverted flight under computer control. A video of the helicopter in inverted autonomous flight is also at

`http://www.cs.stanford.edu/~ang/rl-videos/`

Other videos, such as of a learned controller flying the competition maneuvers mentioned earlier, are also available at the url above.

5 Conclusions

In this paper, we described a successful application of reinforcement learning to the problem of designing a controller for autonomous inverted flight on a helicopter. Although not the focus of this paper, we also note that, using controllers designed via reinforcement learning and shaping [5], our helicopter is also capable of normal (upright) flight, including hovering and waypoint following.

We also found that a side benefit of being able to automatically learn new controllers quickly and with very little human effort is that it becomes significantly easier to rapidly reconfigure the helicopter for different flight applications. For example, we frequently change the helicopter's configuration (such as replacing the tail rotor assembly with a new, improved one) or payload (such as mounting or removing sensor payloads, additional computers, etc.). These modifications significantly change the dynamics of the helicopter, by affecting its mass, center of gravity, and responses to the controls. But by using our existing learning software, it has proved generally quite easy to quickly design a new controller for the helicopter after each time it is reconfigured.

Acknowledgments

We give warm thanks to Sebastian Thrun for his assistance and advice on this project, to Jin Kim for helpful discussions, and to Perry Kavros for his help constructing the helicopter. This work was supported by DARPA under contract number N66001-01-C-6018.

References

1. B. D. O. Anderson and J. B. Moore. *Optimal Control: Linear Quadratic Methods.* Prentice-Hall, 1989.
2. J. Bagnell and J. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *Int'l Conf. Robotics and Automation.* IEEE, 2001.
3. J. Leishman. *Principles of Helicopter Aerodynamics.* Cambridge Univ. Press, 2000.
4. B. Mettler, M. Tischler, and T. Kanade. System identification of small-size unmanned helicopter dynamics. In *American Helicopter Society, 55th Forum*, 1999.
5. Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 278–287, Bled, Slovenia, July 1999. Morgan Kaufmann.
6. Andrew Y. Ng and Michael I. Jordan. Pegasus: A policy search method for large MDPs and POMDPs. In *Uncertainty in Artificial Intellicence, Proceedings of Sixteenth Conference*, pages 406–415, 2000.
7. Andrew Y. Ng, H. Jin Kim, Michael Jordan, and Shankar Sastry. Autonomous helicopter flight via reinforcement learning. In *Neural Information Processing Systems 16*, 2004.
8. Jonathan M. Roberts, Peter I. Corke, and Gregg Buskey. Low-cost flight control system for a small autonomous helicopter. In *IEEE International Conference on Robotics and Automation*, 2003.
9. T. Schouwenaars, B. Mettler, E. Feron, and J. How. Hybrid architecture for full-envelope autonomous rotorcraft guidance. In *American Helicopter Society 59th Annual Forum*, 2003.

10. J. Seddon. *Basic Helicopter Aerodynamics*. AIAA Education Series. America Institute of Aeronautics and Astronautics, 1990.
11. Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

X. Multiple Robots

There are definite advantages in using multiple robots to collective accomplish a task. The area of distributed autonomous robotic systems represents the merging of traditional robotics research and multi-agent systems research typical in artificial intelligence and computer science disciplines.

The first paper by Correll and Martinoli reports work in the use of a *swarm* of robots to perform an inspection on a regular engineered structure. A collective swarm behavior in performing the task emerges out of a simple robot-to-robot interaction algorithm, making it simple and robust yet effective in completing the task. The study also reported the effect of the environmental structure on the effectiveness of the algorithms.

The next article by Yoder and Seelinger shows a cooperative effort between two mobile manipulators in accomplishing a common task. A large and not-so-accurate robot was tasked to carry a smaller and accurate robot to complete a "cherry-picking" task, using uncalibrated vision feedback and minimum communication between the two robots. The overview of the coordinating algorithm and the result of the experiment were presented.

In the third paper, Brooks and co-workers report a study of Active Sensor Network, where information is gathered from a range of heterogeneous sensor platforms. Twelve different stationary sensors were set to monitor the human (and robot) traffic in a room for several days, coming on and off line individually. The paper concentrates mainly on the implementation aspects of an Active Sensor Network.

The fourth paper by Bourgault and colleagues present a similar case study to those in the third and fourth articles of Chapter IV (Localization and Map-Building) whereby a number of heterogeneous mobile robots with a range of different sensors perform a search algorithm for a target in an indoor environment. The paper raised an interesting possibility, as a part of ongoing research, of human intervention in the sensor network to facilitate the search process.

In the last paper, Jones and Mataric present their work on the control of multi-robot systems. The robots are homogeneous, with distributed intelligence and limited amount of non-transient internal state. There is no inter-robot communications. An unexpected collective behavior can emerge out of simple behavior of the individual robots. This paper presents a macroscopic model for calculating the probability that these homogeneous robots would correctly execute the given task. The model aids the decision process related to the appropriate use of internal state and its trade-offs with resulting system task performance.

Collective Inspection of Regular Structures using a Swarm of Miniature Robots

N. Correll and A. Martinoli

Swarm-Intelligent Systems Group, Nonlinear Systems Laboratory, EPFL, 1015 Lausanne
nikolaus.correll|alcherio.martinoli@epfl.ch

Abstract. We present a series of experiments concerned with the inspection of regular, engineered structures carried out using swarms of five to twenty autonomous, miniature robots, solely endowed with on-board, local sensors. Individual robot controllers are behavior-based and the swarm coordination relies on a fully distributed control algorithm. The resulting collective behavior emerges from a combination of simple robot-to-robot interactions and the underlying environmental template. To estimate intrinsic advantages and limitations of the proposed control solution, we capture its characteristics at higher abstraction levels using non-spatial, microscopic and macroscopic probabilistic models. Although both types of models achieve only qualitatively correct predictions, they help us to shed light on the influence of the environmental template and control design choices on the considered non-spatial swarm metrics (inspection time and redundancy). Modeling results suggest that additional geometric details of the environmental structure should be taken into account for improving prediction accuracy and that the proposed control solution can be further optimized without changing its underlying architecture.

1 Introduction

Swarm Intelligence (SI) is an emerging computational and behavioral metaphor for solving distributed problems. SI takes its inspiration from biological examples provided by social insects [4] and by swarming, flocking, herding, and shoaling phenomena in vertebrates [12]. The abilities of such natural systems appear to transcend the abilities of the constituent individual agents, while being mediated by nothing more than a small set of simple local interactions.

Biological systems show a full blend of antagonist mechanisms for coordinating their collective behavior. For instance, an environmental template (e.g., a temperature gradient or a pheromone gradient generated by the termite queen), a sort of centralized source of information, in combination with distributed building activities of insects allow for the construction of extremely sophisticated structures [4]. As a consequence, the resulting overall behavior of the colony is self-organized and extremely robust to noise since it combines *exploitation* (i.e., following the environmental blueprint and pre-programmed individual behavioral rules) with *exploration* (i.e., the resulting individual behavior is heavily influenced by noisy, local perception).

In this paper, we propose a distributed control algorithm that exploits similar SI-based principles in order to achieve robust inspection and/or coverage of a regular

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 375–385, 2006.

structure and overcomes intrinsic limitations of individual robots in terms of processing power and sensorial capabilities. The environmental template is in this case represented by the engineered structure to be inspected, while robots emulate biological entities interacting among themselves and with the structure for carrying out the inspection task. In scenarios where, for instance, robots cannot be endowed with sophisticated positioning systems or navigation algorithms, the SI-based approach represents a valid alternative to deliberative coverage strategies which usually achieve deterministic completeness only in absence of uncertainty in robot position (see for instance [1,7]). Finally, although a SI-based approach leads to a certain inspection redundancy because of its intrinsically probabilistic swarm coordination, depending on the reliability of the sensors used for detecting the desired features targeted in the inspection, this property might be beneficial if an error-free feature detection cannot be guaranteed [1,8].

In order to evaluate the performance of the swarm as inspection system, we have defined two different swarm metrics: time to complete inspection and inspection redundancy. Although in a swarm robotic system, microscopic interactions are often characterized by an important stochastic component and are mathematically intractable, the performance of a swarm as a whole is statistically predictable. Therefore, following the efforts recently performed on distributed manipulation experiments [2,9,11], we propose microscopic and macroscopic models to understand and evaluate the influence of key system parameters on the desired metrics.

2 The Inspection Case Study

As a first challenging case study, we consider the inspection of jet turbine engines. In order to minimize failures, jet turbine engines have to be inspected at regular intervals. This is usually performed visually using borescopes, a process which is time consuming and cost intensive [10]. One possible solution for speeding up and automating the inspection process is to rely on a swarm of autonomous, miniature robots which could be released into the turbine without disassembling it. While this idea is intellectually appealing and could pave the way for other similar applications in coverage/inspection of engineered or natural, regular structures, it involves a series of technical challenges which dramatically limit possible designs of robotic sensors and hence emphasizes a SI-driven approach. For instance, the shielded, complex, narrow structure of a turbine imposes not only strong miniaturization constraints on the design but also prevents the use of any traditional global positioning and communication system. Furthermore, a limited on-board energy budget might prevent computation of a sophisticated deliberative planning strategy and dramatically narrows the sensor and communication range of our robots.

2.1 The Physical Setup emulating a Turbine Scenario

In order to validate our distributed control strategy using the real miniature robots *Alice II* [5], we simplify the turbine inspection scenario by unfolding the axis-symmetric geometry of the turbine into a flat representation with the blades as vertical extrusions. This simplified turbine environment was implemented in a rectangular arena of $1.10x1m^2$ monitored by an external overhead camera. The Alice miniature robot is endowed with a PIC microcontroller (368 bytes RAM, 8Kb FLASH), has a length of 22mm, and a maximal speed of $4\frac{cm}{s}$. It is endowed with 4 IR modules which can serve as very crude proximity sensors (up to 3cm) and local communication devices (up to 6 cm in range). Its energetic autonomy ranges between 5h and 10h, depending on the actuators and sensors used.

This platform was chosen not only because it allows us to perform and monitor experiments with large number of robots in the lab, but also because Alice robots force us to design control architectures that are simple and robust enough to deal with their limited capabilities.

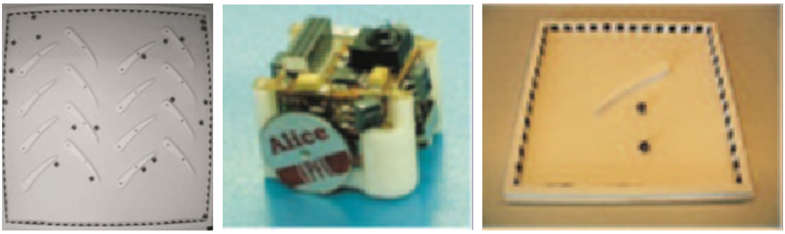


Fig. 1. *Left*: Overview of the robot arena emulating a simplified turbine scenario. *Middle:* The Alice II Micro Robot. *Right:* Simplified setup for the calibration of model parameters.

2.2 The Behavior-Based Robot Controller

In this paper, we are not concerned with detection of flaws - which could be achieved, for instance, by a miniature on-board camera - but rather with the individual and group motion in the turbine scenario. For the sake of simplicity, we therefore assume that circumnavigating a blade in its totality is a good emulation of a scanning-for-flaws maneuver.

The behavior of a single robot is determined by a schema-based controller [3] that tightly links the platforms' actions to sensor perception while using as little representational knowledge of the world as possible. For a schema-based controller, behavioral responses are represented by vectors generated by local potential fields and behavioral coordination is achieved by vector addition. Sequencing of behaviors is achieved by a dynamic action-selection mechanism based on internal timers which are in turn set and reset according to schema response. The overall behavior of a robot can be summarized as follows (see Figure 2, left). The robot searches for blades throughout the turbine, combining schemes that drive the robot forward, avoid obstacles, and follow contours. Using exclusively its on-board sensors, the robot can discriminate between the three different types of objects within the arena: teammates, blades, and external walls. Teammates are distinguishable from any other objects by their active, modulated emission. Blades are distinguishable from

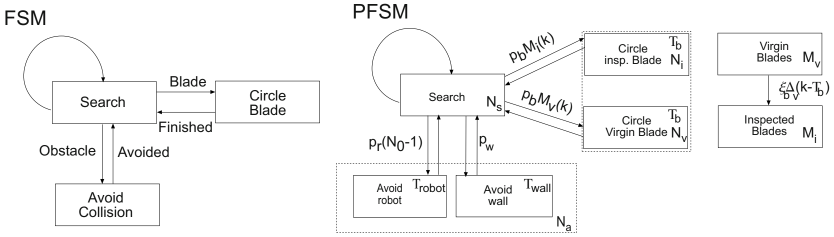


Fig. 2. *Left*: The high-level behavioral flowchart of the robot controller as a Finite State Machine (FSM). *Right*: The corresponding Probabilistic FSM used in the models, capturing details of interest of the schema-based controller.

external walls because the former have uniform white contours while the latter are characterized by a pattern of black and white stripes.

Teammates and external walls are systematically avoided while blades stimulate a contour-following behavior in the robot. A robot can start circumnavigating a blade at any point of its contour but can leave the blade exclusively at its tip. This allows the robot to bias its blade-to-blade trajectory without using any sophisticated navigation mechanisms, exploiting a specific feature of the environmental pattern. The robot can recognizes a blade's tip again via a specific sensorial pattern generated by its on-board proximity sensors.

A further design parameter is introduced in the controller in order to achieve a full inspection of a blade. The blade can be left only if the timeout parameter T_{max} has expired. The corresponding timer is set when the robot attaches to the blade. If the circumnavigating behavior of the robot were perfect, T_{max} would perfectly control the number of blade tours performed by a robot before moving to the next blade. However, rounding the blade's tip without losing contact with the blade represents a challenging maneuver for a miniature robots such as the Alice, in particular because of sensor noise, heterogenous lighting conditions, and so on. We have therefore systematically assessed (see Section 3.2) the average probability for a robot to lose contact with a blade while rounding its tip (p_l), the mean time needed to partially circumnavigate a blade (random attaching point) and systematically leave the blade at its tip (T_{hb}), and the average time to completely circumnavigate a blade ($T_{fb} = 2T_{hb}$). In this paper, we have chosen to set T_{max} so that $T_{fb} < T_{max} < T_{hb}+T_{fb}$. In this case, the robot either leaves the blade at the first encounter with the blade's tip with probability p_l or continues the circumnavigation with probability $1 - p_l$ and leaves at the second encounter with the blade's tip.

3 Microscopic and Macroscopic Models

The central idea of the probabilistic modeling methodology is to describe the experiment as a series of stochastic events with probabilities computed from the interactions' geometrical properties and systematic experiments with one or two real robots [11] or embodied agents [2]. Consistent with previous publications, we can use the controller's FSM depicted in Figure 2 as a blueprint to devise the Probabilistic FSM (PFSM or Markov chain) representing an individual agent at the microscopic level or the whole swarm at the macroscopic level. At the microscopic level, a specific state represents the current mode of a certain individual, while a state at the macroscopic level defines the average number of individuals in that same mode. The state granularity can be arbitrarily chosen in order to capture details of the robot's controller and environment which influence the swarm performance metrics (in our case, the time to completion and the number of redundantly inspected blades). Although at first glance the microscopic model might appear unnecessary since the models presented are essentially linear (and therefore no major discrepancies between mean microscopic and macroscopic predictions can arise), numerical difficulties in assessing the end criterion at the macroscopic level for one of the chosen metrics (time to inspection completion) justify this choice. This is a well-known problem in such types of models when they are used to predict metrics based on discrete quantities (i.e., in our case a finite number of blades). The overall PFSM for the system is represented graphically in Figure 2, *right* using two coupled PFSMs, one representing the robot(s) and one representing the shared turbine environment.

3.1 Modeling Assumptions

As more extensively detailed in [2,11], the modeling methodology relies on three main assumptions. First, coverage of the arena by the group of robots is uniform and robots' trajectories and objects' positions in the arena do not play a role in the non-spatial metrics of interest. Second, a robot's future state depends only on its present state and how much time it has spent in that state (semi-Markov property). Third, agents change their state autonomously but synchronously to a common clock whose time step has been chosen to capture with sufficient precision all time delays considered in the system as well as changes in the metrics of interest.

3.2 Characterization of Models' Parameters

All our models are characterized by two categories of parameters: state-to-state transition probabilities and behavioral delays. In contrast with previous publications, we do not assume any coupling between these two categories of parameters, but we compute and calibrate them based on either systematic measurements (e.g., p_l) or the concept of *encountering rates* [6], keeping a clear separation between *geometric detection probabilities* and *encountering probabilities*. Here, geometric detection probability is the probability that a robot is within the detection area of a certain object. The detection area of an object is determined by its physical size, the sensory configuration and corresponding processing used by the robot to reliably detect it. After defining the contours of the detection area A_i for a given object i, we calculate its geometric detection probability g_i by dividing A_i by the whole arena area A_a. We can hence calculate the corresponding encountering probability, i.e. the probability of encountering the object i per time step, using the corresponding encountering rate r_i (in s^{-1}). The conversion factor from geometric detection probabilities to encountering rates is given by the average robot speed v_r ($4\frac{cm}{s}$), its detection width

w_r, and the detection area A_s of the smallest object in the arena (a robot is a possible type of object). The detection width is defined as twice the maximum detection distance of the smallest robot/object in the arena, measured from center of the robot to the center of the object, given here by $2R_s = 4.2cm$ with $A_s = R_s^2\pi$. Equation 1 shows how to compute the encountering probability for the object i given the geometric detection probability g_i:

$$p_i = r_i T = \frac{v_r w_r}{A_s} g_i T \tag{1}$$

with T being the time step characterizing our time-discrete models. In this paper, we discretize the different average durations of interactions so that changes in our two arbitrary metrics are described with sufficient precision using $T = 1s$. Mean numerical values used for the model parameters are summarized in Table 1. Mean detection range R_s, p_l, and T_{hb} were measured using real robots in a simplified setup (compare Figure 1, right). Encountering probabilities based on the geometric probability of each object were computed by Equation 1.

Table 1. Encountering probabilities for the three objects a robot might encounter, their detection area, and their mean interaction time. Each interaction experiment was repeated 40 times.

Object	Computed enc. rate	Detect. area	Mean interaction time
Wall	$p_w = 0.0458$	$420cm^2$	$T_w = 10s$
Blade	$p_b = 0.0106$	$96.48cm^2$	$T_{hb} = 10s, p_l = 0.4$
Robot	$p_r = 0.0015$	$14cm^2$	$T_r = 4s$

The mean blade interaction time T_b is a function of p_l, T_{hb}, and T_{max}. With the T_{max} used in this paper, it can be calculated as follows: $T_b = p_l T_{hb} + (1 - p_l)(3T_{hb}) = 22s$.

3.3 Mathematical Description of the Macroscopic Model

From Figure 2, *right* we can derive a set of difference Equations (DE) to capture the dynamics of the whole system at the macroscopic level. We formulate one DE per considered state and exploit equations stating the conservation of the number of robots and the number of blades to replace one of the DEs.

Given M_0 blades and N_0 robots, the number of robots covering virgin and inspected blades N_v and N_i, the number of robots in obstacle avoidance N_a, and the number of robots in search mode N_s are given by Equation 2-5 (compare also Figure 2); the number of virgin blades M_v and the number of inspected blades M_i are calculated by Equation 6-7:

$$\begin{aligned} N_s(k+1) &= N_s(k) - \Delta_v(k) - \Delta_i(k) - \Delta_r(k) - \Delta_w(k) + \Delta_v(k - T_b) \\ &\quad + \Delta_i(k - T_b) + \Delta_r(k - T_r) + \Delta_w(k - T_w) \end{aligned} \tag{2}$$

$$N_a(k+1) = N_a(k) + \Delta_r(k) + \Delta_w(k) - \Delta_r(k - T_r) - \Delta_w(k - T_w) \tag{3}$$

$$N_v(k+1) = N_v(k) + \varDelta_v(k) - \varDelta_v(k - T_b) \tag{4}$$
$$N_i(k+1) = N_0 - N_s(k+1) - N_a(k+1) - N_v(k+1) \tag{5}$$

$$M_v(k+1) = M_v(k) - \xi_b \varDelta_v(k - T_b) \tag{6}$$
$$M_i(k+1) = M_0 - M_v(k+1) \tag{7}$$

with k representing the current time step (and absolute time kT); $k = 0 \ldots n$, n being the total number of iterations (and therefore nT the end time of the experiment). The $\varDelta$-functions define the coupling between state variables of the model and can be calculated as follows:

$$\varDelta_v(k) = p_b(M_v(k) - N_v(k))N_s(k) \tag{8}$$
$$\varDelta_i(k) = p_b(M_i(k) + N_v(k))N_s(k) \tag{9}$$
$$\varDelta_r(k) = p_r(N_0 - 1)N_s(k) \tag{10}$$
$$\varDelta_w(k) = p_w N_s(k) \tag{11}$$

Here, p_b, p_r, and p_w represent the encountering probabilities of blades, robots, and wall. T_b, T_r, T_w define the average time needed for circumnavigating a blade, avoiding a teammate, and avoiding a wall respectively. Due to the probability of leaving a blade before it has been completely covered, we introduce a parameter ξ_b, being the percentage of a blade a robot covers on average at each new interaction with it. Similarly to T_b, ξ_b can be calculated from p_l, T_{hb}, and T_{fb}: $\xi_b = p_l T_{hb}/T_{fb} + (1 - p_l)T_{fb}/T_{fb} = 0.8$.

The initial conditions are $N_s(0) = N_0$ and $N_a(0) = N_v(0) = N_i(0) = 0$ for the robotic system (all robots in search mode) while those of the environmental system are $M_v(0) = M_0$ and $M_i(0) = 0$ (all blades virgin). As usual for time-delayed DE, we assume $\varDelta_x(k) = N_x(k) = M_x(k) = 0$ for $k < 0$.

For instance, we can interpret the first DE (Equation 2) as follows. The average number of robots in the searching state is decreased by those that start to cover a virgin blade or an inspected blade and those that start avoiding either a teammate or a wall; it is increased by all robots resuming searching after either an inspection or an obstacle avoidance maneuver, each of them being characterized by a specific duration. The other state equations can be interpreted in a similar way.

3.4 Swarm Performance Metrics

We consider two metrics for evaluating the performance of the swarm: time to completion nT and coverage redundancy. The task is completed if all blades are inspected ($M_v(n) = 0$), while the coverage redundancy R is given by $R = M_r(n) - M_0$ with

$$M_r(n) = \sum_{k=0}^{n} \rho_v \varDelta_v(k - T_b) + \rho_i \varDelta_i(k - T_b) \tag{12}$$

the total number of inspected blades. Here, the ρ-terms reflect the mean redundant coverage on encountering a virgin or an inspected blade respectively. Hence, ρ_i and

ρ_v can be computed as weighted sums in terms of coverage (partial circumnavigation with probability p_l and over-complete circumnavigation with probability $1 - p_l$): $\rho_i = p_l T_{hb}/T_{fb} + (1 - p_l)(T_{hb} + T_{fb})/T_{fb} = 1.1$ and $\rho_v = (1 - p_l)T_{hb}/T_{fb} = 0.3$.

To compute the time to completion nT, $M_v(n) = 0$ is an easy condition to apply in the experiment and in the microscopic model. However, in the macroscopic model, this represents a limit condition as $\lim_{k\to\infty} M_v(k) = 0$. Therefore, we solved the DEs numerically for $M_r(n) = \mu$, with μ the mean resulting from the microscopic model.

4 Results and Discussion

In this section we will present the results collected using real robots and compare them to model predictions with regard to the desired swarm metrics. In contrast to previous experiments in the distributed manipulation class where obstacles were either moving [2] or much smaller [11] than the immovable blades considered in this case study, the swarm performance metrics in the inspection task appear to be more closely influenced by the distribution of the robots, while still being non-spatial. Furthermore, objects in the arena are not all axis-symmetric as in the previous experiments. Finally, the structure created by the blades themselves also plays a relevant role in the prediction accuracy of the models as it biases not only the uniformity of the robot's spatial distribution at a given moment but also its evolution over time. All these features are not taken into account in our current modeling methodology and therefore generate discrepancies between models and experimental results. We will discuss them in more detail in Section 4.2.

4.1 Swarm Performance Metrics

We estimated time to completion to cover 16 blades and the total number of covered blades by doing 10'000 runs of the microscopic model for team sizes of 1 to 20 robots and using the parameters summarized in Table 1. To validate model predictions, we ran 20 experiments each for team sizes of 5, 10, 16, and 20 robots. In order to come closer to our assumption of spatial uniformity (see Section 3.1), robots were initially distributed randomly in the arena. Figure 3 shows experimental and modeling results for both swarm performance metrics (time to complete the inspection, number of redundantly inspected blades). We observe that model predictions for the overall task dynamics in both metrics are only qualitatively correct for small teams. For larger team sizes however, quantitative prediction improves. We believe that this is because an individual robot's trajectory does not satisfy our assumption of spatial uniformity. Increasing the team size instead weakens the effect of individual trajectories and increases the quality of prediction. It is worth noting that, in contrast to experiments reported in [6] where robots never left a blade prematurely during a circumnavigation maneuver, in the experiments reported here the inspection redundancy before task completion increases with the number of robots (see Figure 3, *right*).

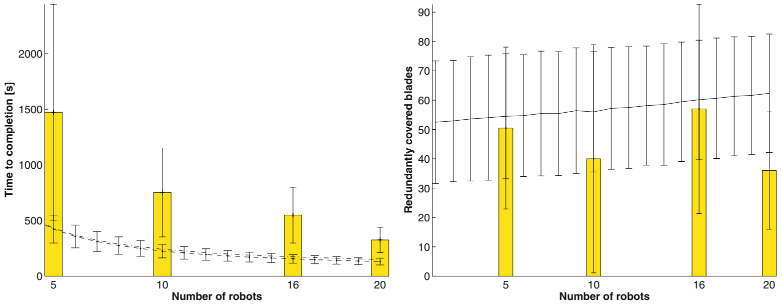


Fig. 3. Comparison of microscopic and macroscopic model predictions (basically superimposed) and experimental results. Experimental results are represented by the median and standard deviation because of long tail distributions while microscopic models by mean and standard deviation. *Left*: Time to completion (16 blades) vs. team size (1 to 20 robots, 5, 10, 16, 20 robots respectively). *Right*: Number of blades redundantly covered until inspection completion vs. team size.

4.2 Influence of Boundary Conditions

As described in Section 2.2, the robots exploit the regularity of the blade pattern to effectively traverse the turbine moving to the next blade only when they reach the tip of the current blade. This behavior in combination with the geometrical structure of the pattern biases the distribution of the robots in space and over time.

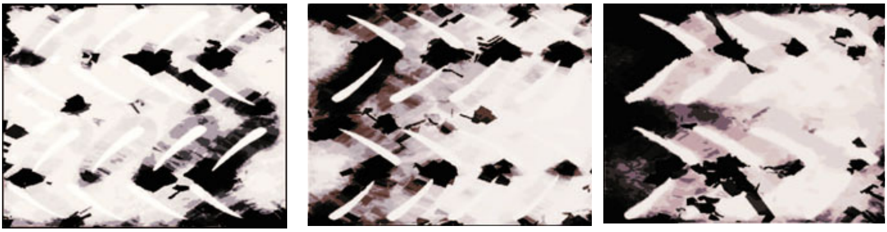


Fig. 4. Distribution of the robots on the arena (dark areas correspond to frequently visited regions). Each experiment lasted 2h with 20 robots. *Left*: Default setup with the default inspection algorithm. *Center*: Set-up rotated of 180 degrees. *Right*: Rotated setup and robots characterized by a random walk behavior.

In order to investigate the influence of the blade configuration, we performed additional experiments with different configurations, the original setup turned by 180 degrees in order to cancel out external environmental influences (light, arena slope etc.), and the rotated setup but with randomly moving robots, avoiding obstacles. The resulting cumulative distributions for a $2h$ period and 20 robots are depicted in Figure 4.

In addition to local high density zones where traffic jams occur, we note an overall tendency: although the robots are scattered randomly in the arena at the beginning of the experiments, fairly independently from their behavior, over time they tend to move in direction of the blades's tips and thus visit some areas of the arena more often than others. In Figure 4 *left*, we observe that frequency of coverage by the robots increases from the left to the right and from the right to the left in Figure 4, *middle* and *right*. In the realistic simulation experiments reported in [6], this effect did not appear as it was compensated by a wrap-around zone at the arena borders (mimicking the cylindrical turbine environment). This is no longer the case in the physical setup presented here since the turbine structure is delimited by walls. As a consequence, in a fully enclosed arena, the probability to encounter a blade is dependent upon its level along the main axis of the structure and the elapsed experimental time ($p_b \rightarrow p_b(x, y, t)$), thus violating our assumptions of spatial uniformity (see Section 3.1).

5 Conclusion and Future Work

In this paper, we proposed a simple distributed control algorithm to inspect a regular structure with a swarm of miniature robots. We investigated the overall collective behavior of the swarm by generating the corresponding non-spatial models. The results show that the time to inspection completion decreases sublinearly with the number of robots, that the inspection redundancy until task completion increases with the swarm size if robots prematurely leave the blade they are inspecting, and that the swarm's center of mass can actually be moved along the regular structure by exploiting local interactions between the robots and the structure itself.

We have also shown that our modeling methodology provides useful insight in this experiment as has been the case in previous distributed manipulation experiments. In particular, we believe that the control design parameter T_{max} might be further optimized to obtain more interesting T_b and ξ_b pairs for a constant p_l. Indeed, this latter parameter might be more difficult to control since it defines the interaction of a miniature robot with limited capabilities with the environmental structure. Finally, we also demonstrated that the current modeling methodology reaches its limitations in case studies where the structure of the environment plays a major role on the robot distribution. In order to be able to exploit models as tools for performance prediction and design, we need to incorporate more geometric details of the environmental template. We believe that this will enable the design of more customized algorithms as a function of the regular engineering structure to be inspected.

Acknowledgements

This work has been initiated at the California Institute of Technology, primarily supported by the NASA Glenn Center and in part by the Caltech Center for Neuromorphic Systems Engineering under the US NSF Cooperative Agreement ERC-9402726. Both authors are currently sponsored by a Swiss NSF grant (contract Nr.

PP002-68647/1). We would like to thank M. Asadpour and G. Caprari for their crucial support while working with the Alice robots.

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*, 22(7-8), 2003.
2. W. Agassounon, A. Martinoli, and K. Easton. Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes. *Autonomous Robots*, 17(2-3):163–192, 2004. Special issue on Swarm Robotics, M. Dorigo and E. Sahin, editors.
3. R. Arkin. *Behavior-Based Robotics*. The MIT press, Cambridge, MA, USA, 2000.
4. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. SFI Studies in the Science of Complexity, Oxford University Press, New York, NY, USA, 1999.
5. G. Caprari and R. Siegwart. Design and control of the mobile micro robot alice. In *Proceedings of the 2nd International Symposium on Autonomous Minirobots for Research and Edutainment*. Brisbane, Australia, pages 23-32, 2003.
6. N. Correll and A. Martinoli. Modeling and optimization of a swarm-intelligent inspection system. In *Proceedings of the 7th Symposium on Distributed Autonomous Robotic System (DARS)*. Toulouse. Distributed Autonomous Systems 6. To Appear, 2004.
7. Y. Gabriely and E. Rimon. Spiral-stc: An on-line coverage algorithm of grid environments by a mobile robot. In *Proceedings of the 2002 IEEE Conference on Robotics & Automation*. Washington, DC, USA, pages 954-960, 2001.
8. Douglas Gage. Many-robot mcm search systems. In A. Bottoms, J. Eagle, and H. Bayless, editors, *Proc. of the Autonomous Vehicles in Mine Contermeasure Symposium*. Monterey, CA, USA, pages 9.55-9.63, 1995.
9. K. Lerman and A. Galstyan. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 2(13):127–141, 2002.
10. K. Martin and C.V. Stewart. Real time tracking of borescope tip pose. *Image and Vision Computing*, 10(18):795–804, July 2000.
11. A. Martinoli, K. Easton, and W. Agassounon. Modeling of swarm robotic systems: A case study in collaborative distributed manipulation. *Int. Journal of Robotics Research*, 23(4):415–436, 2004. Special issue on Experimental Robotics, P. Dario and B. Siciliano, editors. Invited paper.
12. J. K. Parrish and W. M. Hammer. *Animal Groups in Three Dimensions*. Cambridge University Press, UK, 1997.

Visual Coordination of Heterogeneous Mobile Manipulators

John-David Yoder[1] and Michael Seelinger[2]

[1] Ohio Northern University
525 South Main, Ada, OH 45810, USA
j-yoder@onu.edu
[2]Yoder Software, Inc. (YSI)
715 W. Michigan, Urbana, IL 61801, USA
mseelinger@yodersoftware.com

Abstract. This paper presents work done to control robots of different geometries and capabilities to complete a task neither is capable of independently. The algorithms were tested with a ‘cherry picker’ experiment that required one large, inaccurate robot to lift and carry a smaller, more accurate mobile robot in order to complete an inspection task by placing its end-effector at a specified distance from a visually-specified target. Both mobile robots are controlled using an uncalibrated visual guidance method. An overview of the algorithm for coordinating the two mobile robots is presented, along with details and results of experiments conducted to measure the accuracy with which the shared task is completed. Results have shown the system to be accurate and robust while requiring very little communication between the two robots.

1 Introduction

Many researchers have made the case that a great number of the practical applications of mobile robots will require multiple, cooperating robots. In this paper we present the experimental development of a novel approach for controlling two cooperating, heterogeneous mobile manipulator robots. “Heterogeneous” indicates that the multiple robots will have differing capabilities and/or geometries. “Mobile manipulator” is used to describe systems that have the ability to move within their environment (like all mobile robots) and have at least one degree of freedom (DOF) to allow for manipulation. Examples of mobile manipulators include the Spirit and Opportunity NASA Mars rovers, typical industrial forklifts, or a mobile robot equipped with a standard robotic arm. This paper presents research and experiments which allows coordination of the differing capabilities of the two very dissimilar robots to accomplish a task that neither would be capable of doing alone. Both robots are controlled using the mobile camera-space manipulation (MCSM) visual guidance method, which enables them to manipulate objects precisely in their workspace in either a fully autonomous mode (if the task is sufficiently well-defined) or with supervisory level input [1].

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 387–396, 2006.

Coordination of multiple mobile robots can be useful for many purposes. Such purposes could include mapping and localization tasks [2,3] or transportation tasks [4,5], among others. Such tasks typically do not require the ability to position an end-effector precisely relative to another object. Tasks such as inspection, maintenance, construction, and assembly, however, do require this positioning capability. The approach developed here aims to address this set of tasks.

2 Related work

Much of the research done in the area of coordinating and controlling multiple mobile robots has focused on the architecture for sharing information among robotic agents as well as the cooperation strategy between these agents. A survey of this research can be found in [6,7]. Examples of such cooperation include the work done by Rus *et al.* for the distributed coordination of a team of robots in changing the orientation of a piece of furniture [8]. Trebi-Ollennu, *et al.* developed a strategy for transporting a long beam using two mobile manipulator robots [4]. Ahmadabadi *et al.* created a distributed cooperation strategy to control a group of cooperative behavior-based mobile robots for lifting and transporting an object [9]. Fiero *et al.* developed a software architecture for the control of multiple mobile robots deployed in unstructured environments [10]. Khatib *et al.* developed a decentralized control structure specifically for cooperating mobile manipulators [11]. These examples all were implemented with multiple robots of similar design and capability (homogeneous robots).

Some problems in particular lend themselves to solution by multiple heterogeneous robots, some or all of which may be mobile. For instance, construction could be done by multiple, powerful and accurate mobile robots. However, it is more cost effective to be able to have a few, large powerful robots and a few highly accurate, small robots. The analogy in traditional construction would be using a few cranes for the 'heavy lifting' while human workers carry out the final positioning and fastening. See [12] for example. Some exploration tasks could also benefit from the cooperation of multiple heterogeneous robots. Different mobile robot configurations are better suited for different types of terrain. NASA's Spirit and Opportunity rovers, for example, have been built for exploration, but are limited in the size of obstacles they can climb. A 'helper' robot that could lift or hoist a small rover over large obstacles could substantially expand the workspace of exploring robots. Another type of task that would benefit from the cooperation of heterogeneous robots is a maintenance task. The human analogy is a cherry-picker, where a machine is used to lift a human worker into the general region where the maintenance work is needed and the human performs the fine manipulation required. In a similar fashion, a small, accurate maintenance robot could require assistance from a larger mobile robot with lifting capabilities.

This class of problems has led researchers to examine algorithms for sharing of information and control among multiple, heterogeneous robots (see, for instance, [12-14]). Our approach focuses on tasks requiring high precision manipulation, and allows for cooperation among robots without the need for extensive communication. This reduces networking overhead and bandwidth requirements, while improving task completion accuracy.

3 Coordination of heterogeneous mobile manipulators

The approach described in this paper is specifically aimed at tasks requiring high-precision manipulation. Clearly, in order for multiple robots to cooperate for this type of task, the capability to control them for precise end-effector position must be present. The authors have previously developed MCSM, an algorithm for precision vision-based control of individual mobile manipulators [1]. MCSM allows for task completion with a minimal number of DOF and without *a priori* calibration. Tasks are specified visually – either autonomously or remotely by a human supervisor. The remainder of this paper describes the extension of this approach to multiple, heterogeneous robots. The paper will discuss development and testing of the algorithms used for target acquisition, specification, tracking, and task completion.

3.1 System Hardware

The forklift is a Crown SC series three-wheeled electric model with load capacity of 1,400 kg and vertical reach of 7.3 m. A 450 MHz Celeron PC computer running Windows NT 4.0 controls the forklift system. It uses an Acroloop 8000 8-axis motion control card to control all of the actuators, to read all of the analog feedback devices, and to send/receive data through several i/o channels. Two Imagenation PX-610 frame grabbers each having resolution of 640x480 pixels acquire images from the two Sony XC-75 cameras. Five IDC (Industrial Devices Corporation) linear actuators are used to control various functions of the forklift. Three of the five move the forklift's hydraulic valves that control motion of the forks: up/down, left/right, and tilt angle. One linear actuator controls the accelerator and the other the brake. A Kollmorgen motor is used to control the angle of the steering wheel. The system uses three analog feedback devices. Two Penny + Giles Controls Inc. linear potentiometers give feedback of the left/right position as well as the tilt angle of the forks. The third analog feedback device is a UniMeasure, Inc. linear position transducer, which gives feedback of the forks' vertical position. Two optical encoders are mounted via rolling contact on each of the two drive wheels providing feedback on forklift wheel motion.

The rover is constructed of aluminum with an overall weight of approximately 45 kg. Both the robotic arm and the independently driven front

Figure 1: Cherry Picker in Use

wheels are powered by Quality Motion Control 2234 motors equipped with Applied Motion Products 55:1 gearboxes. A 4-Axis Acroloop 2000 motion control card along with DC power supplies and suitable amplifiers provide PID computer control of these three motors. Two Sony XC-75c cameras mounted approximately 65 cm apart on either side of the front of the rover have 5 mm wide-angle lenses. Two Imagenation brand CX-100 frame grabbers with resolution of 512 x 486 pixels are used to acquire video images from the cameras. A Lasiris laser pointer mounted on a Directed Perception two-axis pan/tilt unit (PTU) is attached to the front end of the rover. The laser is turned on and off by an output channel on the motion controller and the PTU is controlled via the computer through a COM port. The main processor for the system is a Pentium II 400 MHz computer running Windows NT 4.0. Power is provided to the system by a standard extension cord. The typical operating speed during automatic maneuvers is about 2 cm/sec. The rover can traverse a wide variety of terrain including loose gravel and sand.

The handshaking between the forklift and the rover is conducted through digital I/O channels on the motion control cards of each system.

3.2 Coordination Task

The two systems described above each have clear similarities and differences. They are both mobile manipulators, but they have vastly different workspaces, weights, power requirements, and accuracies (rover positioning less than 1 mm, forklift approximately 20 mm). Clearly these differences mean that each robot is better suited to particular tasks. It is important to note that they could work together to accomplish tasks that neither could accomplish on their own. This is certainly not an idea that only applies to robotics. Nature is full of examples of symbiosis, two species with different capabilities working together to their mutual benefit. Industry has seen the advantages of interdisciplinary teams working together to benefit from the differing capabilities of the team members [15], and ABET has required the use of such teams in current engineering curricula [16]. Humans have worked with machines to take advantage of the strengths of both (for example, a crane will bear the load of a steel beam while a human positions it accurately for assembly). Another example of symbiotic human-machine collaboration is a typical cherry-picker such as the one shown in Figure 1 that is being used in a construction task. Here the machine is used because of its ability to position the human close to the workpiece. The human, who is far more precise and maneuverable, then moves on the platform and does the actual work of assembly. It should also be noted that the human operator repositions the cherry-picker as needed to complete the task successfully.

The authors decided to use this cherry-picker task as a test case for the coordination of multiple mobile manipulators. The large, high-reaching forklift was retro-fitted with a wooden platform and served as a ‘cherry-picker.’ The lighter, more accurate rover essentially functioned as the ‘human’ in the figure, doing the fine work and specifying the details of the task. Since this rover is equipped with only a 1-DOF arm, an ‘inspection’ task was chosen; the goal of the experiment was to position the rover's end-effector relative to a tear in a large, overhead door. This task requires a robot to touch or place an instrument at a location (about which no information is known *a priori*) designated by the user. The level of precision required cannot be achieved by the forklift and the 3-D point selected is outside of the reach of the small, precise rover. To accomplish the task, the forklift has to elevate and transport the rover.

3.3 Approach

The coordination of these two mobile manipulators to complete the task is based on the ability of both manipulators to use MCSM to position themselves relative to a target. For the human-controlled cherry picker described earlier, the human is responsible for task coordination and completion, thus controlling the entire process. In a similar manner, the rover will have high-level control of this task. Several goals were important in the development of this approach, including:

- *Minimal human interaction*: Currently, the target for the inspection task is specified through a GUI (described in [1]) that shows (to a human supervisor) the image-plane of one of the rover cameras. The operator then uses a mouse to specify the target location in this image plane. This is the only supervisor interaction within the system.
- *High accuracy:* The rover used in these experiments has shown the ability to position its end-effector repeatably with an accuracy of a few millimeters when tracking natural features and under one millimeter when using fiducials. While the forklift is not as accurate (typically 20 – 50 mm), the goal of this coordinated task is to take advantage of the strengths of both systems. The coordinated system, then, should be capable of accuracies similar to that of the rover itself. This is only possible if the task goals are communicated accurately.
- *Robustness:* The system must be able to function reliably despite changing conditions. Because the 'cherry picker' experiment requires the placement of the robots close to solid objects, any errors are likely to lead to collisions. Any interaction between the two systems must have a failsafe to minimize such errors.
- *Low communication requirements:* While information-intensive tasks such as mapping would require large amounts of communication bandwidth, the "cherry-picker" experiment should not. The algorithm developed allows coordination of the two robots with only a few digital hand-shaking lines. Targeting information is shared between systems in much the same way that current military targeting is accomplished – by illuminating ('lighting up') the target with a laser [17]. While the idea of using a laser to enhance robotic accuracy is not new ([1,18,19]) it is novel to use this as the primary means of sharing target information between multiple robots. The simplicity of the approach reduces the required network overhead and bandwidth. More importantly for the high-precision applications discussed here, using a laser to illuminate the

target eliminates the need to calibrate and correlate the coordinate systems of the multiple mobile robots, reducing inaccuracies that result from the ensuing (and necessarily imperfect) coordinate transformations.

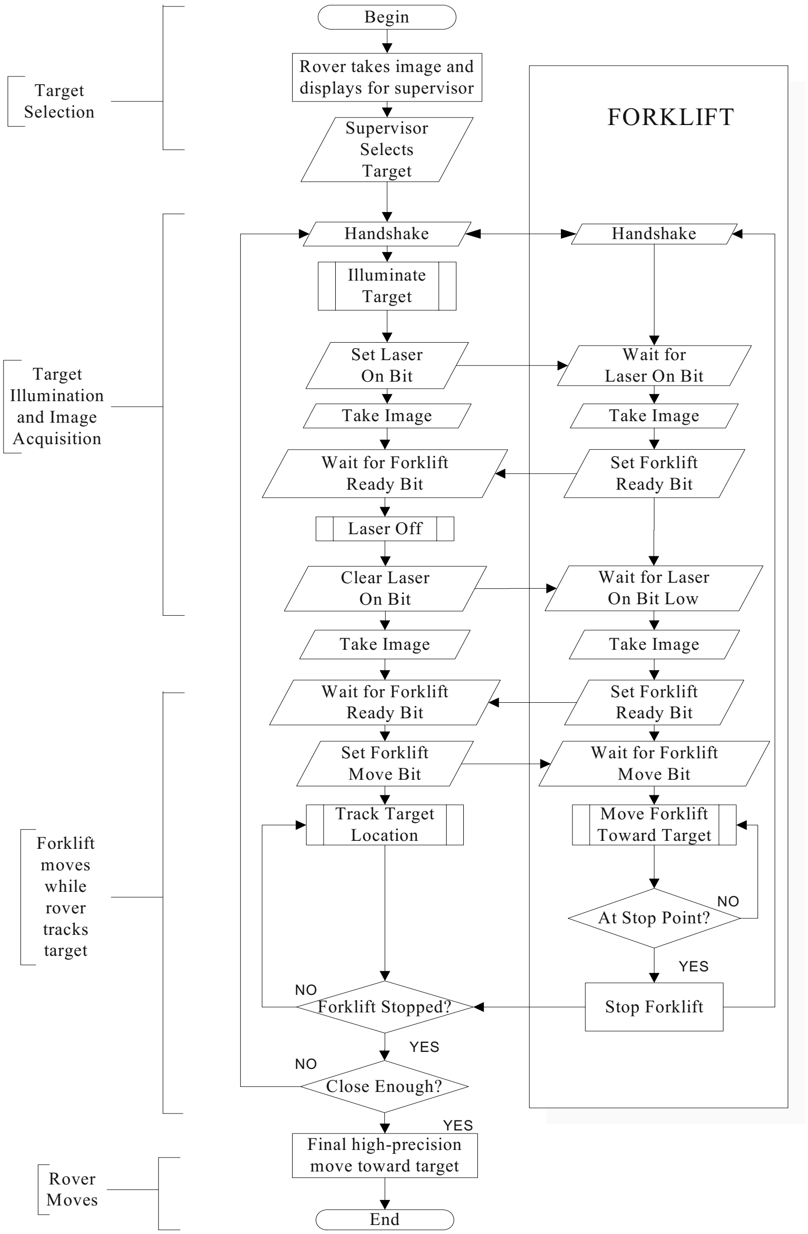


Figure 2: Cooperation Algorithm Flowchart

4 Results

4.1 Algorithm

The algorithm for this task is shown in Figure 2 in flowchart form. A human supervisor begins the task by selecting (via a mouse) a target point in the image plane of one of the rover's cameras. Once this is complete, a simple digital handshake between the two systems ensures that both systems are ready to work on the task.

Next, the systems work together to acquire visual information about the target. The pan-tilt unit onboard the rover is used to move a laser spot onto the target. This is done using visual servoing, and has been previously described in [1]. Once the laser is in place, the rover signals the forklift that the laser is on the target, and the forklift takes an image. At this point, both systems have separate images showing the illuminated target. To make the image-processing algorithms as robust as possible, image differencing is used to find the target location. This is done by taking a second image with the laser turned off (which requires additional handshaking with the forklift). Once these images are taken, both systems find the location of the target. The two systems have shared target information visually, without having to make reference to any external, common coordinate system. The only requirement is that they both have visual access to the target location.

When both systems have acquired the target, the rover sends a digital signal asking the forklift to move when ready. MCSM is used by the forklift system to generate both a target height to raise the rover to as well as a trajectory to get the forklift close enough for the rover to touch the target. The forklift will begin moving along this trajectory, while the rover tracks the visual target in real-time. This visual tracking is based on an algorithm originally developed by researchers at Carnegie Mellon University and adapted for use in MCSM [1]. The forklift moves a set distance along the trajectory and stops. If the forklift is 'close enough' to the target that the rover will be able to complete the task, the forklift stops and is finished with its part of the task. If it is not close enough, the procedure is repeated, and the forklift moves closer. Once the forklift is close enough, the rover begins to move. Again, this motion is controlled using the MCSM algorithms.

4.2 Experiments

Current experiments have all been conducted in an indoor, warehouse environment. The experiments for this project have been based on the inspection of a tear in a large, overhead door. The task begins with the user clicking on this tear in the image-plane of one of the rover's cameras. In our experiments, accuracies (final placement of the rover end-effector relative to the selected target) on the order of a few millimeters have regularly been achieved. These experiments have shown success in meeting the four goals of the project. Only one human interaction is required for task specification, only four digital I/O are required for communication, accuracy has not degraded by involving multiple robots, and the system has been successful in repeated trials.

Figure 3 shows a sequence of images taken during an experiment in which a tear in the overhead door is used as the target. The final frame shows the end-effector has reached this target. Video of this experiment is available at [20].

5 Conclusion

Experimental results indicate that the approach developed here can be successful in controlling two heterogeneous mobile manipulators in order to complete accurate positioning tasks. The algorithm that was developed met the four primary goals of the approach. Human interaction is limited to a single data exchange – the supervisor is shown an image from one of the robot's cameras, and selects a target. High accuracy was maintained, achieving target position within a few millimeters. The system performed repeatably, using minimal communication bandwidth while maintaining a handshaking protocol to ensure the two systems continued to function correctly.

It is the authors' hope to expand the work presented here to allow for additional robots. A next step is likely to include a traditional, fixed robot to add additional diversity to the types of tasks being addressed. Another expansion of this work would be the addition of a separate pan-tilt-zoom camera for target selection. This would allow for much higher accuracy, since the largest current source of inaccuracy in tracking the target location as the robots move.

Clearly, the approach described here does NOT make any attempt to address the 'general' problem of the coordination of multiple robots. The frameworks described in [12,21] are far better suited for such tasks. They can be used to allow the sharing of much more information about tasks, capabilities, and maps. The authors plan to work on 'packaging' the capabilities described in this paper as behaviors that could then become part of one of these larger frameworks.

It should also be noted that the work here relies heavily on MCSM. This algorithm is completely predicated on the ability of mobile manipulators to accurately position themselves relative to a visually-identified target. The approach is not generalizable to systems that do not include at least one mobile manipulator with this basic capability.

The work presented here provides experimental evidence of cooperating, heterogeneous systems working to complete a positioning task that neither could complete independently. This approach will be applicable to many cooperative positioning tasks, including assembly, construction, inspection, and maintenance. Such tasks typically will require heterogeneous robotic systems, and are likely not to require detailed maps, as the work is being done relative to another object rather than at a numerically-specified coordinate in three-dimensional space.

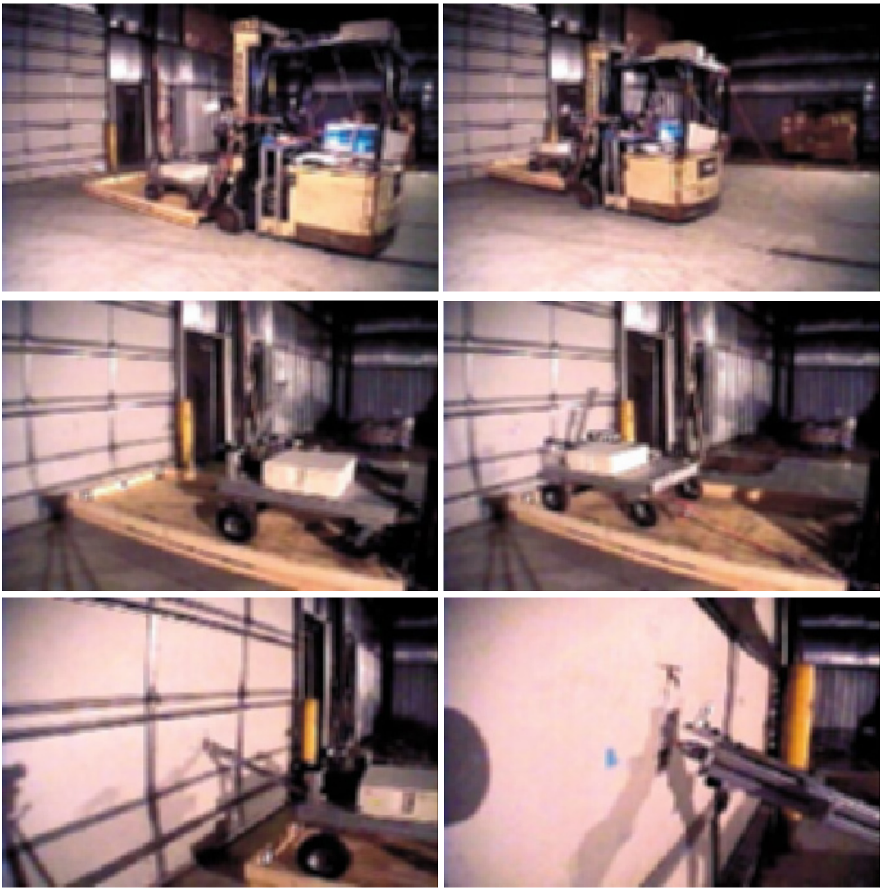


Figure 3: Simulated 'cherry-picker' cooperation task.

References

[1] M. Seelinger, *et al*. "High-Precision Visual Control of Mobile Manipulators," *IEEE Trans. on Robotics and Automation*, Vol. 18, No. 6, pp. 957-965, 2002.

[2] A. Howard, M. Matarić, and G. S. Sukhatme. "Localization for Multiple Robot Teams: A distributed MLE Approach," In *Proc. ISER'02: Intl. Symposium on Experimental Robotics*, Naples, Italy, July 2002.

[3] S. Sukkarieh *et al*. "Cooperative Data Fusion Amongst Multiple Uninhabited Air Vehicles," In *Proc. ISER'02 : Intl. Symposium on Experimental Robotics*, Naples, Italy, July 2002.

[4] A. Trebi-Ollennu , *et al.*, "Mars Rover Pair Cooperatively Transporting a Long Payload," In *Proc. IEEE Intl. Conference on Robotics and Automation*, Washington, D.C. May 11 - 15, 2002.

[5] G.A.S. Pereira *et al*. "Cooperative Transport of Planar Objects by Multiple Mobile Robots using Object Closure," In *Proc. ISER'02: Intl. Symposium on Experimental Robotics*, Naples, Italy, July 2002.

[6] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous Robots*, vol. 4, pp 1-23, 1997.

[7] L. E. Parker, "Current state of the art in distributed robot systems," in *Distributed Autonomous Robotic Systems 4*, pp 3-12, Springer-Verlag 2000.

[8] D. Rus, *et al*., "Moving Furniture With Teams of Autonomous Robots," In *Proc. IEEE/RSJ IROS*, pp. 235-242, 1995.

[9] M. N. Ahmadabadi and E. Nakano, "A 'Constrain and Move' Approach to Distributed Object Manipulation," *IEEE Trans. on Robotics and Automation*, Vol. 17, No. 2, pp. 157-172, 2001.

[10] R. Fierro *et al*., "A Framework and Architecture for Multi-Robot Coordination," *Int. Journal of Robotics Research*, Vol. 21, No. 10-11, pp. 977-995, 2002.

[11] O. Khatib, *et al*., "Vehicle/Arm Coordination and Multiple Mobile Manipulator Decentralized Cooperation," Proc. *IROS* , pp. 546-553, 1996.

[12] R. Simmons, *et al*., "First Results in the Coordination of Heterogeneous Robots for Large Scale Assembly," In *Proc. Intl. Symposium on Robotics Research*, Honolulu, Dec 2000.

[13] C. Castelpietra, *et al*., "Communication and Coordination among heterogeneous Mid-size players: ART99," In *Proc. of 4th Intl. Workshop on RoboCup 2000*, Melbourne, Australia, 2000.

[14] B. P. Gerkey and M. J. Mataric´. "Sold!: Auction methods for multi-robot coordination," *IEEE Trans. on Robotics and Automation, Special Issue on Multi-robot Systems*, Vol. 18, No. 5, pp. 758-768, 2002.

[15] G. Dieter "*Engineering Design: A Materials and Processing Approach, 3rd ed.*" McGraw Hill, 2000.

[16] "Criteria for Accrediting Engineering Programs, Effective for Evaluations during the 2004-2005 Accreditation Cycle," ABET, 2003. (http://www.abet.org/criteria.html)

[17] B. Anderberg and M. L. Wolbarsht, *Laser Weapons The Dawn of a New Military Age*. Plenum Press, New York, 1992.

[18] M. Seelinger, *et al*. "Point-and-Click Objective Specification for a Remote, Semi-Autonomous Robot System," *Sensor Fusion and Distributed Robotic Agents, Proc. SPIE 2905*, Boston, MA, Nov. 21-22, 1996, pp. 206-17.

[19] A. Krupa, *et al*. "Towards semi-autonomy in laparoscopic surgery: first live experiments," In *Proc. ISER'02: Intl. Symposium on Experimental Robotics*, Naples, Italy, July 2002.

[20] J-D Yoder Web site: http://www2.onu.edu/~jyoder/

[21] T. Huntsberger *et al*. "CAMPOUT: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration." *IEEE Trans. on Systems, Man and Cybernetics, Part A*, Vol. 33, No. 5, pp. 550-559, 2003.

Implementation of an Indoor Active Sensor Network

Alex Brooks, Alexei Makarenko, Tobias Kaupp,
Stefan Williams, and Hugh Durrant-Whyte

ARC Centre of Excellence in Autonomous Systems (CAS)
The University of Sydney, Australia
{a.brooks, a.makarenko, t.kaupp, s.williams, hugh}@cas.edu.au
http://www.cas.edu.au

Abstract. This paper describes an indoor Active Sensor Network, focussing on the implementation aspects of the system, including communication and the application framework. To make the system description more tangible we describe the latest in a series of indoor experiments implemented using ASN. The task is to detect and map motion of people (and robots) in an office space using a network of 12 stationary sensors. The network was operational for several days, with individual platform coming on and off line. On several occasions the network consisted of 39 components. The paper includes a section on the lessons learned during the project's design and development which may be applicable to other heterogeneous distributed systems with data-intensive algorithms.

1 Introduction

A large number of autonomous sensing platforms connected into a network promise better spatial coverage, higher responsiveness, survivability and robustness compared to a single vehicle solution. The modular design may also lead to lower costs, despite the increase in complexity of the overall system. The need for such systems exists in many applications involving tasks in which timely fusion and delivery of heterogeneous information streams is of critical importance. Examples include military and civilian surveillance, fire fighting, intelligent buildings, etc.

The Active Sensor Network (ASN) project at the University of Sydney aims to combine decentralized data fusion and control algorithms into a unified yet flexible system architecture suitable for a wide range of sensing tasks. The ASN can be described along three dimensions: the architecture [16], the algorithms [17], and the concrete implementation which is the focus of this paper.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 provides a brief background on the ASN project. The experiment and implementation details are covered in Section 4. Section 5 discusses lessons (positive and negative) learned during the system development. It also comments on current and future work based on these lessons.

2 Related Work

As a rule, distributed systems are more complicated than monolithic ones. To keep the system implementation tractable, our approach employs two design principles:

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 397–406, 2006.

Table 1. Non-functional requirements for different SN categories. Those having direct impact on implementation approach are highlighted.

Requirements	μSN	MSN	MRS
Small unit size	✓		
Long mission duration	✓	✓	
Large team size	✓	✓	
Heterogeneous team		✓	✓
High information accuracy		✓	✓
Low information latency			✓

the use of components which enforces modularity and enables code re-use, and the use of middleware which hides the complexity of component interaction. There has been a strong trend towards using both of these approaches in robotics, but different communities within the Sensor Network (SN) field use them to a different degree. We find it convenient to view the SN research broadly divided into three categories: multi-robot systems (MRS), macro SN (MSN) and micro SN (μSN). The divisions are determined by application domains expressed as non-functional requirements in Table 1. Different drivers naturally lead to very different solution approaches.

μSN. For various domain-specific reasons, small unit size is considered a requirement in these systems. In its most extreme form-factor, these systems are known as "smart dust" [12]. Until recently, most work has been done in simulation, but with the advent of commercially available Berkeley Motes some experiments have been performed [2] [8]. A typical deployment would involve a very large number of small identical units designed by a single team. Combined with limited processing capacity, this makes the need for modality secondary to efficiency (primarily in terms of energy consumption). Severe hardware constraints also preclude the use of standard middleware (as well as OS's, communication stacks, etc.) Specialized middleware is being developed by several groups [11].

MRS. Collecting environment information is not the main objective in these systems but information gathering and sharing is performed as part of the domain-specific task (e.g. in RoboCup [4]). Platforms are often heterogeneous but the team size is small so, just as with μSN, component-based design is of limited value [19]. The situation is beginning to change with experimental deployments reaching the 100 robot barrier. E.g. the CentiBOT project [13] uses Jini for inter-platform communication – a general-purpose Java middleware. Some modern robotic architectures, e.g. Joint Architecture for Unmanned Systems (JAUS) [9], are defined in terms of components. JAUS intends to be distributed and open and therefore can be in principle extended to the sensor network level but currently is not. It defines its own custom message-based middleware. OROCOS [1] [14] is also primarily intended for implementing single vehicle architectures. It is component-based from the ground up and is built on top of a CORBA middleware implementation.

MSN. These networks are comprised of large numbers of capable nodes and are required to operate for long periods of time. These capable nodes are often required to process rich data streams or control arbitrarily complex actuators. The power required for communication is very small compared with locomotion and active sensing. The extra processing power means that complex probabilistic data fusion algorithms can be executed. It also allows use of standard software tools including middleware. Large heterogeneous teams, likely to be designed by different organizations, strongly encourage modular component-based design. The system described in this paper is an example of a MSN. We are not aware of current deployments of other MSN's described in open literature. There is also substantial military interest in MSN's where the doctrine of network-centric warfare aims at assembling a global-scale network of very capable platforms [3].

3 The Active Sensor Network Project

We seek a solution to the problem of distributed information gathering (DIG). We consider a distributed phenomenon which can be described by a state vector x. There is a set of heterogeneous robotic platforms equipped with sensors and actuators. There is also a set of operators who monitor the phenomenon directly using human senses or by interacting with the network through a user interface (UI). We think of all the entities, human and robotic, as members of a team.

The functional requirements of the DIG problem cover three broad areas: a) collecting information (which includes sensing, information fusion, information dissemination, and sensor management); b) interacting with human operators; and c) reconfiguring the system in dynamic environments. Some of the desirable qualities of the solution are: network scalability, robustness to failure of individual components, information accuracy, etc. Additional constraints may limit design choices, e.g. platform size and energy budget, stealth of operation, privacy concerns, etc. It is unlikely that a single approach will be able to solve all possible permutations of this problem. Nevertheless, we are interested a conceptual framework which is flexible enough to allow wide variations in non-functional requirements.

ASN is an architecture for *cooperative autonomous* sensing platforms [16]. Autonomy implies that a platform is able to work in isolation and does not rely on infrastructure services, remote control, or other external inputs. Cooperative means that the platforms share common goals and, when possible, work together to achieve them. Platform are likely to have different capabilities but each comes equipped with power, processing and communication facilities, sensors and actuators. Each one fuses local observations with information communicated from neighboring nodes into a synchronized view of the world. Similarly, each one makes local control decisions based on the knowledge of local platform capabilities and the global synchronized world view.

The ASN system is composed of software components communicating asynchronously with each other. Several component types are defined, each with specific roles. With respect to environmental information, a component can be a source (producer), a sink (consumer), or a fuser/distributor. Similarly, with respect to control commands, a component can be a source (decision maker) or a sink (controlled object). A particular component can play several of these roles at once. To make the reference clear, the component types will be capitalized.

The fundamental principle of the ASN architectural style is decentralization. Compared to a centralized or a distributed system, a decentralized system is characterized by two key constraints: a) no central services and facilities and b) no knowledge of global topology. The resulting system offers a number of advantages over other architectures. *Scalability:* the computational and communication load at each node is independent of the size of the network. *Robustness:* no element of the system is mission critical, so that the system is survivable in the event of run-time loss of components. *Modularity:* components can be implemented and deployed independently from each other. The Bayesian Decentralized Data Fusion (BDDF) algorithm and its extension to information-based control were described in [17].

4 Implementation

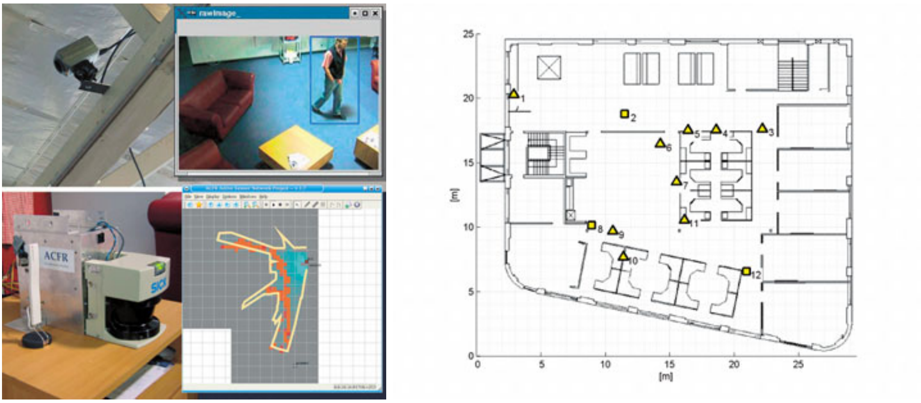
4.1 The Experiment

To make the system description more tangible we describe the latest in a series of indoor experiments implemented using ASN. The aim of the experiment is to use a dynamically configured sensor network to monitor motion in an office environment over a extended period of time. Figure 1(a) shows two types of physical sensors used to detect motion: video cameras and laser scanners. The sensor network was deployed by hand so as to achieve reasonable coverage, as shown in Figure 1(b). Each sensing platform has a processor and executes three software components: a Frame, a Sensor, and a Node.

The Frame is responsible for estimating its position in the global coordinate system. In these experiments the platform poses were surveyed manually and specified on initialization. The Sensor is responsible for reading and pre-processing measurements from the sensing hardware and generating observations using a model of the physical sensor. The pose required to transform local range-bearing measurements into global observations is obtained from the Frame. The Node is responsible for maintaining and sharing a view of the global state using decentralized data fusion algorithms. The estimate is updated using observations from Sensors and information communicated by other Nodes.

The global state of the world is stored in Decentralized Certainty Grid representation [18] – a decentralized version of the original [5]. The space is represented by a grid large enough to cover the 30x25m building. Each cell contains the probability that something is moving in the 30x30cm part of the office it represents. Observations increase the certainty of motion in the cell. In the absence of observations, the certainty of the cell state gradually decreases (the entropy of the distribution increases).

When most of the sensing platforms have been positioned, the first Node is started, initializing its certainty grid with an uninformative prior. This moment marks the beginning of Network Up-Time. Because new Nodes are supplied with



(a) (b)

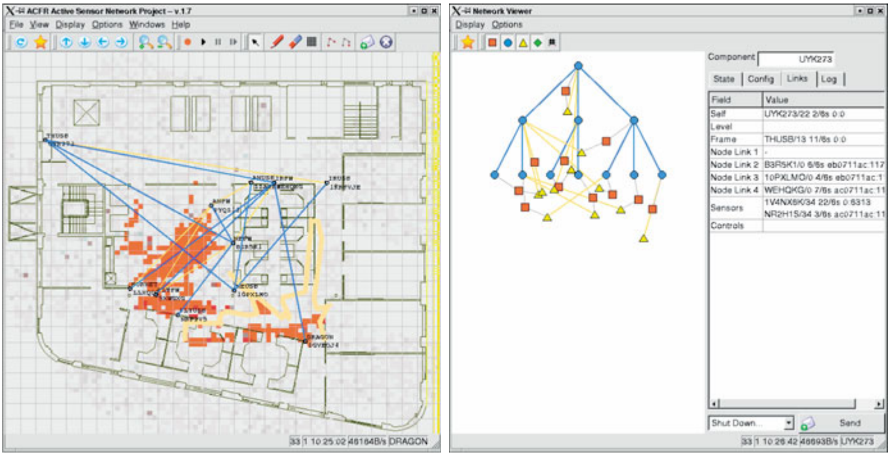
Fig. 1. Indoor sensor network: (a) sensor hardware and (b) deployment throughout the office: cameras are shown as triangles, lasers as squares.

the current estimate of the world state on start-up, this is the moment from which no information will be lost as long as at least one Node remains operational.

One or more operators may monitor the network operation using a Graphical User Interfaces (GUI). The world view of the GUI, shown in Figure 2(a), displays the state of the world as seen from the platform to which the GUI is connected. The motion grid, from the point of view of the Node in the lower-right corner, is shown in shades of gray and red. The dark red color indicates higher certainty of motion in that cell. The large dark red blob near the centre of the image corresponds to a meeting in progress. The floor plan of the building is overlaid on top of the motion map for clarity. The network view of the GUI, shown in Figure 2(b) shows the topology of the network, which defines communication paths. Blue circles indicate Nodes, and the blue lines indicate connections between Nodes. Red squares indicate Frames and yellow triangles indicate Sensors. The right-hand panel shows details of the connections of an individual component.

4.2 Application Framework

ASN is an application framework implemented as a set of C++ libraries on Linux platforms. Currently the system consists of 250 classes implemented in roughly 50,000 lines of code. Included in this are approximately 40 classes implementing the framework which provides the base functionality and infrastructure. The bulk of the generic functionality provided by the library is in automatic system configuration: establishing, maintaining, and re-establishing service connections between the components. The rest of the code implements specific component types. The algorithms for data fusion, sensor management and platform control were largely reused from previous projects. A lot of the effort was saved by across-the-board reliance on


ACFR Active Sensor Network Project – v.1.7
File View Display Options Windows Help
33 1 10:25:02 46164B/s DRAGON
Network Viewer
Display Options
Shut Down...
Send
33 1 10:26:42 46693B/s UYK273

Component UYK273

State | Config | Links | Log

Field	Value
Self	UYK273/22 2/6s 0:0
Level	
Frame	THUSB/13 11/6s 0:0
Node Link 1	-
Node Link 2	B3RSK1/0 6/6s eb0711ac:117
Node Link 3	10PXLMO/0 4/6s eb0711ac:1
Node Link 4	WEHQKG/0 7/6s ac0711ac:11
Sensors	1V4NX6K/34 22/6s 0:6313 NR2H1S/34 3/6s ac0711ac:11
Controls	

Fig. 2. A view of the sensor network in operation, as seen from the operator's GUI: (a) the view of the environment and the platforms and (b) the topology of the network.

hardware abstraction and device-level simulation provided by Player/Stage [7] – an open source project originating at the USC robotics group.

Each component is encapsulated within a separate process. All together, 24 different component types have been or are being implemented. Five Frames: a stationary "box", a simulated Stage platform, a mobile self-localizing Pioneer, a self-localizing stationary camera, and a pan-tilt-zoom camera. Ten Sensors: a partially filled matrix of 3 sensor modalities (laser, vision, and sonar), 3 feature types (stationary target, occupancy, and motion), and 3 feature representations (Gaussian points, Certainty Grid, and grid-based general *pdf*). Three Nodes: one for each feature representation. Three Controllers: exploration using occupancy grid, point feature information surfing, and Bayesian search. Two Operator UI's: a desktop and a hand-held versions.

4.3 Inter-Component Communications

ASN uses message-based communication. A set of messages are defined, which are decipherable by all components according to a *MessageType* field in the message header. Messages are implemented as C++ classes responsible for their own marshalling and de-marshalling.

Similar to [20] we identify three communication types needed to implement our architecture: 1) global; 2) local; 3) point-to-point. Global distribution is facilitated by the Nodes and is implemented using component-to-component communications. It is used to propagate environment information, team control priorities, and other information with global scope. Local distribution uses broadcast which may be constrained by the use of communication channels. Components subscribe to channels organized by information type. This method is used for run-time system configuration. Addressed point-to-point communication forms the backbone of our system. This method is used in sensor-to-node, node-to-node, and most other links.

All three communication types are facilitated by a single library which works as follows. Each host runs a daemon process which is responsible for message routing. On start up, component processes register with the daemon and specify the channels to which they want to listen. Daemons maintain a table of other daemons within communication range and the channels to which they are subscribing. This is done using a custom low-bandwidth broadcast protocol. The daemon is indistinguishable from any other communicating process except that it provides a “gateway” or forwarding facility to other remote hosts. Messages are sent from one component process to another in the following manner. If the destination component is local, the process simply writes its message contents to the addressee’s shared memory. If the message is inter-host, the process first writes to the daemon’s shared memory which then sends an addressed UDP packet to the daemon on the receiving host who forwards it to the appropriate process. Unaddressed (broadcast) messages are sent to daemons in the neighborhood subscribing to the particular channel who forward it to all subscribing processes on the same host.

5 Lessons Learned

Implementation of any design inevitably highlights its strengths and weaknesses. The purpose of this section is to share some of the insights gained by the project team during this process, by outlining what worked, what did not work, and what direction the project is taking as a result of this experience.

5.1 What Worked

Our experience has validated the benefits of open decentralized system architecture. Such systems have a higher implementation entry barrier, but once the basic software infrastructure is in place, it becomes relatively easy to extend the system both physically and functionally. Increasing system capability by adding new platforms is a simple as starting up the necessary components and all connections are done automatically. The system is capable of covering the entire building and supplying enough processing power to process images from many cameras in real-time. The system’s performance degraded gracefully with inevitable node failure and outlived any given Node. The experiment also demonstrated scalability in terms of communication requirement at any particular Node.

Breaking up the system into independent interacting components was also a success. This approach allowed to design, implement, and debug components independently form each other. The message-based middleware approach proved quite adequate for this task. Component-based design also allowed great flexibility in system deployment. Each component can execute on any host as long as it is connected to the network. Implementation of this feature did not require much effort due the use of a location-transparent communication mechanism.

5.2 What Did Not Work

The ASN architecture, as it stands, is fairly inflexible. A small set of fundamental component types are defined, and links between components are defined in terms of the component type on either end. This highly prescriptive architecture is tailor-made for the current system design, however it does not easily generalize to other problems. A more flexible and extensible approach is to define links between components in terms of services, and to leave the granularity of component implementation up to the designer. This way of structuring the problem, with reference to the ASN framework, is described further in [16]. More importantly, it transpired that a number of issues were hidden behind the minimalist framework requirements given in Section 5.1 [6]:

Marshalling and De-Marshalling Complex objects need to be converted to serial form before transmission over a network, then re-assembled on receipt. Additionally, the byte-order for basic data types depends on the host architecture. The task of writing these functions is tedious and error-prone, and must be done for each and every object type unless a general solution is identified. This is particularly difficult for variable-length objects or objects with complex internal structure.

Naming and Addressing ASN had to design a system for assigning unique IDs, with no central ID server, and for mapping from names to IDs. These IDs then need to map to the IP address of a host and a process ID on that host.

Sending Large Objects The ASN communication mechanism can't deal with re-assembling fragmented datagrams. Components therefore need to ensure their messages are small enough, or deal with fragmentation and reassembly themselves.

Routing across subnets Since ASN relies on broadcast UDP, this isn't possible without designing a special ASN message router.

Type Safety ASN has no mechanism to ensure that operations requested on a server actually exist on that server, or that the parameters supplied by a client have matching definitions on the server. This is the source of many potential bugs.

Communication Patterns Synchronous and asynchronous communication patterns are usually needed. ASN had to design these from scratch.

OS and Language Independence Since an ASN components communicate through a well-defined set of messages, in principle a client could be written in any language on any platform. In practice, writing a Java client would involve substantial effort including re-implementing solutions to all the problems discussed. This barrier would be lowered if the application framework and the communications framework weren't so tightly coupled.

Automatic Service Discovery ASN offers a simple ping-reply mechanism for components to connect dynamically. When a component starts up it broadcasts a special message which specifies which of the pre-defined component types it is looking for. A more complete mechanism would allow components to advertise the services they require.

The ASN solution to these issues involved a great deal of work, the solutions are incomplete, and the implementations are sometimes buggy. The primary lessons learned from the implementation of this project are an appreciation for the size of the task and a realization that re-inventing the wheel is unnecessary.

5.3 Future Directions

The challenge of building large distributed applications is not unique to robotics or sensor networks. It has been the subject of intense study in the field of component-based software engineering [10]. A practical solution has emerged in the form of middleware – “a layer between network operating systems and applications that aims to resolve heterogeneity and distribution” [6]. ASN has essentially designed its own middleware, built into the ASN application framework.

Unfortunately the task of designing a robust, flexible, efficient, and well-documented middleware package is not trivial. This fact is highlighted by the numerous deficiencies found in ASN and listed in Section 5.2. In light of this, it would be desirable, at least for some tasks, to be able to adopt a general-purpose middleware package. In other aspects of hardware and software usage, the trend towards using off-the-shelf components and tools in main-stream robotics is already well established. This applies to processors, networking, serial buses, operating systems and programming languages. We believe that in the MSN field it is possible to re-use all of the above plus the middleware and only design specialized application code. In some applications, however, the μSN field being a prominent example, the hardware, operating system and communication stacks are still highly specialized. It is very likely that for those applications, specialized middleware will be required as well.

In view of these lessons, direction for future research is to re-design the ASN framework on top of a standard middleware package. Three general-purpose distributed component models exist today: OMG’s CORBA, Microsoft’s COM+, and Sun’s Enterprize Java Beans [15]. CORBA’s distinct advantage is that it is language-agnostic, can run on Linux, and has open-source and real-time implementations. It is anticipated that the adoption of a standard middleware package will reduce the effort required in implementing communications and allowing inter-operability of components, producing shorter development cycles and promoting code re-use. In addition, the adoption of standard software will lower the barrier for sharing code between institutions, making collaboration easier.

6 Conclusion

This paper has described a long-term, large-scale experiment with an indoor sensor network, paying particular attention to how the system was implemented. This implementation brought to light some of the complexities that must be dealt with when building distributed, and particularly decentralized, systems. A result of this work is that future effort will be directed towards examining the use of standard middleware packages for sensor networks.

Acknowledgement

This work is supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. The authors would like to thank Matt Ridley for his communications library and continual Linux support.

References

1. H Bruyninckx. Open robot control software: the OROCOS project. In *IEEE ICRA*, volume 3, pages 2523–28, 2001.
2. A Cerpa, J Elson, D Estrin, L Girod, M Hamilton, and J Zhao. Habitat monitoring: application driver for wireless communications technology. *SIGCOMM Computer Comm. Review*, 31(2 suppliment):20–41, 2001.
3. Solipsys Corp. Tactical component network: Overview. White paper, www.solipsys.com, 2000.
4. M Dietl, J-S Gutmann, and B Nebel. Cooperative sensing in dynamic environments. In *IEEE/RSJ IROS)*, pages 1706–13, Maui, Hawaii, USA, 2001.
5. A Elfes. Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework. In *Int. Workshop on Reasoning with Uncertainty in Robotics*, pages 93–129, Amsterdam, Netherlands, 1995.
6. W Emmerich. *Engineering Distributed Objects*. John Wiley and Sons, Ltd., 2000.
7. BP Gerkey, RT Vaughan, and A Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Int. Conf. on Adv. Robotics*, Coibra, Portugal, 2003.
8. L Girod, J Elson, A Cerpa, T Stathopoulos, N Ramanathan, and D Estrin. Emstar: a software environment for developing and deploying wireless sensor networks. In *USENIX Tech. Conf.*, 2003.
9. JAUS Working Group. The joint architecture for unmanned systems. Tech. report, www.jauswg.org, Feb 2004.
10. GT Heineman and WT Councill, editors. *Component-based software engineering: putting the pieces together*. Addison-Wesley, Boston, 2001.
11. WB Heinzelman, AL Murphy, HS Carvalho, and MA Perillo. Middleware to support sensor network applications. *IEEE Network Mag.*, 18(1):6–14, 2004.
12. J Kahn, R Katz, and K Pister. Emerging challenges: Mobile networking for “smart dust”. *J. of Comm. Networks*, pages 188–196, 2000.
13. K Konolige, C Ortiz, R Vincent, A Agno, M Eriksen, B Limketkai, M Lewis, L Brieseister, and E Ruspini. Centibots: Large scale robot teams. Tech. report, SRI International, 2002.
14. W Li, HI Christensen, A Oreback, and D Chen. An architecture for indoor navigation. In *IEEE ICRA*, New Orleans, LA, 2004.
15. A Longshaw. Choosing between COM+, EJB, and CCM. In GT Heineman and WT Councill, editors, *Component-based software engineering : putting the pieces together*. Addison-Wesley, Boston, 2001.
16. A Makarenko, A Brooks, S Williams, H Durrant-Whyte, and B Grocholsky. A decentralized architecture for active sensor networks. In *IEEE ICRA*, New Orleans, LA, 2004.
17. A Makarenko and H Durrant-Whyte. Decentralized data fusion and control in active sensor networks. In *Int. Conf. on Info. Fusion*, Stockholm, Sweden, 2004.
18. A Makarenko, SB Williams, and HF Durrant-Whyte. Decentralized certainty grid maps. In *IEEE/RSJ IROS*, pages 3258–63, Las Vegas, NV, 2003.
19. RG Simmons, D Apfelbaum, W Burgard, D Fox, M Moors, S Thrun, and H Younes. Coordination for multi-robot exploration and mapping. In *AAAI Nat. Conf. on AI*, pages 852–58, Austin, TX, 2000.
20. L Subramanian and RH Katz. An architecture for building self-configurable systems. In *Workshop on Mobile and Ad Hoc Networking and Computing*, pages 63–73, 2000.

An Indoor Experiment in Decentralized Coordinated Search

Frédéric Bourgault, George Mathews, Alex Brooks, and Hugh F. Durrant-Whyte

ARC Centre of Excellence for Autonomous Systems (CAS)
Australian Centre for Field Robotics
The University of Sydney, Sydney, NSW 2006, Australia
{f.bourgault, g.mathews, a.brooks, hugh}@acfr.usyd.edu.au
http://www.acfr.usyd.edu.au

Abstract. This paper addresses the problem of coordinating a team of multiple heterogeneous sensing platforms searching for a single mobile target in a dynamic environment. The proposed implementation of an active sensor network architecture combines a general decentralized Bayesian filtering algorithm with a decentralized coordinated control strategy. In this approach, by communicating with their neighbors on the network, each decision maker builds an equivalent representation of the probability density function of the target state on which they base their control decision.

1 Introduction

Search and Rescue (SAR) is a forthcoming domain of application in field robotics. Interconnecting multiple autonomous search platforms into a network to share information and collaborate has the potential to increase the efficiency and the rate of success of such time critical missions where human life is at stake.

This paper addresses the problem of coordinating the search effort of an arbitrary number of decentralized sensing platforms. It presents an indoor implementation of the active Bayesian sensor network approach discussed in [1]. The approach combines a general decentralized Bayesian filtering technique with point-to-point communication and a decentralized coordinated control scheme. By adaptively exchanging information with their network neighbors, each sensor node builds an equivalent estimate of the Probability Density Function (PDF) of the target state on which they base their local control decisions. The information is transmitted between the nodes via managed channel filters guaranteeing the robust recovery of the complete global information at each node provided that the network connectivity is acyclic, i.e. tree-connected without loops. By mutually contributing to their PDF estimates, the decision makers influence each other, rendering their trajectories globally consistent and coordinated [3,9]. The resulting coordinated search trajectories explicitly consider the search vehicles kinematics, the arbitrary detection function of the sensors and the target motion model. The advantage of this modular approach is that a high degree of scalability and real time adaptability can be achieved.

The decentralized coordinated search framework was first explored in [3] where faultless broadcasted communication was assumed. The idea of integrating external motion constraints such as walls into the prediction stage was presented in [2]. A different search framework using Bayesian ideas can be found in [7] while [15] uses game theory combined with heuristics to find evading targets.

The paper is organized as follows. Sec. 2 reviews the decentralized Bayesian filtering algorithm and the decentralized coordinated control strategy implemented

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 407–416, 2006.

in the architecture. Sec. 3 describes the searching problem. Then, Sec. 4 discusses the implementation details of the framework for a team of Pioneer robots searching for a person in a cluttered indoor environment and presents experimental search results. Finally, conclusions and ongoing research directions are highlighted.

2 Architecture

This section describes the framework for coordinating the search effort of a robotic team based on an Active Sensor Network (ASN) architecture [11]. ASN combines decentralized data fusion and decentralized control into a coherent architecture.

2.1 Decentralized Bayesian Filtering

In the searching problem, the unknown variable of interest is the target state vector at time step k, denoted $\mathbf{x}_k^t \in \mathbb{R}^{n_x}$, which in general describes the target location but could also include its attitude, velocity, and other properties. In this paper the superscripts t and s_i indicate a relationship to the target and the sensor i respectively. The subscripts are used to indicate the time index. The purpose of the analysis is to find an estimate for $p(\mathbf{x}_k^t|\mathbf{z}_{1:k})$, the PDF over $\mathbf{x}_k^t$ given the sequence $\mathbf{z}_{1:k} = \{\mathbf{z}_j^i : i = 1, ..., N_s, j = 1, ..., k\}$ of all the observations made from the N_s sensors on board the search vehicles, $\mathbf{z}_j^i$ being the observation from the i^{th} sensor at time step j. The analysis starts by determining a prior PDF $p(\mathbf{x}_0^t|\mathbf{z}_0) \equiv p(\mathbf{x}_0^t)$ for the target state at time 0, given all available prior information including past experience and domain knowledge. If nothing is known other than initial bounds on the target state vector, then a least informative uniform PDF is used as the prior. Once the prior distribution has been established, the PDF at time step k, $p(\mathbf{x}_k^t|\mathbf{z}_{1:k})$, can be constructed recursively using the prediction and update equations alternatively.

Prediction Suppose the system is at time step $k-1$ and the latest PDF update, $p(\mathbf{x}_{k-1}^t|\mathbf{z}_{1:k-1})$, is available. Then the predicted PDF of the target state at time step k is obtained from the following Chapman-Kolmogorov equation

$$p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k^t|\mathbf{x}_{k-1}^t)p(\mathbf{x}_{k-1}^t|\mathbf{z}_{1:k-1})\, d\mathbf{x}_{k-1}^t \tag{1}$$

where $p(\mathbf{x}_k^t|\mathbf{x}_{k-1}^t)$ is a probabilistic Markov motion or process model which maps the probability of transition from a given previous state $\mathbf{x}_{k-1}^t$ to a destination state $\mathbf{x}_k^t$ at time k. The process model is a function of the equations of motion for the target and of the known distribution on their inputs, as well as the environmental constraints impeding the target motion, e.g walls and partitions. Various examples of process models with constraints can be found in [2].

Update At time step k, a new set of observations $\mathbf{z}_k = \{\mathbf{z}_k^1, ..., \mathbf{z}_k^{N_s}\}$ becomes available. For each sensor i, the mapping of the target state observation probability, $\mathbf{z}^i \in \mathbb{R}^{n_z}$, for each given target state, $\mathbf{x}_k^t \in \mathbb{R}^{n_x}$, is denoted $p(\mathbf{z}_k^i|\mathbf{x}_k^t)$ and will be referred to as the observation likelihood, or sensor model. Assuming all the observations to be conditionally independent, the PDF update from the prediction stage (1), $p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1})$, is performed using the following Bayes rule with the normalization coefficient K, also referred to in the literature as the "independent opinion pool"

$$p(\mathbf{x}_k^t|\mathbf{z}_{1:k}) = K\, p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1}) \prod_{i=1}^{N_s} p(\mathbf{z}_k^i|\mathbf{x}_k^t) \tag{2}$$

2.2 Decentralized Coordinated Control

Each of the N_s sensor/vehicle system is governed by its own dynamic model in the form

$$\mathbf{x}_{k+1}^{s_i} = \mathbf{f}_k^{s_i}(\mathbf{x}_k^{s_i}, \mathbf{u}_k^{s_i}, \mathbf{w}_k^{s_i}) \tag{3}$$

where $\mathbf{w}_k^{s_i}$ is the vector representing the process noise and the external forces acting on the system i, and where $\mathbf{u}_k^{s_i}$ is the corresponding control input vector at time k. The controller objective is to produce a command that will place the system in a desired state.

For multiple sensor platforms, an optimal cooperative control solution must be the group decision that is jointly optimal. In decentralized systems, optimal plans are usually intractable and can only be obtained through extensive negotiations between a small number of agents. A solution is to follow a decentralized coordinated control strategy which consists in implementing an independent control law for each sensor. The individual utility function for the i^{th} sensor is denoted $J_k^{s_i}(\mathbf{u}^{s_i}, N_k)$, where $\mathbf{u}^{s_i} = \{\mathbf{u}_1^{s_i}, ...\mathbf{u}_{N_k}^{s_i}\}$ is the control action sequence over a time horizon of length $T = N_k\,\delta t$, where N_k is the number of lookahead steps. The locally optimal control policy $\mathbf{u}^{s_i*}$ is the sequence that maximizes that utility subject to the control bounds $\mathbf{u}_{LB}^{s_i} \leq \mathbf{u}^{s_i} \leq \mathbf{u}_{UB}^{s_i}$ and the constraints $\mathbf{g}^{s_i}(\mathbf{u}^{s_i}, N_k) \leq \mathbf{0}$.

$$\mathbf{u}^{s_i*} = \{\mathbf{u}_1^{s_i*}, ..., \mathbf{u}_{N_k}^{s_i*}\} = \arg\max_{\mathbf{u}^{s_i}} \; J_k^{s_i}(\mathbf{u}^{s_i}, N_k) \tag{4}$$

The simplest form of coordinated control is implemented with a lookahead depth of one-step corresponding to maximizing N_s independent control laws in the form $J_k^{s_i}(\mathbf{u}_k^{s_i}, 1)$. There is no mechanism to reach a negotiated outcome. Each sensor node exchange information via the network ensuring that each platform builds an equivalent representation of the world, i.e. target state PDF. Coordination results from the platforms affecting each other's local control decisions by contributing to the prior on which these local decisions are made [3,9]. Coordinated trajectories are suboptimal, but they have the appealing advantages of being completely decentralized, computationally very cheap and highly scalable as the nodal planning computation costs do not increase with the number of platforms.

2.3 Active Bayesian Sensor Network

Packaging a physical sensor with its own Bayesian filtering processor is an attractive way of making the sensor mobile and modular. Fig. 1 depicts algorithmically the nodal Bayesian filter and how it interacts with the Controller, the Platform, and the Sensor in a network with node-to-node communication. The Platform block represents the actuators and dynamics of both the sensor and the mobile vehicle, if

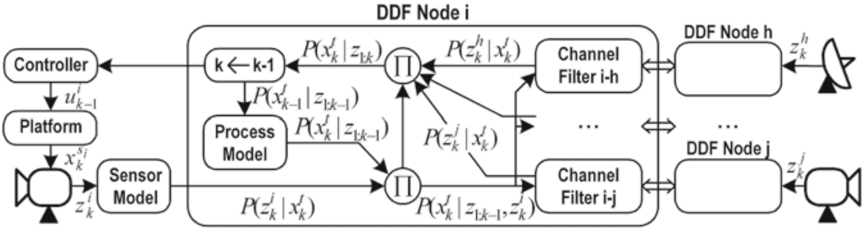


Fig. 1. General active Bayesian sensor node with channel filters for node-to-node communication.

present, on which the sensor is mounted. Based on the latest belief about the world $p(\mathbf{x}_{k-1}^t|\mathbf{z}_{1:k-1})$ and the sensor state $\mathbf{x}_{k-1}^{s_i}$, the Controller sends a command $\mathbf{u}_{k-1}^{s_i}$ to the Platform to place the sensor in a desired position $\mathbf{x}_{k_{des}}^{s_i}$ with respect to the world to take the next observation. When it comes in, the new observation likelihood $p(\mathbf{z}_k^i|\mathbf{x}_k^t)$ is fused with the predicted estimate $p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1})$ to form the new nodal estimate based on an incomplete set of observations $p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1}, \mathbf{z}_k^i) = p(\mathbf{x}_k^t|\mathbf{z}_{1:k}^{*i})$. This latest estimate is then sent to a neighboring node via a Channel Filter whose purpose is to maintain a density estimate based on the common information shared between the two nodes. In order to prevent double-counting, the Channel Filter uses its estimate to remove the common information from the nodal estimate. The residual which corresponds to the new information accumulated by the emitting node, through sensor observations and communication with other neighbors, is then communicated to the Channel Filter of the receiving node to update both the nodal and channel estimates. Likewise, the emitting node also fuses to its estimate prior to communication, $p(\mathbf{x}_k^t|\mathbf{z}_{1:k}^{*i})$, the information that it receives from its neighbors.

If the node is the hub of the network, its new post-communication estimate corresponds to the global estimate based on complete information, $p(\mathbf{x}_k^t|\mathbf{z}_{1:k})$. In general, the nodal density estimate at any given time is based on an incomplete set of observations caused by the time delay necessary for new information to propagate through the network. The discrepancies between the nodal estimates increase with the number of prediction steps and/or observations made between each communication step, as well as the length of the communication chains in the network. However, the channel filter guarantees the nodes to converge to the global estimate given the network propagation delay. The details of the general channel filter are introduced in [1] where it is also proposed to evaluate the amount of estimation error using the following Hellinger affinity measure

$$D(p_i\|p_j) = 2\ln \int \sqrt{p_i(\mathbf{x}_k^t)p_j(\mathbf{x}_k^t)}\, d\mathbf{x}_k^t \tag{5}$$

which is a monotonic distance metric between two densities p_i and p_j [10]. The metric values range from 0, when the densitites are identical, to $-\infty$, when they have nothing in common, i.e. $\int \sqrt{p_i(\mathbf{x}_k^t)p_j(\mathbf{x}_k^t)}\, d\mathbf{x}_k^t = 0$. The channel manager also uses this divergence measure to determine when to communicate on each channel [1]. One necessary condition to maintain proper accounting of the information is that the network connectivity must be acyclic [8]. In other words, no communication loops must exist between the nodes that would enable the information to cycle through multiple times.

3 The Search Problem

This section describes the equations for computing the probability of detection of a lost object referred to as the target by using the outputs of the prediction and update equations from Sec. 2.1. An equivalent but different derivation is presented in [3]. Further details on the searching problem can also be found in [13] and [12] (Chap.9).

Let the target detection likelihood (observation model) of the i^{th} sensor at time step k be given by $p(\mathbf{z}_k^i = D_k^i|\mathbf{x}_k^t)$ where D_k^i represents a ‘detection’ event by sensor i at time k. The likelihood of ‘no detection’ by the same sensor is given by its complement $p(\overline{D}_k^i|\mathbf{x}_k^t) = 1 - p(D_k^i|\mathbf{x}_k^t)$. The combined 'no detection' likelihood for all the sensors at time step k is simply a multiplication of the individual 'no detection' likelihoods

$$p(\overline{D}_k|\mathbf{x}_k^t) = \prod_{i=1}^{N_s} p(\overline{D}_k^i|\mathbf{x}_k^t) \tag{6}$$

where $\overline{D}_k = \overline{D}_k^1 \cap ... \cap \overline{D}_k^{N_s}$ represents the event of a 'no detection' observation by every sensor at time step k. Neglecting the normalization factor K in the update equation (2) gives

$$p(\mathbf{x}_k^t|\mathbf{z}_{1:k})' = p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1})' \prod_{i=1}^{N_s} p(\mathbf{z}_k^i|\mathbf{x}_k^t) \tag{7}$$

The advantage of not normalizing the target PDF at every update is that the joint probability of failing to detect the target in all of the steps from 1 to k, denoted $Q_k = p(\overline{D}_{1:k})$, can be directly obtained from the integration of the pseudo PDF update (7)

$$Q_k = \int p(\mathbf{x}_k^t|\overline{D}_{1:k})' d\mathbf{x}_k^t = \int p(\mathbf{x}_k^t|\overline{D}_{1:k-1})' p(\overline{D}_k|\mathbf{x}_k^t)\, d\mathbf{x}_k^t \tag{8}$$

where $\overline{D}_{1:k}$ corresponds to the set of observations $\mathbf{z}_{1:k}$ where every observation is a 'no detection', i.e. $\mathbf{z}_k = \overline{D}_k, \forall k$. Then, it can be shown that the probability the target gets detected for the first time on time step k, denoted p_k, is given by the volume under the surface resulting from the product of the combined detection likelihood, denoted $\left[1 - p(\overline{D}_k|\mathbf{x}_k^t)\right] = p(D_k|\mathbf{x}_k^t)$, with the predicted target PDF. This is equivalent to the reduction in volume ($-\Delta Q_k$) of the pseudo PDF as in

$$p_k = \int p(\mathbf{x}_k^t|\overline{D}_{1:k-1})' \left[1 - p(\overline{D}_k|\mathbf{x}_k^t)\right] d\mathbf{x}_k^t = Q_{k-1} - Q_k \tag{9}$$

Assuming no false detection from the sensors, the probability that the target *has* been detected in k steps, denoted P_k, is obtained from the cumulative sum of the p_k's as in

$$P_k = \sum_{i=1}^{k} p_i = P_{k-1} + p_k \tag{10}$$

For this reason P_k will be referred to as the 'cumulative' probability of detection to distinguish it from the payoff probability of detection function p_k. Notice that plugging the expressions for p_k from (9) into (10) gives $P_k = 1 - Q_k$ since $Q_0 = \int p(\mathbf{x}_0^t) d\mathbf{x}_0^t = 1$. This signifies that if the target PDF is not normalized after each update as in (7), then its volume, Q_k, represents the residual probability that the target is still present despite the search effort expended. Also, as k goes to infinity, Q_k decreases towards zero and P_k levels off towards one as it becomes harder to generate additional observation payoff, p_k, from hardly any probability mass left in the PDF.

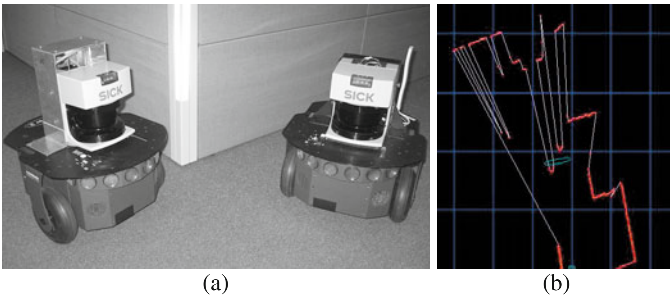
The goal of a searching strategy could be to maximize the chances of finding the target given a restricted amount of time by maximizing P_k over a given time horizon [4]. For a time horizon of one as discussed in Sec. 2.2, the individual utility function reduces to the probability of detecting the target on the next time step, as in

$$J_k^{s_i}(\mathbf{u}_k^{s_i}, 1) = p_k^{s_i} = \int p(\mathbf{x}_k^t|\overline{D}_{1:k-1})' p(D_k^i|\mathbf{x}_k^t)\, d\mathbf{x}_k^t \tag{11}$$

which corresponds to the volume under the surface resulting from the product of the 'detection' likelihood from sensor i with the predicted pseudo target PDF.

4 Experiment

The goal of the ongoing research effort is to demonstrate the effectiveness of the coordinated decentralized search framework on a team of heterogeneous autonomous mobile platforms in real outdoor scenarios. A stepping stone towards this goal is to investigate the problem using simple physical robots in an indoor environment. The remainder of this section presents the implementation details and the results of the decentralized coordinated search framework implemented for a team of three Pioneer 2 robots, each equipped with a laser scanner (Fig. 2a), searching for a single mobile human target throughout the office space of the Australian Centre for Field Robotics (ACFR).



(a) (b)

Fig. 2. Experiment: (a) two of the three Pioneer robots equipped with laser range finders and wireless communication; (b) raw laser scan in which two legs can be distinguished and where the cyan ellipse represents the estimate of the human target location.

4.1 Implementation

The necessary decentralized network components, i.e. controller, platform, sensor and fusion node, were derived from basic C++ classes developed at ACFR as part of the ASN research thrust [11]. The Pioneer platforms are controlled via a publicly available hardware abstraction layer called Player[1]. Prior to the search experiment, the indoor environment (Fig. 4a) was explored to produce an occupancy grid (OG) map of the area (appox. 16x14m), as well as a localization feature map (Fig. 4b) using the SLAM algorithm [6] . The features are stripes of reflective material, as seen on Fig. 2a, returning high laser intensities. Once the SLAM map was judged to be accurate enough, it was stored and subsequently used as a beacon map for localization during the search experiment. The human target motion is modelled *a priori* as a zero mean Gaussian diffusion process, i.e. random walk, with a standard deviation in x and y of 0.35 m/s. The OG map is then used to constrain the resulting tabulated probability of transition $p(\mathbf{x}_k^t|\mathbf{x}_{k-1}^t)$ as in [2]. As for the prior PDF in an indoor environment (Fig. 4b), it could come straight from the estimate of a surveillance system after it lost track of its target. Another possibility could be to use data accumulated over an extended period to produce a prior reflecting the target's preferred location in the environment.

[1] Player was developed jointly at USC Robotics Research Lab. and HRL Labs. and is available under the GNU General Public License from http://playerstage.sourceforge.net

Platform Model Each vehicle i is moving in the xy plane at constant velocity $V_i = .15$ m/s where the single control parameter $u_k^{s_i}$ is the heading rate and is maintained over the time interval δt. In this experiment the maximum heading rate amplitude is set to $u_{max} = \pm 20$ deg/s). The vehicle pose prediction model used for the planning purposes is the following discrete time non-linear constant velocity model

$$x_{k+1}^{s_i} = x_k^{s_i} + \frac{2V_i}{u_k^{s_i}} \sin(\frac{1}{2} u_k^{s_i}\, \delta t) \cos(\theta_k^{s_i} + \frac{1}{2} u_k^{s_i}\, \delta t) \tag{12}$$

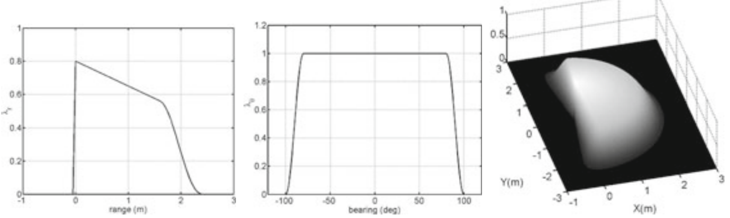
$$y_{k+1}^{s_i} = y_k^{s_i} + \frac{2V_i}{u_k^{s_i}} \sin(\frac{1}{2} u_k^{s_i}\, \delta t) \sin(\theta_k^{s_i} + \frac{1}{2} u_k^{s_i}\, \delta t) \tag{13}$$

$$\theta_{k+1}^{s_i} = \theta_k^{s_i} + u_k^{s_i}\, \delta t \tag{14}$$

Observation Model The search sensor is the same laser range finder that is used for navigation. Mounted on top of the Pioneer, its horizontal sensing plane is about mid-distance between the knees and ankles of an adult. This makes it quite an interesting sensor for this experiment as a pair of legs at this height represents a fairly distinct feature that is easy to discriminate [5]. A leg, as sensed by the laser scanner, is modelled as a semi-circle with radius in a certain bounded range, separated from its background by a threshold distance determined experimentally. A human is defined as a pair of legs within a certain distance of each other. Fig. 2b shows the detection of a pair of likely legs in a laser scan.

Fig. 3b illustrates the detection likelihood with respect to the target distance from the sensor. Notice that to make the experiment less trivial, the detection range was artificially reduce from 10 m to 2 m in order to make the limited search area appears relatively larger. As seen on Fig. 3a, the laser scan goes from -90 to +90 degrees with respect to the robot orientation and the resulting detection likelihood, $p(D_k^i|\mathbf{x}_k^t)$, is illustrated on Fig. 3c. If no obstacles are in the sensor range, this is the detection likelihood that enters the utility function (11) to determine the next control action. When a new scan comes in, it is first process to detect the presence of a human. If no contact is made, then the nodal estimate is updated with the ‘no detection’ likelihood given by $p(\overline{D}_k^i|\mathbf{x}_k^t) = 1 - p(D_k^i|\mathbf{x}_k^t)$.

On a practical note, to ensure proper convergence of the control optimization algorithm, it is important that the detection likelihood be smooth and that it decreases progressively to zero with increasing range and/or bearing without any steps in the function as seen on Fig. 3.



(a) (b) (c)

Fig. 3. Observation Model: (a) detection likelihood vs. range; (b) detection likelihood vs. bearing; (c) 3D view of the detection likelihood over range and bearing, $p(D_k^i|\mathbf{x}_k^t)$, for $\mathbf{x}_k^{s_i} = [0, 0, 0]$.

4.2 Results

Fig. 4 illustrates the decentralized search results for the active Bayesian sensor network algorithm presented in Sec. 2. The robots, namely Hornet, Mozzy and Dragonfly are travelling at constant velocity except when slowing down near obstacles. A modified version of the vector field histogram (VFH) technique [14] ensures that collisions are avoided. One observation and one control decision is made at approximately one second intervals on board each robot. The channel manager communication threshold is set to $D_{thresh} = -.01$. Figs. 4c to f represent snapshots of the evolution of the PDF estimate computed onboard Hornet's node, which is the hub of the network, and the coordinated robot trajectories given the prior density shown on Fig. 4b. Fig. 4g compares the nodal cumulative probability of detection functions $P_k^{s_i}$'s. Fig. 4h is a zoom in of the later and Fig. 4i displays the corresponding divergence evolutions representing the discrepancies between the nodal estimates and the channel estimates over time.

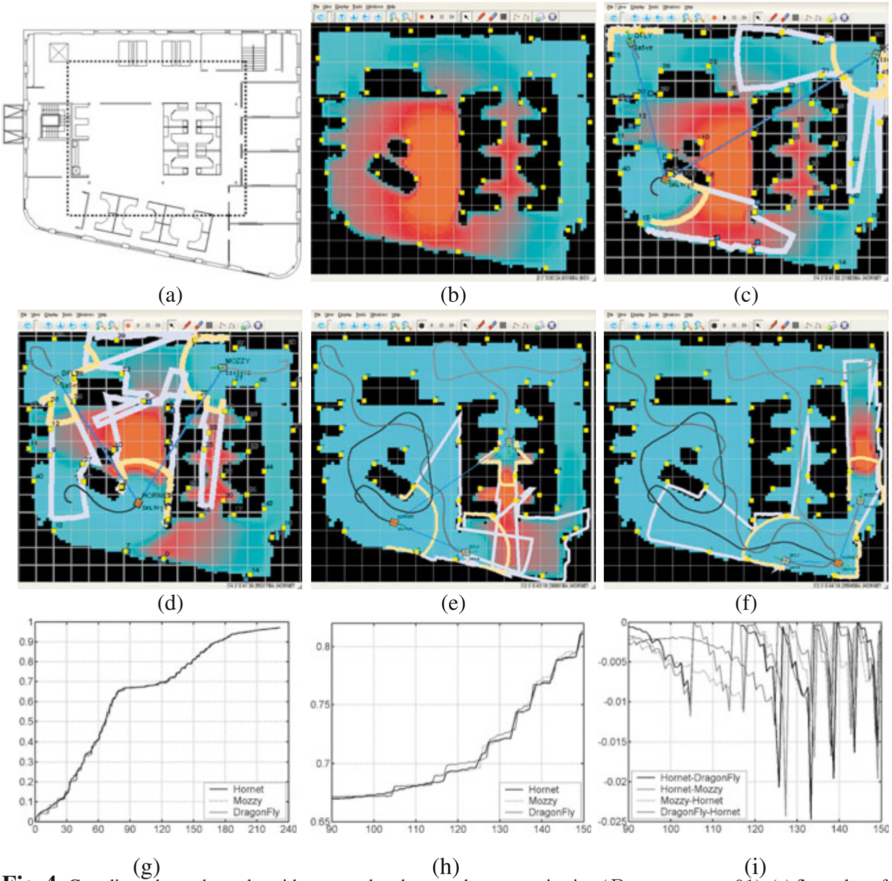


Fig. 4. Coordinated search results with managed node-to-node communication ($D_{thresh} = -.01$): (a) floor plan of ACFR with the 16x14m search area delimited by the dashed line; (b) OG map (black obstacles) and beacon map (small yellow squares) overlaid with the prior target PDF (red equals high density); (c)-(f) snapshots of the target PDF estimate onboard Hornet, the coordinated robot trajectories at 12, 40, 148 and 208 seconds into the search respectively and the network links between the robots (blue lines); (g) cumulative probability of detection P_k vs. time [s]; (h) zoom in of P_k, and (i) the corresponding Hellinger divergence evaluated between the node and channel estimates.

The nodal channel manager uses the divergence measure to adaptively determine when to communicate on any particular channel. The smaller the communication threshold, the more often the nodes communicate and the more accurate their estimates are. This is important as the search platforms are then much less likely to interfere with each other when their density estimates are almost identical. The communication divergence threshold is therefore a compromise between accuracy and bandwidth capacity. Clearly seen on the divergence plot is the synchronization between the nodal and channel estimates occurring at each communication burst. The communication steps can also be easily identified from the steps in the $P_k^{s_i}$ estimates. Also worth noticing on the zoom in of the P_k plot is that once a node communicates, it is usually followed soon after by the receiving nodes which transmit to their other neighbors and so forth until the new information that triggered the chain reaction is propagated throughout the network.

Finally, the efficiency of the coordinated control strategy in allocating the search effort is demonstrated by a final cumulative probability of detection as high as 97.1% for 231s of search. Recall that this performance is reached without the robots exchanging any information about their plans. As mentioned in Sec. 2.2, coordination results from the platforms affecting each other's control decisions by contributing the prior on which these local decisions are made. This is explained by the fact that the utility for a robot to search a region is decreased if another agent is already searching that region. This has the effect of increasing the relative utility of other regions of the space and diverting the former robots towards these regions. In some occasions however, because of the myopic nature of this kind of planning, the robots can fail to detect higher utility regions outside their predicted sensor range and might get stuck circling inside a room instead of getting out and visiting another room with higher payoff values. For these cases, it would be beneficial to implement the control with a time-horizon of adaptive length.

5 Conclusion and Further Work

This paper addressed the implementation issues related to the problem of coordinating multiple, possibly heterogeneous, sensing platforms performing a search mission for a single target in a dynamic environment. The general decentralized Bayesian framework presented constitutes a building block of the much broader problem of management in decentralized systems. It was demonstrated to adaptively find efficient coordinated search plans that explicitly considers the search vehicles kinematics, the sensors detection function, as well as the target arbitrary motion model. Decentralized coordinated solutions, while being suboptimal, are adaptive and offer tremendous scalability potential as their nodal computational costs are kept constant with the number of platforms. The general decentralized Bayesian filtering algorithm was demonstrated to be robust to communication failures and ensured that complete global information was recovered at each node despite delayed communications in the network.

As part of the ongoing research effort, techniques to facilitate human interactions with the active sensor network are being investigated to enable an operator to enter observations in the network and influence the agents control decisions. A novel approach to cooperative control based on a decentralized negotiation filter that will increase the time horizon of the decentralized search plans is also being developed.

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, and by AFOSR/AOARD under contract 03-13.

References

1. F. Bourgault and H.F. Durrant-Whyte. Communication in general decentralized filters and the coordinated search strategy. In *The 7th Int. Conf. on Information Fusion*, Stockholm, Sweden, June 2004.
2. F. Bourgault and H.F. Durrant-Whyte. Process model, constraints, and the coordinated search strategy. In *IEEE Int. Conf. on Robotics and Automation (ICRA'04)*, New Orleans, USA, April 2004.
3. F. Bourgault, T. Furukawa, and H.F. Durrant-Whyte. Coordinated decentralized search for a lost target in a Bayesian world. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'03)*, October 2003.
4. F. Bourgault, T. Furukawa, and H.F. Durrant-Whyte. Optimal search for a lost target in a Bayesian world. In *Int. Conf. on Field and Service Robotics (FSR'03)*, Japan, July 2003.
5. A. Brooks and S. Williams. Tracking people with networks of heterogeneous sensors. In *Australasian Conf. on Robotics and Automation (ACRA'03)*, Brisbane, Australia, December 2003.
6. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csobra. A solution to the simultaneous localization and map building (SLAM) problem. *Robotics and Automation*, 17(3):229–241, 2001.
7. M. Flint, M. Polycarpou, and E. Fernandez-Gaucherand. Cooperative control for multiple autonomous uav's searching for targets. In *41st IEEE Conference on Decision and Control*, Las Vegas, USA, December 2002.
8. S. Grime and H.F. Durrant-Whyte. Communication in decentralized systems. *IFAC Control Engineering Practice*, 2(5):849–863, 1994.
9. B. Grocholsky, A. Makarenko, and H. Durrant-Whyte. Information-theoretic coordinated control of multiple sensor platforms. In *IEEE Int. Conf. on Robotics and Automation (ICRA'03)*, 2003.
10. A.O. Hero, B. Ma, O. Michel, and J. Gorman. Alpha-Divergence for Classification, Indexing and Retreival (Revised 2). Technical Report CSPL-328, Communication and Signal Processing Lab., The University of Michigan, 48109-2120, USA, June 2002.
11. A. Makarenko, A. Brooks, S. Williams, H.F. Durrant-Whyte, and B. Grocholsky. A decentralized architecture for active sensor networks. In *IEEE Int. Conf. on Robotics and Automation (ICRA'04)*, New Orleans, USA, April 2004.
12. J.S. Przemieniecki. *Mathematical Methods in Defense Analyses*. AIAA Education Series. American Inst. of Aeronautics and Astronautics, Inc., Washington, DC, 2nd edition, 1994.
13. L.D. Stone. *Theory of Optimal Search*, volume 118 of *Mathematics in Science and Engineering*. Academic Press, New York, 1975.
14. I. Ulrich and J. Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA '98)*, pages 1572–1577, 1998.
15. R. Vidal, S. Rashid, C. Sharp, O. Shakernia, J. Kim, and S.S. Sastry. Pursuit-evasion games with unmanned ground and aerial vehicles. In *IEEE Int. Conf. on Robotics and Automation (ICRA'01)*, Seoul, Korea, May 2001.

Synthesis and Analysis of Non-Reactive Controllers for Multi-Robot Sequential Task Domains

Chris Jones and Maja J Matarić

Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781, USA
{cvjones|maja}@robotics.usc.edu

Abstract. In this paper we present a macroscopic model for the analysis of homogeneous task-directed multi-robot systems (MRS). The model is used to compute the probability that a given MRS will correctly execute a given sequential Markovian task. We consider distributed MRS composed of non-communicative robots that maintain a limited amount of non-transient internal state. The model shares a common formal framework with our past work on the development of systematic methods for the synthesis of MRS. As such, it can be used to improve design procedures by providing an analytical approach to the evaluation of design decisions. The unified nature of the modeling and synthesis methods are part of our ongoing work toward a general, comprehensive and principled MRS design methodology. We apply the model to the analysis of system performance in a multi-robot construction task domain. Our past work on synthesis methods provides robot controllers for this task domain; the model provides quantitative predictions of system performance. We discuss the assumptions and limitations inherent in the model and discuss how the complementary synthesis and analysis methods may be more fully integrated.

1 Introduction

A critical undertaking in the design of task-directed multi-robot systems (MRS) is managing the complexity introduced by the coordination of multiple, interacting robots. In order for such systems to be effective, the robots' actions must be carried out in a coordinated fashion and directed toward task achievement. The system overall effectiveness typically strongly depends on the underlying coordination mechanism(s) used to mediate the interactions among the robots and between the robots and the task environment. Such coordination mechanisms, in turn, typically depend on the use of inter-robot communication and the maintenance of some form of internal state or representation by each of the robots.

Our work is focused on the design and analysis of homogeneous distributed MRS, in which each robot operates independently, under local sensing and control. The design of such systems, especially at a large-scale, is challenging and expensive. Unexpected collective behaviors may emerge due to unanticipated ramifications of the robots' local interactions, often resulting from uncertainty in sensing and action. There exist few principled tools to aid in the design process; resource-intensive design through trial and error is frequently the only available option.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 417–426, 2006.

To address the lack of principled tools for MRS design, our previous work presented a formal framework and an accompanying suite of principled, task-*non*specific methods for the synthesis of coordinated MRS. Each method represents a systematic approach using a variety of known coordination tools, including inter-robot communication [6] and the maintenance of internal state [7]. In the case of the latter, one is confronted with many options as to how the robots should utilize internal state to achieve the desired system-level coordination. System performance can frequently be improved by using more internal state. However, there is a point in the design space at which the cost of additional internal state outweighs improvements in performance. Analogously, a stateless system can usually be used if resulting task performance or efficiency are not a critical design considerations. Intelligently evaluating such trade-offs through direct experimentation is inefficient and costly; our work is aimed toward a formal framework that can address these issues.

In this paper, we present a macroscopic model for calculating the probability that a given homogeneous task-directed MRS, comprised of non-communicative robots that maintain a limited amount of non-transient internal state, will correctly execute a given sequential Markovian task. This model can be used to improve the design process by aiding in decisions related to the appropriate use of internal state and its trade-offs with resulting system task performance. The model shares a common formal framework with our synthesis methods. The unified nature of the synthesis methods and the modeling technique are part of our ongoing work to develop a complete and systematic MRS design methodology. The use of the unified synthesis and analysis, in conjunction with indispensable real-world experiments, should allow for a more systematic, efficient, and complete exploration of the design space, thereby leading to more effective design.

We apply the model to the analysis of coordinated MRS in a multi-robot construction task domain. Our controller synthesis method, presented in [7], provides robot controllers for this domain. Analysis of the synthesized MRS is then performed using the analytical model presented in this paper. The model is shown to provide quantitatively correct predictions of system performance. Extensive experiments were conducted in physically-realistic simulation to verify the correctness of the model's predictions. A limited number of real robot experiments were also performed to validate the simulation results [7].

2 Related Work

Work related to the analysis of coordinated MRS includes the work of Donald [2], who presented the derivation of information invariants aimed at defining the information requirements of a given task and ways in which those requirements can be satisfied in a robot controller. Donald et al. [1] applied information invariants to the study of distributed manipulation. Parker [13] extended the idea of information invariants by defining equivalence classes among task definitions and robot capabilities to assist in the choice of appropriate controller class. Sugawara and Sano [14] studied a macroscopic modeling technique applied to a multi-robot foraging task.

Martinoli et al. [12] presented a macroscopic probabilistic modeling methodology for the study of collective robot behavior. Martinoli and Easton [11] discussed the effectiveness and accuracy of microscopic and macroscopic modeling techniques compared to real robot experiments and embodied simulations. Gerkey and Matarić [3] presented a principled framework and analysis methodology, based on theories from economics and operations research, toward the formal study of multi-robot task allocation. Lerman et al. [10] presented a macroscopic analytical model of the dynamics of collective behavior in a collaborative stick-pulling domain using a series of coupled differential equations. Lerman and Galstyan [9] described a general macroscopic model for the study of adaptation in multi-agent systems and applied it to analysis of a multi-robot adaptive task allocation domain that our own work had previously addressed experimentally [5].

3 Definitions and Notation

The *world* is the domain in which the MRS is expected to perform a defined task. We assume the world is Markovian and the state is an element of the finite set S of all possible states. The set of all robots is denoted by the finite set R. We assume the robots are homogeneous. An action a_r performed in the world by a single robot r is drawn from the finite set A of all possible actions. An *observation* x made by robot r, drawn from the finite set of all observations X, consists of accessible information external to the robot and formally represents a subset of the world state. The world is defined by a probabilistic state transition function $P : S \times X \times A \times S \rightarrow [0, 1]$. That is, given a world state s at time t, a robot r making observation x and executing action a, and a world state s' at time $t + 1$, $P(s, x, a, s') = Pr(S^{t+1} = s'|S^t = s, X_r^t = x, A_r^t = a)$. We note that the world state transition function involves an observation because the tasks we consider are spatial in nature and the physical location where an action is performed is just as important as the action itself. In this representation, an observation x is equated with the spatial location where the action a is performed. Therefore, an action a executed upon the observation of x_i will transition the world differently than the same action a performed upon the observation of x_j. We define a *task*, assumed to be sequential and Markovian, as a set of n ordered world states $T_s = \{s_0, s_1, ..., s_n\}$ which must be progressed through in sequence. We assume the initial state of the world is s_0 and state s_n is terminal. We define *correct task execution* to be the case where, for all task states $s_i \in T_s$, $i < n$ the only actions executed by any robot are those that transition the world state to s_{i+1}. Once the world state is $s_n \in T_s$ the task is terminated. Therefore, we define an observation and action pair for a robot, x and a, to be correct for task state s_i if $P(s_i, x, a, s_{i+1}) > 0$. We assume that an observation x and action a cannot be correct for more than one task state. The probabilistic *observation function* $O(s, x) = Pr(X_r^t = x|S^t = s)$ gives the probability observation x will be made in state s by a robot r. For each state s, the observation function is normalized as follows: $\sum_{\forall x} O(s, x) = 1$. A robot's internal state value m at any time is a member of the finite set $M = \{m_0, ..., m_p\}$, with m_0 being the initial internal state value for all robots. Two probabilistic functions define a robot r's behavior in the world, known collectively as the robot's *controller*. The controller is comprised of an *action function* $A(x, m, a) = Pr(A_r^t = a | X_r^t = x, M_r^t = m)$ and an *internal state transition function* $L(m, x, m') = Pr(M_r^{t+1} = m' | X_r^t = x, M_r^t = m)$. We prohibit the internal state transition function from containing cycles. Although the controller is modeled with probabilistic functions for generality, in this paper these functions are assumed to be binary – A and L will always be either 0 or 1.

4 Macroscopic Model

Using the above formal framework, we next present a macroscopic analytical model for calculating the probability that a given MRS will correctly execute a particular sequential Markovian task. The model is specifically intended for a homogeneous MRS composed of non-communicative robots whose actions are based on their current local observation and internal state. A prime difficulty in constructing such a controller is in defining how available internal state values should be assigned and transitioned during the course of task execution so as to maximize resulting system performance.

During task execution, errors in sensing can inadvertently cause inappropriate internal state transitions and, consequently, the execution of incorrect actions. Therefore, the accuracy of our model relies on an understanding of how, given a controller, the robots' internal state values change during task execution. To address this issue, Equation 1 presents a method to calculate the probability a robot's internal state value will be m_j in task state s_i. This allows for the recursive calculation of the probability distribution of robot internal state values over task states. We note that $Pr(m_j | s_i) = 0$, if $j < 0$, $j >= |M|$, $i < 0$, or $i >= |T_s|$. Similarly, $L(m_j, x, m_k) = 0$, if $j < 0$.

$$\begin{aligned} Pr(m_j | s_i) = 1 - ((1 - Pr(m_{j-1}, s_i)(1 - \prod_{\forall x}(1 - O(s_i, x) \\ \cdot L(m_{j-1}, x, m_j))))(1 - Pr(m_j | s_{i-1})(1 - Pr(m_{j+1} | s_{i-1})))) \end{aligned} \tag{1}$$

Equation 2 provides a method to calculate the probability that a homogeneous MRS, composed of robots executing a given controller, will achieve correct task execution (CTE).

$$\begin{aligned} Pr(CTE) = \prod_{\forall s_i \forall m \forall x \forall a} (1 - Pr(m | s_i) O(s_i, x) A(x, m, a) \\ \cdot (1 - (P(s_i, x, a, s_{i+1}) + P(s_i, x, a, s_i)))) \end{aligned} \tag{2}$$

As with any modeling technique, the accuracy of the results depends on the accuracy and completeness of the underlying model. Constructing a model can be tedious and difficult. Fortunately, the definitions required by our model are task domain specific but are not specific to a particular task instance in that domain. Therefore, a single model of a given task domain can be used unchanged for many specific task instances within that domain.

We assume there is no implicit "structure" in the world or the movements of the robots that correlates observation or action sequences. To satisfy this assumption, or at least minimize the likelihood it is not valid, in our investigations all robots execute random walks when not engaged in task-specific actions.

We are currently considering a number of extensions to our modeling approach. The approach and the MRS synthesis methods presented in our previous work are not fully integrated. Currently, the model is used for *post hoc* analysis of a previously synthesized MRS. However, as they share a common formal foundation, we are in the process of integrating them toward a fully unified framework for MRS synthesis and analysis. The analysis method will provide online analysis during synthesis, thus enabling more effective and efficient MRS design. Also, the current model does not consider robots capable of inter-robot communication. We are now extending the model to include such capabilities, thereby enabling the modeling and analysis of MRS composed of communicative robots that maintain internal state.

5 Case Study: Coordination in Multi-Robot Construction

We apply the macroscopic model to the analysis of coordinated behavior in a multi-robot construction task domain. This task requires the sequential placement of a series of cubic, colored bricks into a planar structure. For all examples, a brick's color is denoted by the letters `R`, `G`, `B`, `Y`, `C`, and `Bl` which stand for `Red`, `Green`, `Blue`, `Yellow`, `Cyan`, and `Black`, respectively. The construction task starts with a seed structure, a small number of initially placed bricks forming the core of the structure.

Experimental data were collected through extensive physically-realistic simulation trials, using Player and Gazebo simulation environments, and through a limited number of real robot experiments. Player [4] is a server that connects robots, sensors, and control programs over a network; Gazebo [8] simulates a set of Player devices in a 3-D physically-realistic world with full dynamics. Together, the two represent a high-fidelity simulation tool for individual robots and teams that has been validated on a collection of real robot robot experiments using Player control programs transferred directly to physical Pioneer 2DXs. In all simulation experiments 8 robots were used, and in all real robot experiments 3 robots were used. The robots were either realistic models of or actual ActivMedia Pioneer 2DX mobile robots. Each robot, approximately 30 cm in diameter, is equipped with a differential drive, a forward-facing 180 degree scanning laser rangefinder, and a forward-looking color camera with a 100-degree field-of-view and a color blob detection system. The bricks are taller than the robot's sensors, so the robots can only sense the local bricks on the periphery of the structure (i.e., robots do not have a birds-eye view of the entire structure). Figure 2 shows snapshots of our simulation and real-world experimental setup.

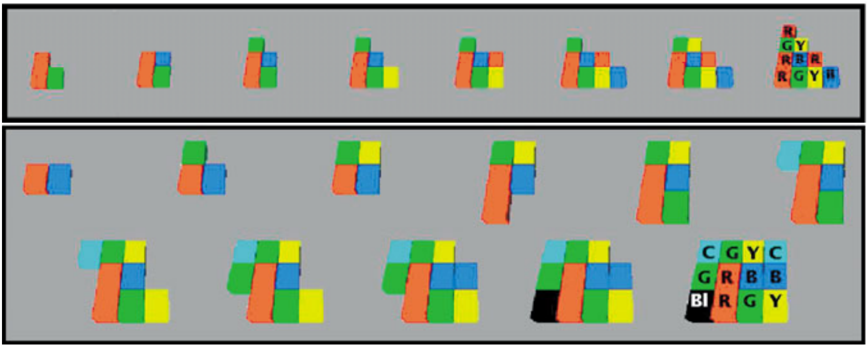


Fig. 1. The sequence of brick placements defining construction task 1 (upper) and 2 (lower).

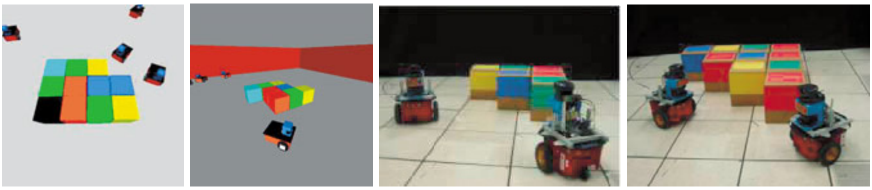


Fig. 2. Gazebo simulation (left) and real robot (right) MRS construction experiment snapshots.

Our robots do not have the ability to independently manipulate bricks during the construction process, either in simulation or in the physical world. To address this issue in simulation, when a robot wants to execute a brick placement action, it commands the simulator to place a brick of a given color at a specified location relative to the robot's current pose; this process contains action uncertainty. In real robot experiments, the experimenter manually placed the appropriate brick in response to the robot's audible command (e.g., "Place yellow brick in the corner formed by the red and blue bricks directly in front of my position"). Due to the manual brick placement, there is no associated action error in real robot experiments.

5.1 Formal Definitions for Construction Task

To ground the construction task in the formal framework presented in Section 3, we now define the world, task definitions, observations, and actions in the construction task domain. The *world* state is defined as a specific spatial configuration of bricks, including their color. A construction *task* is defined as a sequence of brick configurations (i.e., world states), providing a specific construction sequence. The two construction tasks discussed in this paper are shown in Figure 1. Observations in the construction domain are made up of the spatial configuration and color of bricks in the field-of-view of the robot's laser rangefinder and color camera and within an appropriate range and bearing. Two categories of observations are considered. The

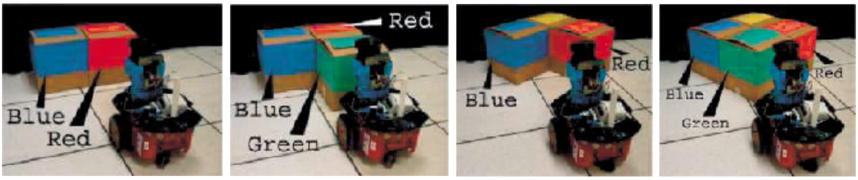


Fig. 3. Example observations and actions in the construction domain. Far left: Robot in position to make observation <FLUSH R B>. Center left: Immediately after robot performs action <G RIGHT FLUSH R B>. Center right: Robot in position to make observation <CORNER R B>. Far right: Immediately after robot performs action <G CORNER R B>.

first is two adjacent, aligned bricks. A situation in which such an observation would be made is shown in Figure 3 and is denoted as <FLUSH R B>. The second is two adjacent bricks forming a corner. A situation in which such an observation would be made is shown in Figure 3 and is denoted as <CORNER R B>. The observations <FLUSH R B> and <FLUSH B R> constitute two different observations in which the spatial relationships between the Red and Blue bricks are mirrored. A similar point holds for the observations <CORNER R B> and <CORNER B R>.

Actions are the placement of individual bricks to the growing structure. We do not consider construction tasks in which robots may remove bricks from the structure nor those in which substructures consisting of multiple bricks may be connected together. Other actions performed by the robots, such as moving through the environment, do not affect the world state and therefore do not need to be explicitly considered. Three categories of actions are considered. The first is the placement of a brick on the right side (from the perspective of the acting robot) of a pair of adjacent, aligned bricks; the immediate result is demonstrated in Figure 3 and is denoted as <G RIGHT FLUSH R B>. The second is identical to the first except that the brick is placed on the left side of a pair of adjacent, aligned bricks. This action is denoted as <G LEFT FLUSH R B>. The third is the placement of a brick in the corner formed by two other bricks; the immediate result is demonstrated in Figure 3 and is denoted as <G CORNER B R>.

5.2 Analysis Using the Macroscopic Model

We now apply the model to the analysis of task performance in the construction of the two tasks shown in Figure 1. The controllers executed by each robot for both tasks are shown in Table 1. The formal method by which they were synthesized, complete controller description and associated simulation and real robot experimental results are found in [7].

In order to build the model needed in the analysis, we ran extensive simulation trials on a baseline construction task different from the tasks used in this section. Using experimental data from these trials, we constructed the required model, including the world state transition function and the observation function, which was applied unchanged to the analysis of both construction tasks discussed in this section.

Table 1. Controller action and internal state transition functions for construction task 1 (top) and 2 (bottom) as shown in Figure 1. $m_0, m_1, ..., m_7 \in M$. The robots' initial internal state value is m_0. All probabilities not shown are 0.

Task 1 Action Function
A(<CORNER R G>, m_0, <B CORNER R G>) = 1
A(<FLUSH R B>, m_0, <G RIGHT FLUSH R B>) = 1
A(<FLUSH B G>, m_1, <Y LEFT FLUSH B G>) = 1
A(<CORNER B Y>, m_1, <R CORNER B Y>) = 1
A(<FLUSH R Y>, m_1, <B LEFT FLUSH R Y>) = 1
A(<FLUSH B R>, m_2, <Y RIGHT FLUSH B R>) = 1
A(<FLUSH G Y>, m_2, <R RIGHT FLUSH G Y>) = 1

Task 1 Internal State Transition Function
L(m_0, <CORNER G B>, m_1) = 1
L(m_1, <CORNER R B>, m_2) = 1

Task 2 Action Function
A(<FLUSH R B>, m_0, <G RIGHT FLUSH R B>) = 1
A(<CORNER G B>, m_0, <Y CORNER G B>) = 1
A(<FLUSH B R>, m_1, <R LEFT FLUSH B R>) = 1
A(<CORNER B R>, m_1, <G CORNER B R>) = 1
A(<FLUSH R G>, m_2, <C LEFT FLUSH R G>) = 1
A(<FLUSH B G>, m_3, <Y LEFT FLUSH B G>) = 1
A(<CORNER R C>, m_4, <G CORNER R C>) = 1
A(<FLUSH Y B>, m_5, <B LEFT FLUSH Y B>) = 1
A(<CORNER R G>, m_6, <Bl CORNER R G>) = 1
A(<CORNER Y B>, m_7, <C CORNER Y B>) = 1

Task 2 Internal State Transition Function
L(m_0, <FLUSH G Y>, m_1) = 1
L(m_1, <FLUSH G R>, m_2) = 1
L(m_2, <CORNER R C>, m_3) = 1
L(m_3, <CORNER B Y>, m_4) = 1
L(m_4, <FLUSH G C>, m_5) = 1
L(m_5, <CORNER Y B>, m_6) = 1
L(m_6, <FLUSH Bl G>, m_7) = 1

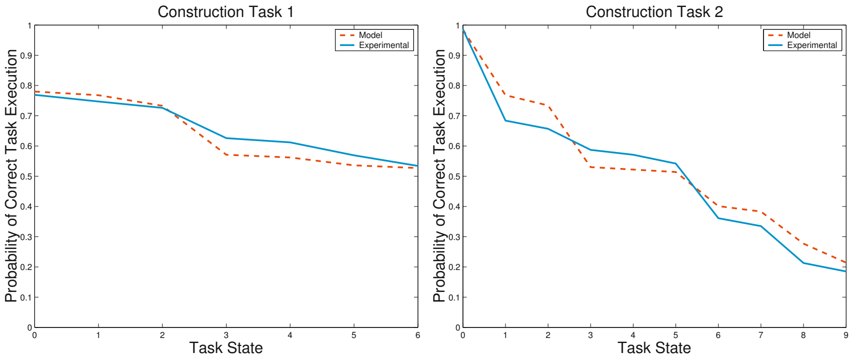


Fig. 4. Analysis for construction task 1 (left) and task 2 (right). The probability of correct task execution given for each task state is the probability the task was correctly executed up to and including the action performed in that task state.

For each of the two tasks shown in Figure 1, we ran 500 simulation experiments. Figure 4 compares the results to those provided by the model for tasks 1 and 2. As can be seen, the model provides quantitatively correct predictions of task performance. The inaccuracies seen in the model's predictions are primarily due to two factors. First, the action and sensing uncertainties observed in the experimental trials can deviate from the values used by the model. Second, the method used to calculate the distribution of internal state values over task states is in need of improvement. As is evident in Figures 5, the probability distribution of internal

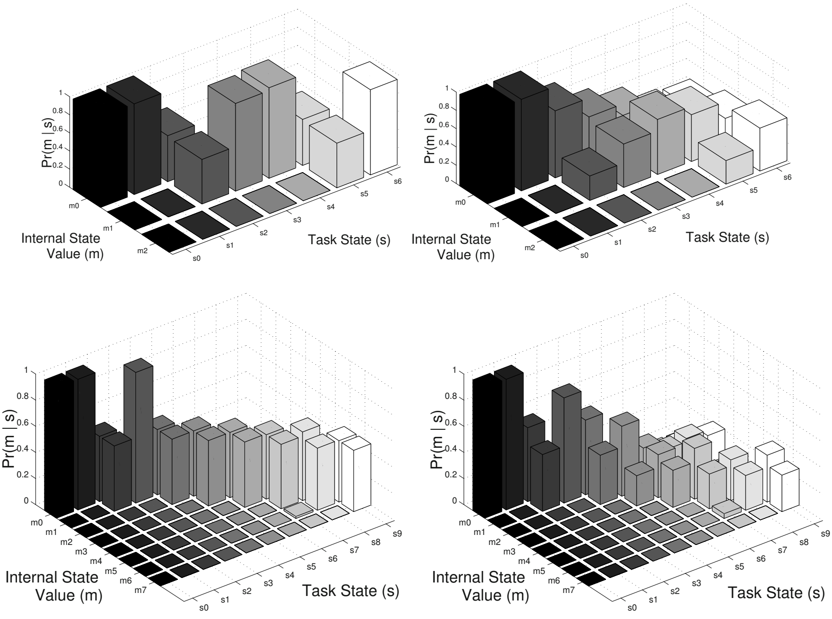


Fig. 5. Predicted (left) and actual (right) probability distribution of internal state values over task states for construction task 1 (top) and 2 (bottom).

state values predicted by the model is generally in qualitative agreement with the experiments, but quantitatively leaves much to be desired. Errors in this part of the model are most likely to manifest themselves due to perceptual aliasing in temporally proximate task states.

6 Conclusions

We have presented a macroscopic modeling approach for the predictive analysis of homogeneous MRS task performance. The model was shown to provide quantitatively correct results in predicting the probability that a MRS would correctly execute a given sequential Markovian task in a multi-robot construction task domain. Extensive experiments in physically-realistic simulations were conducted to validate the model. Real robot experiments are currently underway in the construction task domain to provide more extensive validation of the model.

7 Acknowledgments

This work was supported by Defense Advanced Research Projects Agency (DARPA) Grant F30602-00-2-0573.

References

1. B. Donald, J. Jennings, and D. Rus. Information invariants for distributed manipulation. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *International Workshop on the Algorithmic Foundations of Robotics*, pages 431–459, 1995.
2. B. R. Donald. Information invariants in robotics. *Artifical Intelligence*, 72(1–2):217–304, 1995.
3. B. Gerkey and M. Matarić. Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3862–3867, Taipei, Taiwan, Sep 2003.
4. B. Gerkey, R. Vaughan, K. Stoey, A. Howard, G. Sukhatme, and M. Matarić. Most valuable player: A robot device server for distributed control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1226–1231, Maui, Hawaii, Oct 2001.
5. C. Jones and M. Matarić. Adaptive division of labor in large-scale minimalist multi-robot systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1969–1974, Las Vegas, Nevada, Oct 2003.
6. C. Jones and M. Matarić. Automatic synthesis of communication-based coordinated multi-robot systems. Technical report, University of Southern California Center for Robotics and Embedded Systems, CRES-04-007, 2004.
7. C. Jones and M. Matarić. Toward a multi-robot coordination formalism. Technical report, University of Southern California Center for Robotics and Embedded Systems, CRES-04-008, 2004.
8. N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. Technical report, USC Center for Robotics and Embedded Systems, CRES-04-002, 2004.
9. K. Lerman and A. Galstyan. Macroscopic analysis of adaptive task allocation in robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1951–1956, Las Vegas, Nevada, Oct 2003.
10. K. Lerman, A. Galstyan, A. Martinoli, and A.J. Ijspeert. A macroscopic analytical model of collaboration in distributed robotic systems. *Artificial Life*, 7(4):375–393, 2001.
11. A. Martinoli and K. Easton. Modeling swarm robotic systems. In B. Siciliano and P. Dario, editors, *Experimental Robotics VIII*, pages 297–306. Springer-Verlag, 2002.
12. A. Martinoli, A. Ijspeert, and F. Mondada. Understanding collective aggregation mechanisms: From probabilistic modeling to experiments with real robots. *Robotics and Autonomous Systems*, 29:51–63, 1999.
13. L. Parker. Toward the automated synthesis of cooperative mobile robot teams. In *Proceedings of SPIE Mobile Robots XIII*, volume 3525, pages 82–93, Boston, MA, 1998.
14. L Sugawara and M. Sano. Cooperative acceleration of task performance: Foraging behavior of interacting multi-robot system. In *Physica D*, volume 100, pages 343–354, 1997.

XI. Walking Robots

The first paper by Kimura and Fukuoka studies the ability of a quadruped robot to achieve high-speed mobility on irregular terrain with less knowledge of the ground. This animal-inspired ability models the animal's muscle and joint with virtual spring and damper system controlled by a "neural system model" consists of a CPG (central pattern generator), reflexes and responses.

In the next paper, Lin and co-workers present odometry sensing for an autonomous hexapod. The newly developed body-pose estimator is tested on the all-terrain RHex hexapod and compared to an independent camera based ground truth measurement system (GTMS) to assess the deterioration in performance over increasingly slippery ground and at varying speeds. It is also compared to the quality of elapsed distance measurements arising from the leg sensor based estimator, a legged open loop scheme, and the traditional axle revolution count performed on a wheeled version of RHex.

An interesting experiment is presented where robots join the climbing enthusiasts on the cliff. Bretl and co-workers present a four-limbed robot LEMUR IIb robot is being developed at Stanford University to free-climb almost vertical rock surfaces. Relying more on the planning algorithm than a specific hardware, the robot has been demonstrated to climb an indoor, near-vertical surface with small, arbitrarily distributed, natural features.

The robots in this last article by Yim and co-workers may not be walking in the traditional sense, as they do not even possess limbs, but they are capable of conforming to the environment and locomoting through it. The robots are made up of modular robots and they are capable of forming a conforming loop to move through more open space and snake-like configurations for narrow space. They are suitable for search and rescue operations through tight, unstructured environments that are too narrow for human to pass through. These robots showed an amazing capability in moving through all the unstructured terrains.

Adaptive Walking of a Quadruped Robot in Outdoor Environment based on Biological Concepts

Hiroshi Kimura and Yasuhiro Fukuoka

Graduate School of Information Systems, University of Electro-Communications
{hiroshi, fukuoka}@kimura.is.uec.ac.jp

Abstract. We have been trying to induce a quadruped robot to walk with medium walking speed on irregular terrain based on biological concepts. We propose the essential conditions for stable dynamic walking on irregular terrain in general, and we design the mechanical system and the neural system by comparing biological concepts with those essential conditions described in physical terms. PD-controller at joints constructs the virtual spring-damper system as the visco-elasticity model of a muscle. The neural system model consists of a CPG (central pattern generator), reflexes and responses. We report our experimental results of dynamic walking on irregular terrain in outdoor environment using a self-contained quadruped robot in order to verify the effectiveness of the designed neuro-mechanical system.

1 Introduction

Many previous studies of legged robots have been performed, including studies on running and dynamic walking on irregular terrain. However, all of those studies assumed that the structure of terrain was known, even though the height of the step or the inclination of the slope was unknown. The purpose of this study is to realize high-speed mobility on irregular terrain with less knowledge of it using a mammal-like quadruped robot, the dynamic walking of which is less stable than that of hexapod robots, by referring to the marvelous abilities of animals to autonomously adapt to their environment.

As many biological studies of motion control progressed, it has become generally accepted that animals' walking is mainly generated at the spinal cord by a combination of a CPG (central pattern generator) and reflexes receiving adjustment signals from a cerebrum, cerebellum and brain stem[1]. A great deal of the previous research on this attempted to generate walking using a neural system model, including studies on dynamic walking in simulation[2–4], and real robots[5–7]. But autonomously adaptive dynamic walking on irregular terrain was rarely realized in those earlier studies except for our studies [8,9]. This paper reports on our progress using a self-contained (power autonomous) quadruped robot called "Tekken2" (Fig.1) which was newly developed for adaptive walking on irregular terrain in outdoor environment (Fig.2).

2 Self-contained Quadruped: Tekken2

Each leg of Tekken2 has a hip pitch joint, a hip yaw joint, a knee pitch joint, and an ankle pitch joint (Fig.1). The direction in which Tekken2 walks can be changed by

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 429–438, 2006.

using the hip yaw joints. Two rate gyro sensors and two inclinometers are mounted on the body in order to measure the body pitch and roll angles.

In order to obtain appropriate mutual entrainment between neural system and mechanical system, mechanical system should be well designed to have the good dynamic properties. In addition, performance of dynamic walking such as adaptability on irregular terrain, energy efficiency, maximum speed and so on highly depends on the mechanical design. The design concepts of Tekken2 are:

(1) high power actuators and small inertia moment of legs for quick motion and response,
(2) small gear reduction ratio for high backdrivability to increase passive compliance of joints,
(3) small mass of the lowest link of legs to decrease impact force at collision,
(4) small contacting area at toes to increase adaptability on irregular terrain.

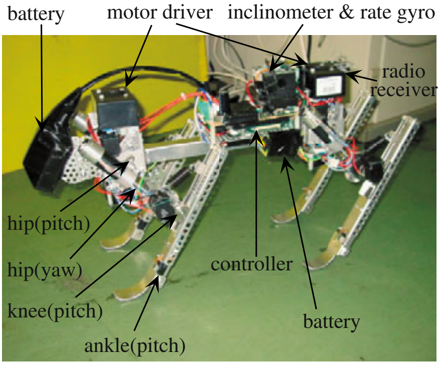


Fig. 1. Tekken2. The length of the body and a leg in standing are 30 [cm] and 20 [cm]. The weight including batteries is 4.3 [kg].

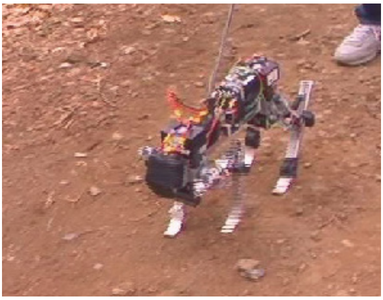


Fig. 2. Photo of walking in outdoor environment.

3 Necessary Conditions for Stable Dynamic Walking

We propose the necessary conditions for stable dynamic walking on irregular terrain, which can be itemized in physical terms:

(a) the period of the walking cycle should be shorter enough than the upper bound of it, in which stable dynamic walking can be realized[10],
(b) the swinging legs should be free to move forward during the first period of the swing phase,
(c) the swinging legs should land reliably on the ground during the second period of the swing phase,

(d) the angular velocity of the supporting legs relative to the ground should be kept constant during their pitching motion or rolling motion around the contact points at the moment of landing or leaving,
(e) the phase difference between rolling motion of the body and pitching motion of legs should be maintained regardless of a disturbance from irregular terrain, and
(f) the phase differences between the legs should be maintained regardless of delay in the pitching motion of a leg receiving a disturbance from irregular terrain.

We design the neural system for these necessary conditions to be satisfied in order to realize adaptive walking.

4 Implementation of Neural System for Adaptive Walking

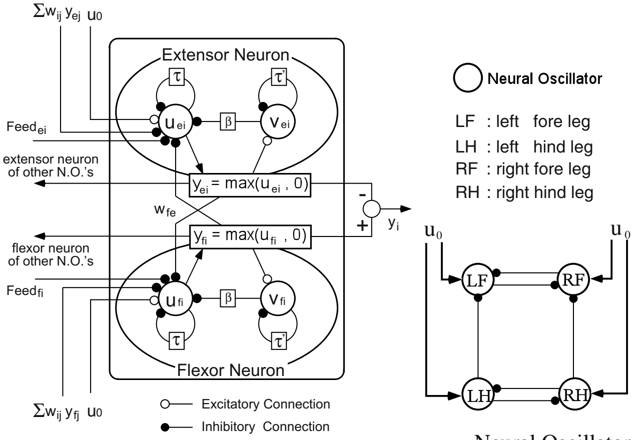
The basic neural system model of Tekken2 is same with the one of Tekken1. We define a "reflex" as joint torque generation based on sensor information and a "response" as CPG phase modulation through sensory feedback to a CPG. Several reflexes and responses are newly employed in Tekken2 for adaptive walking in outdoor environment.

4.1 Rhythmic Motion by CPG

We construct the neural system centering a neural oscillator as a model of a CPG, since the exchange between the swing and stance phases in the short term and the quick adjustment of these phases on irregular terrain are essential in the dynamic walking of a quadruped where the unstable two-legged stance phase appears. Although actual neurons as a CPG in higher animals have not yet become well known, features of a CPG have been actively studied in biology, physiology, and so on. Several mathematical models were also proposed, and it was pointed out that a CPG has the capability to generate and modulate walking patterns and to be mutually entrained with a rhythmic joint motion [1]. As a model of a CPG, we used a neural oscillator proposed by Matsuoka , and applied to the biped simulation by Taga[2]. A single neural oscillator consists of two mutually inhibiting neurons (Fig.3-(a)). Each neuron in this model is represented by the following nonlinear differential equations:

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

where the suffix e, f, and i mean an extensor neuron, a flexor neuron, and the i-th neural oscillator, respectively. $u_{\{e,f\}i}$ is u_{ei} or u_{fi}, that is, the inner state of an extensor neuron or a flexor neuron of the i-th neural oscillator; $v_{\{e,f\}i}$ is a variable representing the degree of the self-inhibition effect of the neuron; y_{ei} and y_{fi} are



(a) Neural Oscillator (b) Neural Oscillator Network

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

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

In Fig.3-(a), the output of a CPG is a phase signal: y_i .

$$y_i = -y_{ei} + y_{fi} \tag{2}$$

The positive or negative value of y_i corresponds to activity of a flexor or extensor neuron, respectively.

We use the following hip joint angle feedback as a basic sensory input to a CPG called a "tonic stretch response" in all experiments of this study. This negative feedback makes a CPG be entrained with a rhythmic hip joint motion.

$$Feed_{e \cdot tsr} = k_{tsr}(\theta - \theta_0), \;\; Feed_{f \cdot tsr} = -Feed_{e \cdot tsr} \tag{3}$$

$$Feed_{\{e,f\}} = Feed_{\{e,f\} \cdot tsr} \tag{4}$$

where θ is the measured hip joint angle, θ_0 is the origin of the hip joint angle in standing and k_{tsr} is the feedback gain. We eliminate the suffix i when we consider a single neural oscillator.

By connecting the CPG of each leg (Fig.3-(b)), CPGs are mutually entrained and oscillate in the same period and with a fixed phase difference. This mutual entrainment between the CPGs of the legs results in a gait. The gait is a walking pattern, and can be defined by phase differences between the legs during their pitching motion. The typical symmetric gaits are a trot and a pace. Diagonal legs and lateral legs are paired and move together in a trot gait and a pace gait, respectively. A walk gait is the transversal gait between the trot and pace gaits. We used a trot gait and a walk gait. The autonomous gait transition in changing walking speed was discussed in our former study [9].

Although the size and weight of Tekken2 are different from those of Tekken1, the values of the parameters of CPGs used for Tekken2 were same with those used for Tekken1.

4.2 Virtual Spring-damper System

We employ the model of the muscle stiffness, which is generated by the stretch reflex and variable according to the stance/swing phases, adjusted by the neural system. The muscle stiffness is high in a stance phase for supporting a body against the gravity and low in a swing phase for compliance against the disturbance. In order to generate each motion such as swinging up (A), swinging forward (B) and pulling down/back of a supporting leg (C), all joints of Tekken2 are PD controlled to move to their desired angles in each of three states (A, B, C). The timing for all joints of a leg to switch to the next state are:

$A \rightarrow B$: when the hip joint of the leg reaches the desired angle of the state (A)
$B \rightarrow C$: when the CPG extensor neuron of the leg becomes active ($y_i \leq 0$)
$C \rightarrow A$: when the CPG flexor neuron of the leg becomes active ($y_i > 0$)

Since Tekken2 has high backdrivability with small gear ratio in each joint, PD-controller can construct the virtual spring-damper system with relatively low stiffness coupled with the mechanical system. Such compliant joints of legs can improve the passive adaptability on irregular terrain.

4.3 CPGs and Pitching Motion of Legs

The diagram of the pitching motion control consisting of CPGs and the virtual spring-damper system is shown in the middle part of Fig.4. Joint torque of all joints is determined by the PD controller, corresponding to a stretch reflex at an α motor neuron in animals. The desired angle and P-gain of each joint is switched based on the phase of the CPG output: y_i in Eq.(2) as described in Section *4.2*. As a result of the switching of the virtual spring-damper system and the joint angle feedback signal to the CPG in Eq.(4), the CPG and the pitching motion of the leg are mutually entrained.

The necessary condition (f) can be satisfied by the mutual entrainment between CPGs and the pitching motion of legs, and the mutual entrainment among CPGs[8].

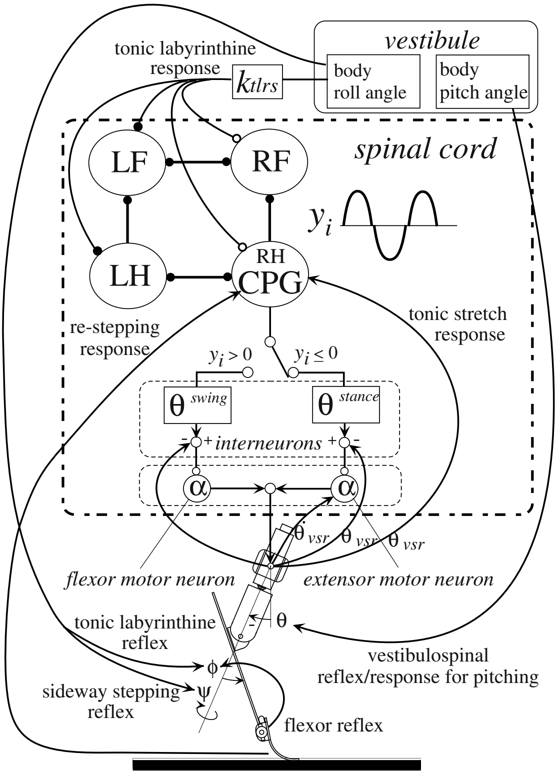


Fig. 4. Control diagram for Tekken2. PD-control at the hip yaw and knee pitch joints are eliminated in this figure.

4.4 Reflexes and Responses

Referring to biological knowledge, we employed the several reflexes and responses (Table 1, Fig.4) to satisfy the necessary conditions (b)∼(e) described in physical terms in Section **3** in addition to the stretch reflex and response described in Section *4.2* and *4.1*. In Table 1, the sideway stepping reflex and the re-stepping reflex/response were newly employed on Tekken2. Other reflexes and responses had already been employed on Tekken1 [9].

Sideway Stepping Reflex to Stabilize Rolling Motion It is known that the adjustment of the sideway touchdown angle of a swinging leg is effective in stabilizing rolling motion against disturbances[11,12]. We call this a "sideway stepping reflex," which helps to satisfy the condition (d) during rolling motion. The sideway stepping reflex is effective also in walking on a sideway inclined slope.

For examples, when Tekken2 walks on a right-inclined slope (Fig.5), Tekken2 continues to walk while keeping the phase differences between left and right lges

Table 1. Reflexes and Responses employed on Tekken2.

	sensed value or event	activated on	necessary conditions
flexor reflex	collision with obstacle	sw	(b)
stepping reflex	forward speed	sw	(d)
vestibulospinal reflex/response	body pitch angle	sp	(d)
tonic labyrinthine response	body roll angle	sp&sw	(c),(d),(e)
sideway stepping reflex	body roll angle	sw	(d)
re-stepping reflex/response	loss of ground contact	sw	(d)

The sp and sw mean the supporting leg and swinging leg, respectively.
The corresponding necessary conditions are described in Section **3**.

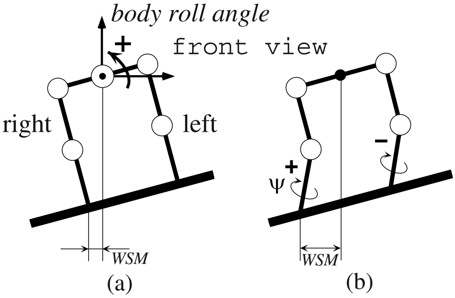


Fig. 5. Walking on a sideway inclined slope. (a):without a sideway stepping reflex, (b):with.

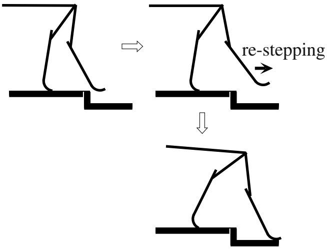


Fig. 6. Re-stepping reflex and response.

with the help of the tonic labyrinthine response. But Tekken2 cannot walk straight and shifts its walking direction to the right due to the difference of the gravity load between left and right legs. In addition, Tekken2 typically falls down to the right for the perturbation from the left in the case of Fig.5-(a), since the wide stability margin: WSM [1] is small. The sideway stepping reflex helps to stabilize the walking direction and to prevent the robot from falling down while keeping WSM large on such sideway inclined slope (Fig.5-(b)).

Since Tekken2 has no joint round the roll axis, the sideway stepping reflex is implemented as changing the desired angle of the hip yaw joint from 0 to ψ^* according to Eq.(5).

$$\psi^* = \delta(leg)\, k_{stpr} \times \text{(body roll angle)} \tag{5}$$

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

[1] the shortest distance from the projected point of the center of gravity to the edges of the polygon constructed by the projected points of legs independent of their stance or swing phases[9].

Re-stepping Reflex and Response for Walking Down a Step When loss of ground contact is detected in a swing phase while walking over a ditch, a cat activates re-stepping to extend the swing phase and make the leg land on the forwarder position[13]. We call this "re-stepping reflex/response," which is effective for the necessary condition (c) and (d) to be satisfied also in walking down a large step (Fig.6).

4.5 Active Landing Control on the Soft Ground

While walking on the soft ground, the rolling motion is much disturbed since it takes longer to establish the reliable landing of the swinging legs. Tekken2 changes the state of the virtual spring-damper system from the swinging to the stance before the actual contact of a leg on the ground, when the output phase of a CPG changes from the flexor neuron active phase to the extensor neuron active phase as described in Section *4.2*. This control contributes to obtain the reliable landing of the swinging legs as soon as possible, and helps the necessary condition (c) be satisfied.

5 Experiments

5.1 Walking on a Sideway Inclined Slope

We made Tekken2 walk on a right-inclined slope of 4 [deg] (0.07 [rad]) in indoor environment in order to confirm the effectiveness of a sideway stepping reflex. As a result of the experiment, the body roll angle and the hip yaw angle ψ of the right foreleg and left foreleg are shown in Fig.7, where Tekken2 walked on the right-inclined slope from 3 to 6.7 [sec]. In Fig.7, we can see that the body roll angle was positive (0.03∼0.18 [rad]) while walking on the right-inclined slope and WSM was kept large (approx. 0.5∼0.7). The hip yaw joint of the right foreleg moved to the outside of the body (right) by approx. 0.13 [rad] due to the sideway stepping reflex in the swing phase and moved to the inside of the body by approx. -0.04 [rad] due to the gravity load in the stance phase. We can see similar motion of the left foreleg in Fig.7. Consequently, Tekken2 succeeded in straight walking on the sideway inclined slope. Without a sideway stepping reflex, rolling motion of Tekken2 was much disturbed on the sideway inclined slope and Tekken2 sometimes failed in keeping walking.

5.2 Walking Down a Large Step

Tekken2 successfully walked down a large step with approx. 0.5 [m/s] speed using the re-stepping reflex/response. In Fig.8, a re-stepping response was activated when the contact of the right fore leg had not been detected for 0.14 [s] after the activity of the flexor neuron became zero. Without the re-stepping reflex/response, Tekken2 typically fell down forward because fore legs landed on the backwarder position excessively and could not depress the increased forward speed.

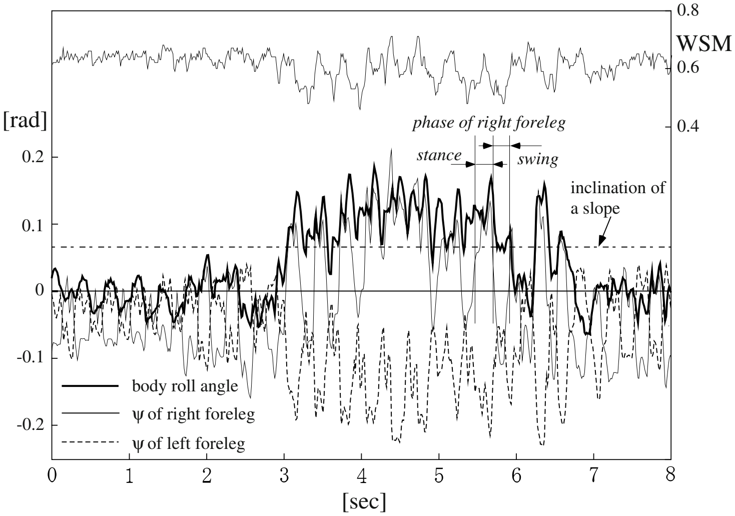


Fig. 7. Walking on a right-inclined slope of 0.07 [rad] (4 [deg]) with a sideway stepping reflex.

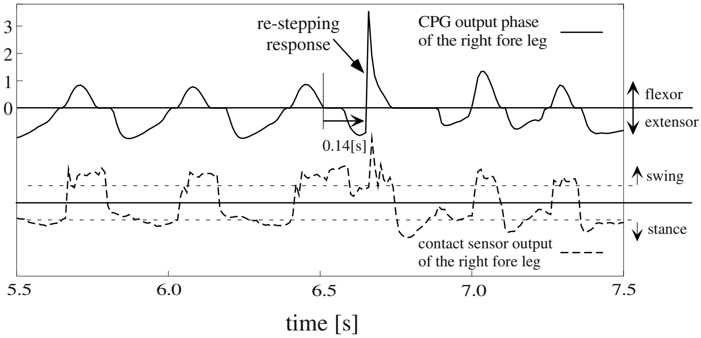


Fig. 8. Walking down a step of 7 [cm] in height with a re-stepping reflex/response.

5.3 Outdoor Experiments

Even on a paved road in outdoor environment, there exist a slope of 3 [deg] at most, bumps of 1 [cm] in height and small pebbles everywhere. With all responses and reflexes described in Section *4.4*, Tekken2 successfully maintained a stable gait on the paved road for 4 [min] with approx. 0.5 [m/s] speed while changing its walking speed and direction by receiving the operation commands from the radio controller. In addition, the effectiveness of the active landing control on the soft ground was confirmed by the successful experiment of walking on the natural ground with scattered pebbles and grasses (Fig.2). MPEG footage of these experiments can be seen at: *http://www.kimura.is.uec.ac.jp*.

6 Conclusion

In this study, we designed the neural system consisting of CPGs, responses, and reflexes referring to biological concepts while taking the necessary conditions for adaptive walking into account. In this neural system model, the relationships among CPGs, sensory input, reflexes and the mechanical system were simply defined, and motion generation and adaptation were emergingly induced by the coupled dynamics of a neural system and a mechanical system by interacting with the environment.

In order to make the self-contained quadruped robot walk in outdoor natural environment, we newly employed a sideway stepping reflex, a re-stepping reflex/response, and the active landing control of the swinging legs. We should employ additional reflexes and responses, and also navigation ability at the high level using vision to increase the degrees of terrain irregularity which Tekken2 can cope with.

References

1. Grillner, S. 1981. Control of locomotion in bipeds, tetrapods and fish. *Handbook of Physiology II* American Physiol. Society, Bethesda, MD, pp. 1179–1236.
2. Taga, G. 1995. A model of the neuro-musculo-skeletal system for human locomotion II. - real-time adaptability under various constraints. *Biolog. Cybern.* 73:113–121.
3. Ijspeert, A. J. 2001. A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biolog. Cybern.* 84:331–348.
4. Tomita, N., Yano, M. 2003. A Model of Learning Free Bipedal Walking in Indefinite Environment - Constraints Self-Emergence/Self-Satisfaction Paradigm -. *Prof. of SICE Annual Conf.*, pp. 3176–3181.
5. Ilg, W., Albiez, J., Jedele, H., Berns, K., and Dillmann, R. 1999. Adaptive periodic movement control for the four legged walking machine BISAM. *Proc. of ICRA1999*, pp. 2354–2359.
6. Tsujita, K., Tsuchiya, K., and Onat, A. 2001. Adaptive Gait Pattern Control of a Quadruped Locomotion Robot, *Proc. of IROS2001*, pp. 2318–2325.
7. Lewis, M. A., Etienne-Cummings, Hartmann, M. J., Xu, Z. R., and Cohen, A. H. 2003. An in silico central pattern generator: silicon oscillator, coupling, entrainment, and physical computation. *Biolog. Cybern.* 88:137–151.
8. Kimura, H., Fukuoka, Y., and Konaga, K. 2001. Adaptive dynamic walking of a quadruped robot using neural system model. *Advanced Robotics* 15(8):859–876.
9. Fukuoka, Y., Kimura, H., and Cohen, A. H. 2001. 2003. Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain based on Biological Concepts, *Int. Journal of Robotics Research* 22(3-4):187–202.
10. Kimura, H., Shimoyama, I., and Miura, H. 1990. Dynamics in the dynamic walk of a quadruped robot. *Advanced Robotics* 4(3):283–301.
11. Miura, H., and Shimoyama, I. 1984. Dynamical walk of biped locomotion. *Int. J. Robotics Research* 3(2):60–74.
12. Bauby, C. E., and Kuo, A. D. 2000. Active control of lateral balance in human walking. *J. of Biomechanics* 33:1433–1440.
13. Hiebert, G., et al. 1994. Corrective responses to loss of ground support during walking II, comparison of intact and chronic spinal cats, *J of Neurophys.* 71:611–622.

Legged Odometry from Body Pose in a Hexapod Robot

Pei-Chun Lin[1], Haldun Komsuoḡlu[2], and Daniel E. Koditschek[3]

[1] Department of Mechanical Engineering, The University of Michigan, Ann Arbor, Michigan, USA
pclin@umich.edu
http://www.umich.edu/~pclin

[2] Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan, USA
hkomsuog@umich.edu
http://www.eecs.umich.edu/~hkomsuog

[3] Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan, USA
kod@eecs.umich.edu
http://ai.eecs.umich.edu/people/kod/

Abstract. We report on a continuous time odometry scheme for a walking hexapod robot built upon a previously developed leg-strain based body pose estimator. We implement this estimation procedure and odometry scheme on the robot RHex and evaluate its performance at widely varying speeds and over different ground conditions by means of a 6 degree of freedom vision based ground truth measurement system (GTMS). We also compare the performance to that of sensorless odometry schemes — both legged as well as on a wheeled version of the robot — using GTMS measurements of elapsed distance.

1 Introduction

The hexapod, RHex [1], exhibits unprecedented mobility for a legged autonomous robot [2]. Motivated by the desire to improve dynamical performance of the present open loop controller through the introduction of continuous rigid body state estimates [3], we have recently reported on the development of a novel leg-strain based body pose[1] sensor [4]. Within a stride that maintains at least three non-collinear toes fixed in ground contact, this system delivers a continuous stream of body frame estimates relative to the ground frame at 300 Hz [4]. Additional body proprioceptive sensors — a three degree of freedom (DOF) rate gyro and a six DOF accelerometer array — now under development will lead to the complete 6 DOF continuous time state estimator necessary for advanced feedback control. Notwithstanding this central focus upon sensory feedback control for dynamical gaits, it seems quite interesting to explore along the way the capabilities of the strain based body pose sensor in isolation from these other proprioceptive modalities within quasi-static operational regimes.

[1] We use the term *body pose* to denote the position and orientation of the robot's present body frame relative to the local ground tripod frame within this tripod stance.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 439–448, 2006.

In this paper we apply the leg strain based body pose estimator reported in [4] to develop a new sensor that provides complete legged odometry for alternating tripod gaits satisfying the previously stated ground contact conditions: at least three noncollinear toes on the ground at all times. We present a straightforward but general odometry-from-pose algorithm that uses periods of "double support" (where two triads of legs are simultaneously in ground contact) to express the frame of the successor triad in terms of the frame of the predecessor, thereby "remembering" the body's frame relative to the world coordinate system prior to the onset of motion. We stress the system by placing it on increasingly slippery surfaces and report the degradation in performance as a function of surface traction. We compare our sensed odometry scheme to the sensorless alternative of average distance traveled per leg stride[2] as well as to a standard axle revolution count implemented on a wheeled version of the same robot.

The word *odometry*[3] seems to be used in several different contexts in robotics. The traditional notion associated with wheeled vehicles corresponds to planar 3 DOF dead-reckoning by encoder-like sensors that count the rotation of wheels in the absence of other sensors. Here, we generalize this approach to examine the quality of "complete" 6 DOF dead-reckoning for the legged robot — the position and orientation of the robot's present body frame relative to that prior to the initiation of motion.

In reviewing the literature it is similarly useful to distinguish between the positioning problem and the orientation problem. The positioning problem has been treated extensively for wheeled vehicles where the traditional "elapsed distance" measure is combined with body proprioception (accelerometer and gyro data) [5–7] to obtain high quality estimates of the local translation. More recent research is concerned with the fusion of exteroceptive (vision [8], ultrasonic [9], sonar [10], or GPS [11]) sensor data for purposes of obtaining precise global positioning by periodically recalibrating the accumulated dead-reckoning error. In contrast, body orientation estimation has been investigated in the legged robot literature for high-DOF bipeds [12, 13], whose stability and balance must be actively controlled by state feedback. There is some prior work on exteroceptive (vision based) [14–16] approaches to positioning for legged machines but no account of the legged odometry problem. Thus, our treatment of complete 6 DOF odometry appears to be novel.

Section 2 presents the odometry algorithm — a matter of standard geometry and linear algebra involving the position of the presumably known contact-toes in body coordinates. Section 3 examines the accuracy of the resulting odometry estimator implemented on RHex pictured in Figure 1(upper left) using our previously developed leg strain based body pose sensor. We use an independent camera based ground truth measurement system (GTMS) to assess the deterioration in performance over increasingly slippery ground and at varying speeds, as well as to compare the quality of elapsed distance measurements arising from the leg sensor based estimator,

[2] We thank Dr. Johann Borenstein for suggesting this comparison to us.

[3] The Oxford English Dictionary traces the meaning back to its Greek roots *measuring the way*.

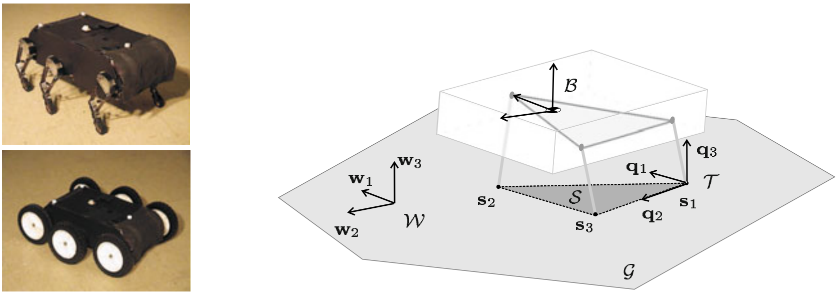


Fig. 1. (Left) RHex and Wheeled RHex; (Right) Sketch illustrating the robot on a flat ground plane, $\mathcal{G}$, within a single stance phase during which it is supported by only three legs whose toes define the support triangle, $\mathcal{S}$. Attached to the support triangle, we define a tripod coordinate system, $\mathcal{T}$.

a legged open loop scheme, and the traditional axle revolution count performed on a wheeled version of RHex pictured in Figure 1(lower left). Section 4 concludes this paper.

2 Computation of Odometry from Body Pose

We develop our odometry algorithm for a hexapod robot in an alternating tripod walking gait because of its utility for RHex. However, the computations below generalize to a family of gaits characterized by two conditions: A) the body is supported by at least three legs with non-collinear toes at any given time; and B) ground contact legs have no toe slippage[4].

In an alternating tripod walking gait we identify two intervals: the *single stance phase* when the body is supported by only one tripod, which denotes the mode of leg contact wherein the three toes of the front and rear ipsilateral legs and the middle contralateral leg of a tripod are all in contact with the ground as depicted in Figure 1(Right); and the *double stance phase* when all legs are in ground contact. This suggests a hierarchically structured algorithm with two levels: 1) a low level, originally presented in [4] and briefly reviewed in Section 2.1, operating during individual single stance phases computing the body pose in a locally defined coordinate system termed tripod coordinate system, $\mathcal{T}$, which is rigidly related to the world coordinate system, $\mathcal{W}$; and 2) a high level sequential composition method relating the tripod coordinate systems in consecutive single stance phases to evaluate the odometry with respect to the world coordinate system, $\mathcal{W}$.

[4] These conditions guarantee that the toe contacts yield a well defined coordinate system fixed in the world frame. Appropriate generalizations of the calculations would extend the computation of odometry to other kinds of legged robots, like quadrupeds or even bipeds with foot (surface) contact.

2.1 Body Pose in Single Stance

It is intuitively clear that knowledge of the configuration relative to the body of each leg in contact with the ground, together with information about the ground contact points yields complete pose information. Assume a "leg model," $\mathbf{s}_i(z_i)$, for each tripod toe, $i = 1, 2, 3$, where z_i denotes the sensory measurements available regarding the configuration of the kinematic chain connecting the robot body to the i$^{\text{th}}$ toe and $\mathbf{s}_i$ represents the point of toe contact with respect to the robot body coordinate system, $\mathcal{B}$. For example, in the RHex implementation, $\mathbf{s}_i(\gamma_i, \sigma_i) = \mathbf{n}_i^{\gamma_i} \circ \mathbf{m}_i(\sigma_i), i = 1, 2, ..., 6$ detailed in [4] consists of the kinematic parameters, γ_i, relating the i$^{\text{th}}$ hip frame, $\mathcal{C}_i$, to the body frame, $\mathcal{B}$, together with the strain across the compliant portion of the leg as read from the sensor suite, σ_i. By defining the tripod coordinate system, $\mathcal{T}$, detailed in [4], whose orthonormal basis $(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$ with origin at $\mathbf{s}_1$ is a function of $\mathbf{s}_{i, i=1,2,3}$ as

$$\mathbf{q}_1 := \frac{\mathbf{e}_1 - (\mathbf{q}_2^T \mathbf{e}_1)\mathbf{q}_2}{\left|\left|\mathbf{e}_1 - (\mathbf{q}_2^T \mathbf{e}_1)\mathbf{q}_2\right|\right|_2} \quad \mathbf{q}_2 := \mathbf{e}_2 \quad \mathbf{q}_3 := \mathbf{q}_1 \times \mathbf{q}_2$$

represented in the body coordinate system, $\mathcal{B}$, where $\mathbf{e}_1 := \frac{\mathbf{s}_2 - \mathbf{s}_1}{||\mathbf{s}_2 - \mathbf{s}_1||_2}$ and $\mathbf{e}_2 := \frac{\mathbf{s}_3 - \mathbf{s}_1}{||\mathbf{s}_3 - \mathbf{s}_1||_2}$, we can construct a homogeneous coordinate transformation, $\mathbf{h} : \mathcal{B} \to \mathcal{T}$, relates the coordinates, $\mathbf{b}$, in body coordinate system, $\mathcal{B}$, to that in the tripod coordinate system, $\mathcal{T}$,

$$\mathbf{h}(\mathbf{b}) := \mathbf{B}\,(\mathbf{b} - \mathbf{s}_1) \tag{1}$$

where $\mathbf{B} := [\mathbf{q}_1\, \mathbf{q}_2\, \mathbf{q}_3]^T$.

2.2 Leg Based Odometry via Composition of Single Stance Measurements

We will now detail how the single stance phase pose computations described above can be integrated over multiple steps to generate this continuous computation of absolute body pose.

First, compute the homogeneous transformation between the tripod coordinate systems of consecutive single stance phases , $\mathcal{T}_j$ and $\mathcal{T}_{j+1}$, as follows. Assuming the toes defining these tripod coordinate systems are stationary (there is no slippage and no liftoff) throughout their presumed stances, and assuming there is an adequate period of double support (the two stance phases overlap for a time sufficient to complete their respective single stance pose computations), both coordinate systems are related to the same (moving) body coordinate system, $\mathcal{B}$. Now, assuming that the prior tripod coordinate system, $\mathcal{T}_j$, has been expressed in world coordinates, the representation of its successor, $\mathcal{T}_{j+1}$, in the world coordinate system follows by the properties of rigid transformations in a straightforward manner that we now detail.

We index single stance phases, $j = 0, 1, ...$, according to their occurrences over the course of locomotion where we denote the j$^{\text{th}}$ single stance tripod coordinate system by $\mathcal{T}_j$. Without loss of generality, assume that the tripod coordinate system of the first single stance phase coincides with the world coordinate system, $\mathcal{T}_0 \equiv \mathcal{W}$.

The key for establishing the odometry is there exists an intermediate double stance phase where all legs are in ground contact, which allows us to find relationship between tripod coordinate systems of two consecutive single stance phases, $\mathbf{g}_j^{j-1} : \mathcal{T}_j \to \mathcal{T}_{j-1}, \mathbf{g}_j^{j-1} := \mathbf{h}_{j-1} \circ \mathbf{h}_j^{-1}$, before and after this double stance phase. In the on-line implementation where the robot operates on "flat ground," we determine double stance by evaluating a "planarity measure," $\rho(\mathbf{s})$, based upon the sampled covariance matrix of the "sampled toe distribution[5]," $\mathbf{S} = \{\mathbf{s}_1, .., \mathbf{s}_6\}$. The status of double stance is checked if the planarity measure, $\rho(s)$, is dropped under a empirically set threshold, $\overline{\rho}$. When this event is detected we compute the transformations between tripod coordinate systems of consecutive single stance phases, $\mathbf{g}_j^{j-1}$. Figure 2(left) illustrates the flow diagram of the complete on-line algorithm and Figure 2(Right) illustrates the sequential relationship between the tripod coordinate systems, $\mathcal{T}_j$. As a direct consequence, the map, $\mathbf{g}_j^0 : \mathcal{T}_j \to \mathcal{W}$, relating the j$^{\text{th}}$ tripod coordinate system, $\mathcal{T}_j$, to the world coordinate system, $\mathcal{W} \equiv \mathcal{T}_0$, can be defined recursively,

$$\mathbf{g}_j^0 := \mathbf{g}_{j-1}^0 \circ \mathbf{g}_j^{j-1}, j \in \{1, 2, ...\}$$

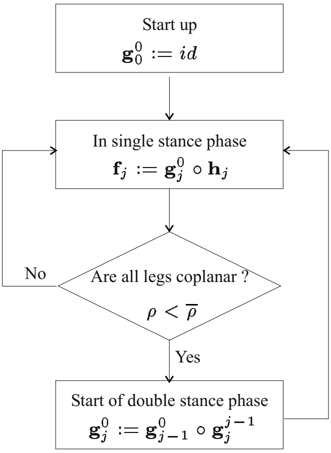
where $\mathbf{g}_0^0 := id$. This in turn leads to the definition of the map, $\mathbf{f}_j : \mathcal{B} \to \mathcal{W}$, that relates the body coordinate system, $\mathcal{B}$, during the j$^{\text{th}}$ single stance phase to the world coordinate system, $\mathcal{W}$, $\mathbf{f}_j := \mathbf{g}_j^0 \circ \mathbf{h}_j$, which is a homogeneous transformation that we prefer to write as

$$\mathbf{f}_j(\mathbf{b}) := \mathbf{A}_j \left(\mathbf{b} - \mathbf{c}_j\right) \qquad (2)$$

where $\mathbf{A}_j$ and $\mathbf{c}_j$ denote the rotation matrix and COM translation of the homogeneous transformation accordingly.

The body pose is now read off the entries of the transformation matrices in the familiar manner. The COM translation in lateral (x) and fore/aft (y) directions as well as body orientation in yaw (γ) are computed from rigid transformation, $\mathbf{f}$, shown in (2) in the standard manner [17] because these three configuration variables require the whole history of locomotion. Taking advantage of the flat ground plane, $\mathcal{G}$, the COM translation in vertical (z) direction as well as body orientation in pitch (α) and roll (β) are computed from the within-stride rigid transformation, $\mathbf{h}$, shown in (1) in the standard manner [17] to reduce the possible accumulation error.

[5] Namely, let $\mathbf{P} := \sum_{i=1}^{6} \mathbf{s}_i \mathbf{s}_i^T - \overline{\mathbf{s}}\overline{\mathbf{s}}^T$ where $\overline{\mathbf{s}} := \frac{1}{6} \sum_{i=1}^{6} \mathbf{s}_i$ is the sample mean. Denoting by, $\mathbf{v}$, the eigenvector associated with the smallest eigenvalue of $\mathbf{P}$, we define $\rho(\mathbf{s}) := \frac{1}{6} \sum_{i=1}^{6} [(\mathbf{s}_i - \overline{\mathbf{s}}).\mathbf{v}]^2$ which gives the mean squared distance of the distribution around the best common plane.


Start up
$\mathbf{g}_0^0 := id$
In single stance phase
$\mathbf{f}_j := \mathbf{g}_j^0 \circ \mathbf{h}_j$
Are all legs coplanar ?
$\rho < \overline{\rho}$
No
Yes
Start of double stance phase
$\mathbf{g}_j^0 := \mathbf{g}_{j-1}^0 \circ \mathbf{g}_j^{j-1}$

$$\begin{tikzcd}
\mathcal{T}_0 \arrow[r, "id"] & \mathcal{W} \\
\mathcal{T}_1 \arrow[u, bend left, "\mathbf{g}_1^0"] & \\
\mathcal{T}_2 \arrow[u, bend left, "\mathbf{g}_2^1"] & \\
\vdots \arrow[u, bend left, "\mathbf{g}_3^2"] & \\
\mathcal{T}_{j-1} & \\
\mathcal{T}_j \arrow[u, bend left, "\mathbf{g}_j^{j-1}"] & \mathcal{B} \arrow[l, "\mathbf{h}_j"] \arrow[uuuuu, bend right, "\mathbf{f}_j"']
\end{tikzcd}$$

Fig. 2. (Left) Flow chart for the within-stride body pose computation and leg odometry functions; (Right) Commutative diagram relating the tripod coordinate systems, $\mathcal{T}_j$.

3 Performance

We now evaluate the performance of the odometry scheme implemented on RHex in three different scenarios: 1) at varying walking speeds; 2) over dramatically different ground conditions; 3) in comparison to wheeled and sensorless legged versions of the same machine. We report the results using the mean and standard deviation from five runs in each experiment set. The reader should note that RHex's relatively constrained kinematics preclude the exercise of its yaw degree of freedom when it walks with no aerial phase and no toe slippage, hence the implementation we discuss in this section will entail no data of that nature. We compare the estimator output to that of the GTMS (independent visual ground truth measurement system introduced in [4]) with respect to five configuration components — the lateral (x), fore-aft (y), and vertical (z) components of COM translation as well pitch (α) and roll (β) — all in world coordinates, $\mathcal{W}$. Figure 3 plots the comparison for each component over a typical run. We will quantify performance by presenting the standard root mean squared (RMS) error, given by

$$\zeta(d, \hat{d}) := \sqrt{(\left\|d - \hat{d}\right\|_2^2 / M)}$$

where $d = (d_1, ..., d_M)$ and $d_i := (x_i, y_i, z_i, \alpha_i, \beta_i)$ represents the state trajectory from GTMS; $\hat{d}$ denotes the corresponding state trajectory from output of the algorithm.

Table 1 summarizes the outcome of repeated runs at slow (0.25 m/s), medium (0.35 m/s), and fast (0.51 m/s) speeds, which also includes GTMS measured elapsed distance, ν, and number of tripod strikes for each run, υ, as references. RMS error

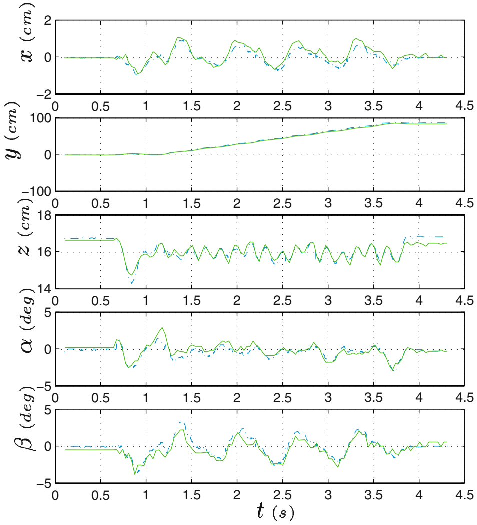


Fig. 3. Odometry measured by GTMS (solid green line) and computed according to our algorithm (dashed blue line).

Table 1. RMS Odometry Error at Varying Walking Speeds (each row representing data averaged over 5 runs with distance, ν and number of strides, υ, shown in reference)

Walking Speed		State					Reference	
		x (cm) avg (std)	y (cm) avg (std)	z (cm) avg (std)	α (deg) avg (std)	β (deg) avg(std)	ν (cm) avg(std)	υ avg(std)
Slow	(0.25 m/s)	0.42 (0.13)	4.53 (0.80)	0.30 (0.08)	0.78 (0.09)	0.73 (0.06)	84.0 (6.2)	8.4 (0.5)
Medium	(0.35 m/s)	0.41 (0.13)	1.42 (0.32)	0.22 (0.04)	0.60 (0.10)	0.71 (0.08)	83.8 (8.0)	7.8 (0.7)
Fast	(0.51 m/s)	0.39 (0.05)	1.41 (0.25)	0.27 (0.04)	0.63 (0.08)	0.68 (0.08)	84.8 (6.5)	7.4 (0.7)

values compared to the robot size (50cm x 25cm x 15cm) indicate successful odometry computation with mean error in angular states less than 1 degree and lateral and vertical positions less than 1 cm. Since slippage during double stance has significant impact on the fore/aft direction of the robot, the corresponding RMS error of y is the largest. We observe that the error in the fore/aft position decreases with increasing speed, attributable to: 1) shorter double stance decreasing the error in the sequential composition computation; and 2) increasing ground reaction forces that decrease slippage.

Table 2 summarizes the outcome of repeated walks at slow speed over four types of surfaces: cardboard ($\mu_s = 0.65$, $\mu_k = 0.60$); plastic ($\mu_s = 0.33$, $\mu_k = 0.27$); plastic with wet soap ($\mu_s = 0.20$, $\mu_k = 0.11$); and plastic with dry soap ($\mu_s = 0.07$, $\mu_k = 0.05$), where the static friction ("stiction") coefficient, μ_s, and kinetic friction coefficients, μ_k, are empirically determined. As these coefficients decrease we observe significant deterioration in computation of linear position in the horizontal plane, (x, y). This is a direct result of the increase in slippage. The angular states,

Table 2. RMS Odometry Error over Different Ground Conditions at slow speed (each row representing data averaged over 5 runs with distance, ν and number of strides, υ, shown in reference)

Ground	State					Reference	
	x (cm)	y (cm)	z (cm)	α (deg)	β (deg)	ν (cm)	υ
	avg (std)	avg (std)	avg (std)	avg (std)	avg(std)	avg(std)	avg(std)
Cardboard ($\mu_s = 0.65$)	0.42 (0.13)	4.53 (0.80)	0.30 (0.08)	0.78 (0.09)	0.73 (0.06)	84.0 (6.2)	8.4 (0.5)
Plastic ($\mu_s = 0.33$)	0.40 (0.08)	4.45 (0.82)	0.51 (0.05)	0.83 (0.05)	0.65 (0.06)	79.0 (6.2)	8.0 (0.6)
Plastic w/ wetsoap ($\mu_s = 0.20$)	1.39 (0.22)	9.72 (1.30)	0.45 (0.07)	0.63 (0.07)	1.12 (0.22)	73.8 (5.6)	7.4 (0.5)
Plastic w/ drysoap ($\mu_s = 0.07$)	1.82 (0.28)	9.40 (0.56)	0.42 (0.07)	0.63 (0.09)	0.65 (0.05)	78.6 (3.1)	8.2 (0.4)

(α, β), are not affected as severely since they are invariant under the COM translation resulting from uniform leg slippage and their computation only depends on the current single stance phase measurements with no accumulated odometry error. In general, excluding the specially prepared slippery surfaces detailed below, the algorithm performs well and consistently within normal ground conditions characterized by μ_s = 0.33 – 0.65, within which range lie most of the common surfaces when contacted by the rubber toe of robot's leg.

Table 3 compares our leg strain based odometry estimates with sensorless schemes by reference to discrepancies with GTMS measurements of elapsed distance as well as with a wheeled implementation of the Robot pictured in Figure 1(lower left). With no sensing apart from motor shaft measurements, "blind odometry" estimates result from counting the number of leg cycles and multiplying by a previously calibrated "distance-per-cycle" constant. Of course, this is the traditional approach to odometry in wheeled vehicles as well. We ran calibration tests for RHex and a wheeled implementation of the Robot, counting the number of motor shaft cycles over the same long flat "slipless" surface to get the best possible conversion constant. The table presents discrepancies, $\kappa(\%)$ (=$|\Delta\nu|/\nu$), as a percentage of the GTMS measured elapsed distance, ν, for each of the three odometry methods: sensorless legged; pose-based legged; and sensorless wheeled . The results show that the leg strain based odometry from body pose measurements is greatly superior to the blind predictions of the open loop scheme, by nearly an order of magnitude at the higher speeds where the inaccuracies of double support have less effect. The dynamical nature of legged walking (even in the absence of an aerial phase, RHex's gaits exhibit a significant interchange of body kinetic and leg spring potential energy in stance) causes speed variations during locomotion that incur significantly more slippage (exacerbated at slower gaits by prolonged double support) than the far smoother ride afforded by wheels. Thus, our sensor based legged odometry is significantly less accurate than the blind results of counting motor shaft revolutions on the wheeled version of the same machine.

4 Conclusion

We have introduced odometry measurements from a full body pose estimator for a walking hexapod robot based on the kinematic configuration of its legs. We have

Table 3. RHex vs. Wheeled RHex at Varying Speeds and over Different Ground Conditions

	Legged RHex					Wheeled RHex	
	GTMS	sensorless		pose-based		GTMS	
	ν (cm) avg (std)	$\lvert\Delta\nu_d\rvert$ (cm) avg (std)	κ_d (%) avg (std)	$\lvert\Delta\nu_l\rvert$ (cm) avg (std)	κ_l (%) avg (std)	ν_w (cm) avg (std)	κ_w (%) avg (std)
at Varying Speed on Cardboard							
Slow (0.25 m/s)	84.0 (6.2)	17.9 (0.7)	21.4 (2.0)	5.1 (1.9)	6.1 (0.7)	87.6 (3.3)	0.3 (0.09)
Medium (0.35 m/s)	83.8 (8.0)	17.6 (1.8)	21.1 (1.0)	1.8 (0.8)	2.2 (1.0)	83.3 (1.8)	0.3 (0.07)
Fast (0.51 m/s)	84.8 (6.5)	13.9 (3.1)	16.3 (2.9)	1.5 (0.6)	1.7 (0.6)	80.8 (0.9)	0.5 (0.26)
over Different Ground at Slow Speed							
Cardboard	84.0 (6.2)	17.9 (2.2)	21.4 (2.0)	5.1 (1.9)	6.1 (0.7)	87.6 (3.3)	0.3 (0.09)
Plastic	79.0 (6.2)	18.0 (1.9)	22.8 (1.8)	5.3 (3.4)	6.6 (4.4)	87.6 (5.2)	0.3 (0.08)
Plastic w/ wet soap	73.8 (5.6)	16.0 (0.6)	21.8 (1.5)	6.0 (0.4)	8.2 (0.7)	90.2 (4.0)	1.7 (0.22)
Plastic w/ dry soap	78.6 (3.1)	20.9 (2.5)	26.5 (2.9)	14.0 (0.7)	17.8 (0.6)	93.2 (5.1)	1.2 (0.43)

implemented this algorithm on the robot RHex [2] and used a separate visual ground truth measurement system to evaluate the performance at various walking speeds and conditions of surface friction, as well to compare that performance with a wheeled implemented version of the robot. The estimator is shown to perform well at all speeds over normal ground conditions — achieving, for example, mean error in angular states less than 1 degree and lateral and vertical positions less than 1 cm compared to the robot size (50cm x 25cm x 15cm), and five times more accurate legged odometry than computed from averaged open loop distance-per-stride estimates — close to the performance of wheeled vehicles. The estimator continues to function well over a variety of ground conditions, with the onset of significant performance degradation on the most slippery surfaces (soaped plastic) whose coefficient of friction is less than a third that of normal linoleum.

The odometry computation algorithm as described here cannot function if the operating regime includes aerial phases such as those dynamical gaits studied in [18]. To remedy this shortcoming our future work will introduce other sensor modalities such as linear accelerometers and rotational rate gyroscopes to complement the leg kinematic configuration sensor. The more elaborate sensor suite will not only allow us to perform pose estimation during aerial phases but also enable us to detect slippage, the primary source of error in the present scheme, and correct measurements accordingly.

Acknowledgment

The authors would like to thank Prof. R. Brent Gillespie, Prof. Arthur D. Kuo, Dr. Johann Borenstein from University of Michigan, and Prof. Mark R. Cutkosky from Stanford University for their valuable advice. The authors would also like to thank Dr. Gregory C. Sharp for his visual tracking system implementation and Dr. Richard Groff for his suggestions leading to the planarity measure. We would also like to thank Ben Wegbreit from the International Foundation of Robotics Research for the “IFRR Student Fellowship for ISER 2004” to cover the travel expense of P.-C. Lin. This work is supported by DARPA/SPAWAR Contract N66001-00-C-8026 and N66001-03-C-8045.

References

1. M. Buehler, U. Saranli, and D. E. Koditschek, “Single actuator per leg robotic hexapod,” U.S. Patent 6,481,513, 2002.
2. U. Saranli, M. Buehler, and D. E. Koditschek, “Rhex - a simple and highly mobile hexapod robot,” *International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.
3. U. Saranli and D. E. Koditschek, “Template based control of hexapedal running,” in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, 2003, pp. 1374–1379.
4. P.-C. Lin, H. Komsuoglu, and D. E. Koditschek, “A leg configuration sensory system for dynamical body state estimates in a hexapod robot,” in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, 2003, pp. 1391–1396.
5. H. H. S. Liu and G. K. H. Pang, “Accelerometer for mobile robot positioning,” *IEEE Trans. Industry Application*, vol. 37, no. 3, pp. 812–819, 2001.
6. M. C. Kim and W. K. Chung, “Posture estimation of a car-like mobile robot using disturbance conditions,” *Advanced Robotics*, vol. 13, no. 2, pp. 189–202, 1999.
7. H. Chung, L. Ojeda, and J. Borenstein, “Accurate mobile robot dead-rockoning with a precision-calibrated fiber-optic gyroscope,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 1, pp. 80–84, 2001.
8. Y. Ma, J. Kosecka, and S. S. Sastry, “Vision guided navigation for a nonholonomic mobile robot,” *IEEE Trans. Robot. Automat.*, vol. 15, no. 3, pp. 521–536, Jun 1999.
9. L. Moreno, J. M. Armingol, S. Garrido, A. D. L. Escalera, and M. A. Salichs, “A genetic algorithm for mobile robot localization using ultrasonic sensors,” *Journal of Intelligent and Robotic Systems*, vol. 34, no. 2, pp. 135–154, 2002.
10. O. Wijk and H. I. Christensen, “Triangulation-based fusion of sonar data with application in robot pose tracking,” *IEEE Trans. Robot. Automat.*, vol. 16, no. 6, pp. 740–752, 2000.
11. P. Goel, S. I. Roumeliotis, and G. S. Sukhatme, “Robust localization using relative and absolute position estimate,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 2, 1999, pp. 1134–1140.
12. P. Gorce, “Dynamic postural control method for biped in unknown environment,” *IEEE Trans. Robot. Automat.*, vol. 29, no. 6, pp. 616–626, 1999.
13. H. Yoshida, K. Inoue, and Y. Mae, “Mobile manipulation of humanoid robots - optimal posture for generating large force based on statics,” in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 3, 2002, pp. 2271–2276.
14. E. Martinez, “Qualitative vision for the guidance of legged robots in unstructured environments,” *Pattern Recognition*, vol. 34, no. 8, pp. 1585–1599, August 2001.
15. G. A. Lopes and D. E. Koditschek, “Visual registration and navigation using planar features,” in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 3, 2003, pp. 3935–3940.
16. K. Hosoda, M. Kamado, and M. Asada, “Vision-based servoing control for legged robots,” in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 4, 1997, pp. 3154–3159.
17. R. P. Paul, *Robot Manipulators - Mathematics, Programming, and Control*. The MIT Press, 1983.
18. U. Saranli, “Dynamic locomotion in a hexapod robot,” Ph.D. dissertation, Univerisity of Michigan, August 2002.

Free-Climbing with a Multi-Use Robot

Tim Bretl[1], Stephen Rock[1], Jean-Claude Latombe[2], Brett Kennedy[3], and Hrand Aghazarian[3]

[1] Department of Aeronautics and Astronautics, Stanford University
{tbretl,rock}@stanford.edu
[2] Computer Science Department, Stanford University
latombe@cs.stanford.edu
[3] Jet Propulsion Laboratory, California Institute of Technology
{bkennedy,hrand}@helios.jpl.nasa.gov

Abstract. This paper presents a new four-limbed robot, LEMUR IIb (Legged Excursion Mechanical Utility Rover), that can free-climb vertical rock surfaces. This robot was designed to have a number of capabilities in addition to climbing (e.g., assembly, inspection, maintenance, transport, intervention) and to be able to traverse a variety of other types of terrain (e.g., roads, talus, dirt, urban rubble). To maximize its flexibility in this regard, LEMUR IIb will need to exploit sophisticated control, planning, and sensing techniques in order to climb, rather than rely on specific hardware modifications. In particular, this paper describes a new algorithm for planning safe one-step climbing moves, which has already enabled LEMUR IIb to climb an indoor, near-vertical surface with small, arbitrarily distributed, natural features. To the authors' knowledge, this is the first experimental demonstration of a multi-use, multi-limbed robot climbing such terrain using only friction at contact points (i.e., free-climbing).

1 Introduction

Various types of robots that climb vertical surfaces have been created previously. These include adhesive robots that "stick" to a featureless, flat or smoothly curved surface by using specific end-effectors (e.g., suction cups and pads [8,16,18,21,24], or magnets [9,10]), robots whose end-effectors match engineered features of the environment (e.g. pegs [4], peg-holes [23], fences or porous materials [25], handrails or bars [2,3], and poles [1,19]), and robots designed to climb within pipes and ducts [17,20,26]. Each of these robots was designed for a particular vertical environment, and relies on its specific hardware design in order to climb.

We focus instead on enabling multi-use robots of more general hardware design to climb. We consider robots with a small number of articulated limbs. We do not distinguish between limbs, and call the end-point of each one a *hand*. To climb vertical terrain, the robot must go through a continuous sequence of configurations satisfying certain constraints (e.g., equilibrium, collision, joint-torque limits). At each configuration, some of the robot's hands are in contact with the terrain – a surface with small, arbitrarily distributed

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 449–458, 2006.

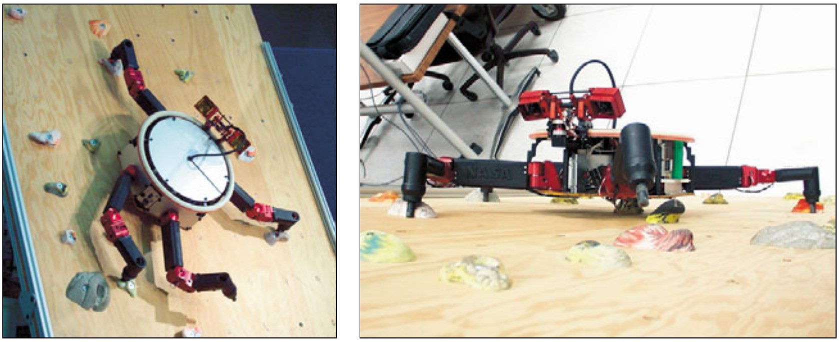


Fig. 1. LEMUR IIb, a multi-use robot capable of free-climbing.

features (e.g., protrusions or holes) called *holds*. During a *one-step* motion, the robot brings one hand to a new hold while using frictional contacts at other hands and internal degrees of freedom (DOF's) to maintain equilibrium. A *multi-step* motion is a sequence of one-step motions.

Our motivation for this approach is the ultimate development of flexible, intervention-capable, multi-limbed robots that can navigate through many different types of terrain. Potential applications include search-and-rescue, surveillance, personal assistance, and planetary exploration. The design of these robots is still critical; however, problems of motion and manipulation in specific environments are addressed by control, planning, and sensing techniques rather than hardware modifications.

In particular, our recent work has focused on the problem of careful foot-placement and trajectory generation for multi-limbed robots, which is necessary on steep, irregular terrain. Other works that have addressed the foot-placement problem make assumptions not valid for climbing (e.g., massless legs, frictionless surfaces, strictly horizontal foot-placements) [5,13,15]. Previously, we presented a fast planner to compute one-step climbing moves for multi-limbed robots, and demonstrated this planner in simulation [7]. Here, we apply our planner to enable a real, multi-use robot (LEMUR IIb) to climb a near-vertical, artificial rock surface. Our experimental results demonstrate the feasibility of free-climbing with such a robot, and have a number of implications for future development.

2 Experimental setup

2.1 Robot and terrain

LEMUR IIb consists of four identical limbs attached to a circular chassis, with a total mass of 7 kg (Figs. 1 and 2). Each limb contains three revolute joints, providing two in-plane (yaw) and one out-of-plane (pitch) degrees

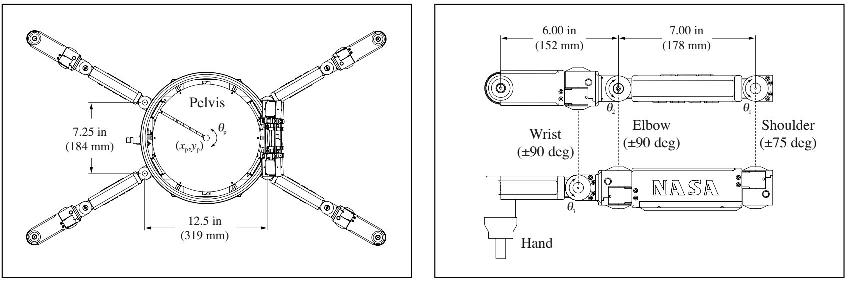


Fig. 2. A schematic diagram of LEMUR IIb, showing dimensions and notation.

of freedom (DOF's). Each joint has an identical drive-train, capable of a maximum continuous torque of 5.0 N-m and a maximum speed of 45 deg/s. Each end-effector is a single peg wrapped in high-friction rubber. LEMUR IIb can be field-operated, with on-board batteries, processing (using a PC104 architecture), and sensors (including a swiveling stereo camera pair, a 6-axis force/torque sensor at each shoulder, a 3-axis accelerometer, and joint angle encoders). When untethered, the robot's symmetry allows it to climb at an arbitrary orientation.

The terrain climbed by LEMUR IIb in our tests is an indoor, near-vertical, planar surface. This surface is covered with small, artificial rock features (holds) exactly as are indoor "climbing gyms" for human climbers. These holds are of arbitrary size and shape. The robot only uses friction to keep contact with holds – this requires careful placement of its center of mass.

2.2 Problem statement

In each experiment, LEMUR IIb is initially placed at an arbitrary, statically-stable configuration on the climbing surface. The robot is then commanded to grasp a particular, distant hold. Typically, this goal hold will be unreachable while maintaining its initial set of contact points, so the robot will have to make a multi-step climbing motion. The challenge is to complete such a motion (autonomously) without falling.

2.3 Scope and limitations

In order to focus on the one-step planning algorithm, we make several simplifying assumptions. First, the location and friction characteristics of each hold are identified manually and pre-surveyed. Also, in our current implementation we maintain LEMUR IIb's chassis parallel to and at a fixed distance from the climbing surface, and use the out-of-plane DOF in each limb only to make or break contact with features. We do not exploit momentum or dynamic movements – at this stage in our research, the robot's motion is slow enough to be assumed quasi-static. We allow the robot to contact the terrain only with its hands (i.e., no "whole-arm" manipulation). Finally, although each one-step motion is planned autonomously (see Section 3), in the experiments described in this paper the user must provide each one-step motion goal (i.e., which hold to grab or release next) along the multi-step path.

There are two other current limitations of our hardware system. First, although joint-angle encoders allow measurement of the robot's internal configuration, the vision sensors are not yet able to provide the location of the robot relative to holds (i.e., global motion is executed in open-loop). Second, for convenience, position-based control rather than hybrid force-motion control (e.g., as in [12]) is currently used to control the robot's configuration (i.e., no attempt is made to sense or control contact forces). We are in the process of correcting these two limitations; in the meantime, the early success of our climbing experiments is a testament both to the importance and effectiveness of our planning algorithm, and to the quality of the existing position-based control system.

3 One-step motion planning

3.1 Model and notation

We call the robot's circular chassis the *pelvis*. In each limb, the first joint (nearest the pelvis) is called the *shoulder*, the second joint is called the *elbow*, and the third (out-of-plane) joint is called the *wrist*. Assuming that the pelvis moves at a fixed distance parallel to the wall, any configuration of the robot is defined by 15 parameters: the position/orientation (x_p, y_p, θ_p) of the pelvis and the joint angles $(\theta_1, \theta_2, \theta_3)$ of each limb (Fig. 2).

Any point on the terrain is a potential contact – either on the contour of a continuous rock feature, or on the planar climbing surface. We assume that a discrete number of useful contacts have been identified; these points are called *holds*. For LEMUR IIb, all holds lie on an inclined plane but can have arbitrary orientation, so each is defined by a 2-D point (x_i, y_i) and a 3-D direction $\boldsymbol{\nu}_i$. Holds at which hands are in contact are the *supporting holds*. We model friction at contacts by Coulomb's law.

When climbing, LEMUR IIb always maintains either three or four supporting holds. The set of supporting holds is a *stance*, denoted σ – to differentiate between 3-hold and 4-hold stances, we write $\sigma 3$ and $\sigma 4$. The linkage between the supporting holds – containing the pelvis and either three or four limbs – is called the *contact chain*. When only three supporting holds are used, the fourth limb is the *free limb*.

Because of the closed-chain constraint, the robot's continuous motion with four supporting holds occurs on a 3-D manifold $C_{\sigma 4}$ in the robot's configuration space. With three supporting holds, motion occurs on a 6-D manifold $C_{\sigma 3}$. This motion is subject to four additional constraints: quasi-static equilibrium, joint angle limits, joint torque limits, and collision. The *feasible space* at a stance σ is the subset F_σ of C_σ satisfying each of these constraints. The limbs have non-negligible mass, so their motion affects the robot's equilibrium. If two points in F_σ are connected by a continuous path in F_σ, we say they are in the same *component* of F_σ. (Henceforth, a "continuous path" will always be taken to mean a continuous motion of the robot at some fixed stance, i.e., with fixed supporting holds.)

To climb upward, the robot must switch between 3-hold and 4-hold stances. Two stances $\sigma 3$ and $\sigma 4$ are *adjacent* if $\sigma 4 = \sigma 3 \cup \{i\}$ for some hold i. The robot can only switch between adjacent stances σ and σ' (i.e., place or remove a hand) at points $q_t \in F_\sigma \cap F_{\sigma'}$. We call such points *transition points*. Given a start configuration $q_s \in F_\sigma$ at a stance σ, we say that a component of the feasible space $F_{\sigma'}$ at an adjacent stance σ' is *reachable* if there is a continuous path connecting q_s to a transition point in that component. This path is a *one-step motion*. Examples of one-step motions are shown in Figs. 3 and 4.

In the experiments described in this paper, the user specifies each one-step motion goal, i.e., for each stance σ, the user specifies a desired adjacent stance σ'. The *one-step planning problem* is to determine whether this stance σ' is reachable, and if so, to construct a continuous path to reach it.

3.2 Algorithm

As described above, the robot's motion takes place in either a 3-D or 6-D space subject to multiple constraints. Since many one-step motion queries will be made along a multi-step path, it is computationally impractical to determine exactly whether an adjacent stance σ' is reachable from each start configuration $q_s \in F_\sigma$. Therefore, we use an approximate method: first, we sample transition points $q_t \in F_\sigma \cap F_{\sigma'}$, then we try to construct a continuous path from q_s to each q_t.

Assume that $\sigma' = \sigma \cup \{i\}$ for some hold i, so σ is a stance with three supporting holds. Then to sample q_t, we search for points in $F_{\sigma'}$ – a 3-D space – and check that these points are also in F_σ. Note that if $q \in F_{\sigma'}$, then we can verify $q \in F_\sigma$ only by checking the equilibrium constraint at stance σ. Therefore, feasible q_t can be found very quickly. Instead, it is the path-planning problem that dominates computation time.

Because of its flexibility and speed, the Probabilistic-RoadMap (PRM) approach, or one of its variants, is widely used for this type of problem [11]. We use a lazy, bi-directional PRM (as in [22]), exploiting the efficient check of quasi-static equilibrium with arbitrary frictional contacts we described in [7]. However, these planners tend to lose efficiency when the feasible subset of the configuration space contains narrow passages or is subject to closed-chain constraints, as these features are difficult to sample. In [7], we showed that equilibrium constraints for simple climbing robots create narrow passages that occur in a low-dimensional subspace, which can be sampled separately to speed up planning. We have used these results to generate efficient PRM sampling strategies for LEMUR IIb, allowing fast one-step planning.

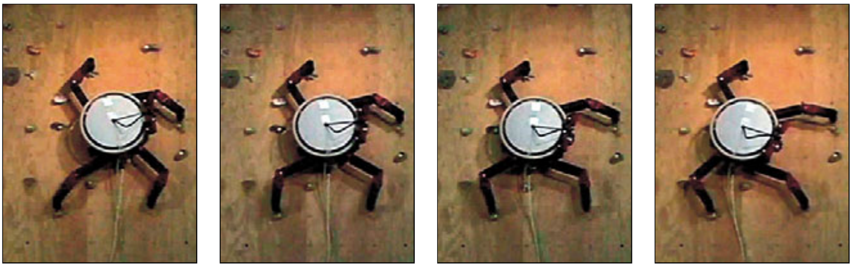


Fig. 3. A one-step motion with a 4-hold stance, to remove the bottom right hand.

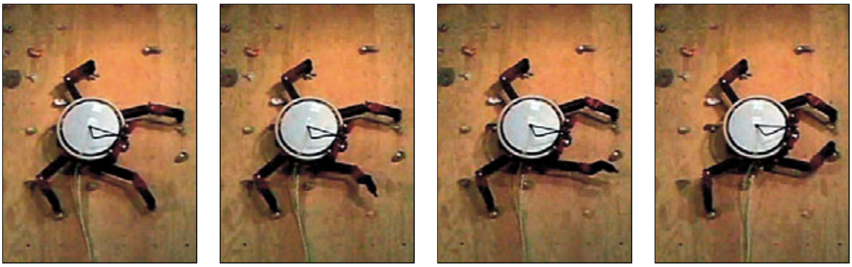


Fig. 4. A one-step motion with a 3-hold stance, to place the bottom right hand.

These heuristics primarily consist of two modifications to the algorithm presented in [7]. First, configurations of the robot with $\theta_2 = 0$ in any limb of the contact chain (straight-limb configurations) are initially sampled explicitly, since the feasible space F_σ is most likely to be disconnected on these manifolds. During subsequent exploration of F_σ, we only attempt to connect those pairs of configurations with identical elbow-bends (i.e., when θ_2 in each limb of the contact chain has identical sign for both configurations).

Second, at stances with three supporting holds, we add several deterministic configurations corresponding to each sampled one. Given a sampled configuration of the contact chain, we explicitly calculate the configurations of the free limb that bring the robot closest to infeasibility with respect to the equilibrium constraint (in general, there are two). If these points are in F_σ, they are added to the roadmap as well. This strategy tends to approximate the analytical decomposition technique described in [7]. (Note the interesting similarity between our heuristics and those based on "manipulability" [14].)

4 Experimental Results

4.1 Summary

We applied our one-step planner to generate climbing motions for LEMUR IIb, which were subsequently executed by the real robot. Snapshots from one

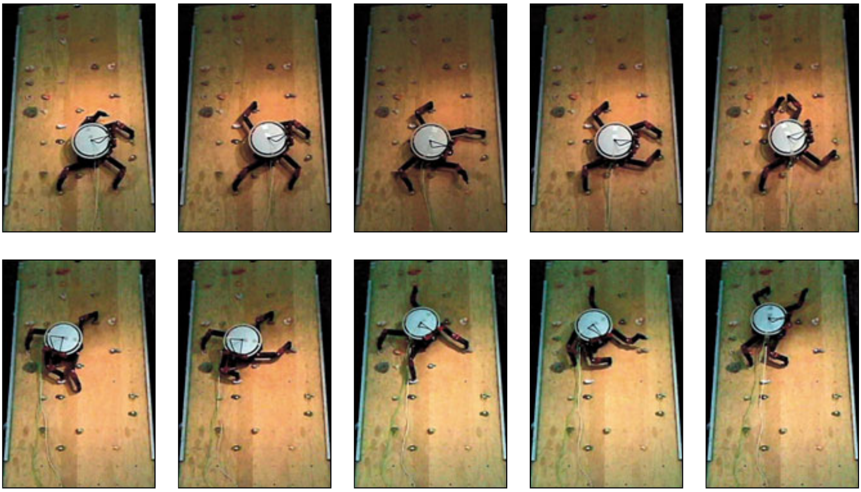


Fig. 5. Snapshots of LEMUR IIb climbing a near-vertical surface covered with artificial rock features.

climb (see also a video at `http://arl.stanford.edu/~tbretl`), taking the robot from bottom to top of the climbing surface, are shown in Fig. 5. In this example, the one-step planner generated, on-line, each of 88 one-step motion trajectories forming the multi-step path. Our experiments demonstrate the feasibility of autonomous free-climbing with a multi-limbed robot, given sophisticated planning and a basic level of control and sensing. Even using tele-operation, these experiments would have been difficult or even impossible without our planner – one-step motions are very challenging to construct by hand, due to the interplay between equilibrium and joint-angle constraints.

4.2 Implications for multi-step planning

Requirements. In addition to demonstrating the usefulness of autonomous one-step planning for free-climbing robots, our experiments clearly show a need for autonomous multi-step planning as well.

For example, it took one of the authors two full days to design the terrain and a sequence of feasible one-step motion goals for the 88-move path shown in Fig. 5, despite the fact that he is a human rock-climber. The reason is that multi-step planning is a hard combinatorial problem, even given a fast one-step planner. There are over 20000 feasible stances for LEMUR IIb in the terrain shown, populating a graph search of moderate breadth and high depth. (The situation becomes even worse if holds are sampled from a continuous environment rather than specified and pre-surveyed.) One might think that because the author, an experienced human climber, is skilled at constructing multi-step paths for himself, and because there is an obvious

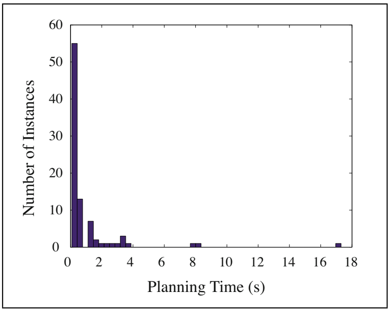


Fig. 6. The distribution of one-step planning times along a multi-step path consisting of 88 one-step motions. Minimum planning time was 0.09 s, maximum was 17.3 s. Mean planning time was 1.02 s, but over 75% of the one-step motions were computed quicker than average.

similarity between LEMUR IIb's motion and his own (also see [7]), he would have an advantage when planning for the robot. However, it is difficult to map climbing motions across morphologies; joint-angle constraints and the effect of body position on the center of mass are both much different.

Likewise, consider the large number of steps needed for LEMUR IIb to climb a relatively short distance: the robot has a reach of 1 m, but to climb a distance of only 2 m (as in Fig. 5) took 88 moves. The shortest possible path with the same multi-step goal consists of about 76 moves, still quite large. In fact, it generally takes 4-12 one-step motions for the robot even to climb to the limit of its tactile sensor range, given an initial stance. This suggests that multi-step planning, rather than reactive motion, is required.

Approach. Our experiments also have strongly influenced our current work in designing an autonomous multi-step planner. For example, the distribution of planning times for the multi-step path in Fig. 5 is shown in Fig. 6. Most one-step moves were planned very quickly (less than 1.0 s on a 1GHz PowerPC). However, several difficult moves took more time (more than 5.0 s), even as much as 17.3 s. This reveals an important issue when searching for multi-step plans, in which many more potential one-step motions are explored than ultimately used. Since PRM planners lack a formal stopping criterion, how much time T_{max} should be spent on each one-step motion query before it is declared infeasible? In this example, suppose $T_{max} = 2.0\ s$ (twice the mean value) – infeasible one-step motions are rejected quickly, but several difficult moves (15% of the multi-step path) likely are not found. Alternatively, suppose $T_{max} = 20.0\ s$ (greater than the maximum value) – now, all feasible one-step motions likely are found, but every infeasible query takes ten times longer, drastically increasing total search time. A similar problem exists for non-gaited motion of humanoid robots and for manipulation planning. We are currently investigating several possible solutions to this problem (see [6]).

4.3 Other lessons learned

Several other lessons were learned as a result of our experiments, that seem obvious in hindsight. For example, most experimental failures occurred when the robot's hands (rigid pegs) rolled along hold contours. Resulting cumulative errors led either to missed hand placements or to torque overloads from jamming. This problem could be addressed in software, but a hardware solution involving passive, articulated hand endpoints (rather than rigid pegs) should prove more practical.

Additional issues raised include future control and sensing requirements (e.g., the need for hybrid force-motion control, the integration of tactile sensing, or the required precision and range of visual sensors), hardware modifications (e.g., whether skewed rather than symmetric joint-angle limits might increase mobility), and user interface design (e.g., how to communicate risk, particularly when one-step motions are "almost feasible").

5 Conclusion

This paper presented a new multi-limbed, multi-use robot, LEMUR IIb. The ability of this robot to free-climb vertical rock surfaces, using a previously presented one-step planning algorithm, was experimentally demonstrated. These experiments revealed a number of additional issues, many concerned with the need for and requirements of an autonomous multi-step planner. Many other challenges remain to be addressed (e.g., integration of local visual and tactile sensing, implementation of hybrid force-motion control, and consideration of dynamic motion).

Acknowledgments. T. Bretl is supported by a Herbert Kunzel Fellowship. The authors would like to thank T. Miller, D. Black-Schaffer, and K. Hauser for their contributions.

References

1. M. Almonacid, R. Saltarén, R. Aracil, and O. Reinoso. Motion planning of a climbing parallel robot. *IEEE Tr. on Rob. and Aut.*, 19(3):485–489, 2003.
2. H. Amano, K. Osuka, and T.-J. Tarn. Development of vertically moving robot with gripping handrails for fire fighting. In *IEEE/RSJ Int. Conf. on Int. Rob. and Sys.*, Maui, HI, 2001.
3. C. Balaguer, A. Giménez, J. Pastor, V. Padrón, and M. Abderrahim. A climbing autonomous robot for inspection applications in 3d complex environments. *Robotica*, 18:287–297, 2000.
4. D. Bevly, S. Dubowsky, and C. Mavroidis. A simplified cartesian-computed torque controller for highly geared systems and its application to an experimental climbing robot. *ASME J. of Dynamic Systems, Measurement, and Control*, 122(1):27–32, 2000.
5. J.-D. Boissonnat, O. Devillers, and S. Lazard. Motion planning of legged robots. *SIAM J. on Computing*, 30(1):218–246, 2000.
6. T. Bretl, S. Lall, J.-C. Latombe, and S. Rock. Multi-step motion planning for free-climbing robots. To appear in *WAFR*, Zeist, The Netherlands, 2004.
7. T. Bretl, J.-C. Latombe, and S. Rock. Toward autonomous free-climbing robots. In *Int. Symp. of Rob. Res.*, Siena, Italy, 2003.

8. H. Dulimarta and R. L. Tummala. Design and control of miniature climbing robots with nonholonomic constraints. In *World Congress on Intelligent Control and Automation*, Shanghai, P.R.China, Jun 2002.
9. J. C. Grieco, M. Prieto, M. Armada, and P. G. de Santos. A six-legged climbing robot for high payloads. In *IEEE Int. Conf. on Cont. App.*, Trieste, Italy, 1998.
10. S. Hirose, A. Nagabuko, and R. Toyama. Machine that can walk and climb on floors, walls, and ceilings. In *ICAR*, pages 753–758, Pisa, Italy, 1991.
11. L. E. Kavraki, P. Svetska, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Tr. on Rob. and Aut.*, 12(4):566–580, 1996.
12. O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. of Rob. and Aut.*, RA-3(1), 1987.
13. J. J. Kuffner, Jr., K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue. Motion planning for humanoid robots under obstacle and dynamic balance constraints. In *IEEE Int. Conf. on Rob. and Aut.*, 2001.
14. P. Leven and S. Hutchinson. Using manipulability to bias sampling during the construction of probabilistic roadmaps. *IEEE Tr. on Rob. and Aut.*, 19(6):1020–1026, 2003.
15. A. Madhani and S. Dubowsky. The force workspace: A tool for the design and motion planning of multi-limb robotic systems. *ASME Journal of Mechanical Design*, 119(2):218–224, 1997.
16. A. Nagakubo and S. Hirose. Walking and running of the quadruped wall-climbing robot. In *IEEE Int. Conf. on Rob. and Aut.*, volume 2, pages 1005–1012, 1994.
17. W. Neubauer. A spider-like robot that climbs vertically in ducts or pipes. In *Int. Conf. on Int. Rob. and Sys.*, pages 1178–1185, Munich, Germany, 1994.
18. M. Rachkov. Control of climbing robot for rough surfaces. In *Int. Workshop on Robot Motion and Control*, pages 101–105, 2002.
19. Z. Ripin, T. B. Soon, A. Abdullah, and Z. Samad. Development of a low-cost modular pole climbing robot. In *TENCON*, volume 1, pages 196–200, Kuala Lumpur, Malaysia, 2000.
20. T. Roßmann and F. Pfeiffer. Control of an eight legged pipe crawling robot. In *Int. Symp. on Experimental Robotics*, pages 353–346, 1997.
21. S. W. Ryu, J. J. Park, S. M. Ryew, and H. R. Choi. Self-contained wall-climbing robot with closed link mechanism. In *IEEE/RSJ Int. Conf. on Int. Rob. and Sys.*, Maui, HI, 2001.
22. G. Sánchez and J.-C. Latombe. On delaying collision checking in PRM planning: Application to multi-robot coordination. *Int. J. of Rob. Res.*, 21(1):5–26, 2002.
23. Y. Xu, H. Brown, M. Friendman, and T. Kanade. Control system of the self-mobile space manipulator. *IEEE Tr. on Cont. Sys. Tech.*, 2(3):207–219, 1994.
24. W. Yan, L. Shuliang, X. Dianguo, Z. Yanzheng, S. Hao, and G. Xuesban. Development and application of wall-climbing robots. In *IEEE Int. Conf. on Rob. and Aut.*, Detroit, MI, 1999.
25. M. Yim, S. Homans, and K. Roufas. Climbing with snake-robots. In *IFAC Workshop on Mobile Robot Technology*, Jejudo, Korea, 2001.
26. A. Zagler and F. Pfeiffer. MORITZ a pipe crawler for tube junctions. In *IEEE Int. Conf. on Rob. and Aut.*, pages 2954–2959, Taipei, Taiwan, 2003.

Limbless Conforming Gaits with Modular Robots

Mark Yim, Craig Eldershaw, Ying Zhang, David Duff

Palo Alto Research Center
3333 Coyote Hill Rd, Palo Alto, CA 94304, USA
{yim, celdersh, yzhang, dduff} @parc.com

Abstract: This paper presents experimentation using the PolyBot modular robot of two limbless gaits which conform to environment. A conforming loop gait is low profile and traverses over a variety of obstacles where the ratio of the height of the obstacle to robot is up to 1.3. A concertina snake gait is capable of negotiating narrow passages (for example a width of double the width of the robot itself), including those with bends as sharp as 110º. It is well suited to locomotion in unstructured tunnels. One little appreciated point that is emphasized in this paper is that the difficulty of an environment is always relative to size of the robot itself.

1 Introduction

Mobility in unstructured environments is difficult, yet important for tasks like search and rescue. While a variety of methods for mobility exist, these largely attempt to defeat the obstacles, rather than making use of them. This work focuses on the notion of conforming to obstacles as way to enable locomotion through unstructured environments. Two sensor-based gaits demonstrate the effectiveness of this approach. Conforming to obstacles allows a system to increase stability while maintaining a low profile. This is important in unstable cluttered environments like a rubble pile resulting from a collapsed building.

The snake-like body shape is well-suited for moving through unstructured environments as its small cross-section allows the body to move through narrow gaps, yet its long reach allows it to cross over larger obstacles. Robot snakes however, have yet to be shown to be useful in moving through unstructured environments [1-5]. The body-following-head type gaits (e.g. lateral undulation) are useful for going through narrow passages since the robot never occupies a space much larger than the cross-section. However, most robots which use this gait have employed passive wheels to provide differential friction on smooth flat terrain; these do not work on rough terrain.

This paper presents gaits which make non-sliding contact with the terrain and conforms to it. This is demonstrated through two types of gaits; a conforming loop that rolls over rough terrain and a concertina snake gait for motion through winding tunnels.

The intuition for the conforming loop is very simple: somewhat like a blob rolling over terrain. The gait was introduced in [6] showing the implementation and control, but not any quantitative data on the performance. The concertina snake gait is a slightly more complicated gait. It is typically used by snakes inside tunnels, where the snake wedges the back part of his body against the tunnel walls and extends the front part, then wedges the front part and repeats.

Hirose has demonstrated sensor based conforming locomotion through a winding planar channel using lateral undulation [1]. The snake robot in this case had wheels to reduce friction and moved through a smooth tunnel that was very nearly the same width as the robot. Hirose’s work is somewhat similar to the concertina gait presented here in that it conforms to a tunnel. However, the fundamental difference is that his gait works best where no friction exists, whereas the conforming gaits shown here in fact *rely* upon strong frictional interactions with terrain. In addition, it is not clear how Hirose’s system would fare in 3D unstructured environments, or those environments with rough surfaces (which is the topic of this paper).

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 459–468, 2006.

This paper is organized in a parallel fashion where the conforming loop and concertina gaits are presented in parallel in each section. Section 2 presents the technical approach of each gait, along with an introduction to the modular robotic platform used. Section 3 introduces the experiments performed. Section 4 presents the results of the experiments with some analysis. Final conclusions with implications for future work are presented in Section 5.

2 Technical Approach

The PolyBot G1v4 and PolyBot G1v5 [7] modules form the robots that demonstrate the conforming loop and concertina gaits respectively. The PolyBot robot systems consists of many identical one degree of freedom modules that are about 5 cm long. Both G1v4 and G1v5 modules contain a touch sensor, a PIC 16F877 microprocessor, a hobby servo and an NiMH battery pack and are illustrated in Figures 1a and 1b. The two modules are slightly different in that the G1v4 modules are more easily reconfigured as they are assembled by a simple press and twist mechanism, where as the G1v5 modules are more robustly held together with screws.

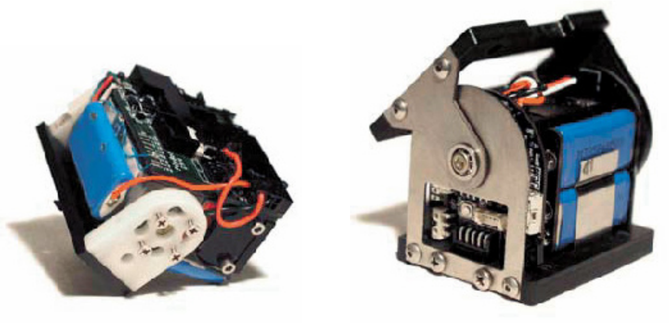


Fig. 1 a) PolyBot G1v4 module, **b)** PolyBot G1v5

Each module can be attached to any other module with 90º rotational symmetry. Thus modules may be attached together such that their axes of rotation are parallel and they move in a plane, or they maybe attached such that their axes are perpendicular to obtain out of plane motions. The G1v4 modules have four connection plates versus the two connection plates of G1v5, however, only the end two connection plates are used in these demonstrations.

While every module has onboard computation and ability to hold batteries, these demonstrations use a tether simply because it is easier to implement, and the focus of this work is locomotion. Tetherless and standalone operation have been demonstrated for other gaits using PolyBot and would not be difficult to implement.

Conforming Loop

The implementation of the *conforming loop* configuration consists of a chain of 16 G1v4 modules attached such that all the axes of rotation are parallel. The two ends of the chain are linked together to form a loop as in Figure 2. This loop can “roll” like a tank tread through a series of coordinated changes in the joint angles of the modules. Each of these G1v4 modules has force sensors attached on the face outside of the loop. These are FSR (force sensing resistor) sensors, and their analog outputs are simply thresholded to detect contact with the environment.

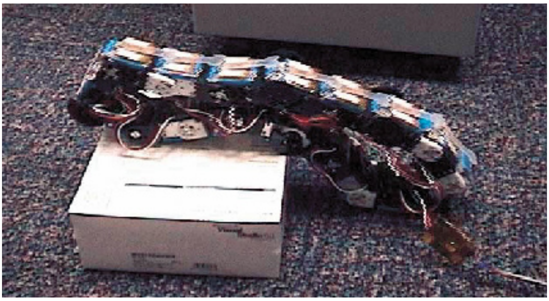


Fig. 2: 16 G1v4 modules in a loop configuration conforming over a box

As the loop rolls forward, the touch sensors on the forward part of the robot detect contact with the environment, and can thus react by stopping those moving modules. This results in the loop conforming to the shape of the environment as it rolls over it.

There are essentially two processes running in parallel. One process causes the back end of the loop to form two 90º bends so the chain folds over on top of itself (see Figure 3). By propagating this fold (straightening one 90º bend and bending a straight module) the back end of the robot moves forward as if it were a track rolling. The front end does the same thing, however the bend and unbend are done in a sensor-based fashion. The front end module continuously moves until contact is detected with the FSR. If contact is made early, (for instance the front end hitting an obstacle) the unbending module stops and the next module starts to unbend causing the front end to "climb up" the obstacle (see Figure 4a). By the same token if no obstacle is there and the ground drops away (the edge of a cliff) the unbending module continues to move past the straight position into a negative bend (see Figure 4b).

This gait can be implemented in a distributed manner with local processes and message passing. The phase automata programming paradigm is a natural one for implementing the conforming loop and is described in detail in [6].

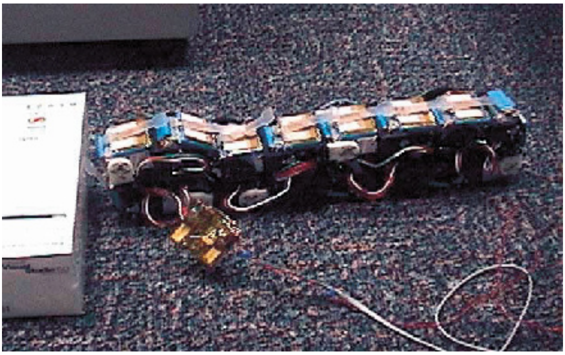


Fig. 3: Conforming loop rolling on flat terrain

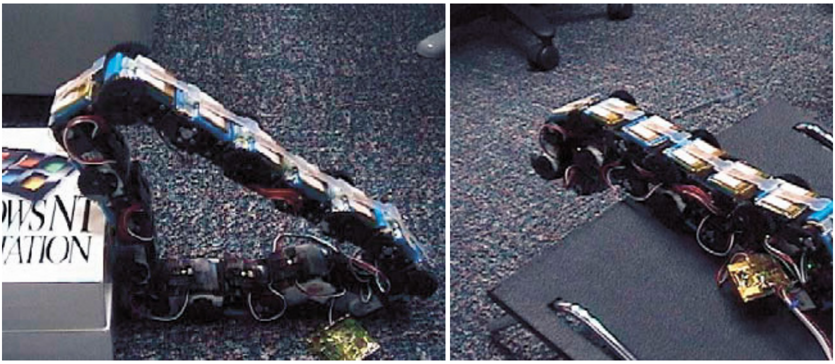


Fig. 4: **a)** Conforming loop climbing up an obstacle. **b)** A negative bend over the edge of an obstacle

Concertina Gait

The *concertina* gait is one of several snake locomotion gaits. To the authors knowledge, the demonstrations by PolyBot are the first to be been demonstrated by a robot, especially in unstructured terrain. Snakes use the concertina gait to move through parallel constrained areas such as tubes or tunnels. Essentially, the snake "bunches" itself, pushing against the sides of the environment; then straightens itself starting from the front end, making some forward motion. As implemented on PolyBot, the robot bunches until contact with the obstacles (e.g. side walls) is sensed. This action not only engages the environment but is used to conform the gross shape of the robot to the shape of the tunnel. Since the robot "grabs" the walls by pushing against it, a rough surface is beneficial to the gait. This combined with the conforming nature of the gait makes it particularly apt for moving through unstructured tunnels for example, those found built by animals in hard earth.

While a variety of touch sensors were tested, a more indirect method of contact sensing turns out to be more effective. The problem with touch sensors in snake-like robots is that it is difficult to ensure the entire surface of the robot is sensitive. In unstructured environments an outcropping may make contact with the robot outside of the touch sensitive area. Instead of using touch sensors the module servos are commanded to a position and the position sensors are checked to see if the module has reached the commanded position, relying on the compliance inherent in the modules. By taking many small steps contact with side walls of a tunnel may be detected with contact on any portion of the robot. Using this method a "guarded move" primitive is the basic component for this gait. A module is commanded to move to a final position taking many small steps until it either reaches the target position or makes solid contact with an object stopping its motion. This method of contact sensing is not very sensitive and so larger side forces are obtained during the concertina gait. This leads to larger normal forces and thus larger beneficial frictional during the "wedging" portion of the gait.

For this implementation the PolyBot modules are arranged by the following. Let a chain of modules be represented by a chain of letters X and Y, with X being modules oriented such that the axis of rotation is parallel with gravity and Y such that the axis is perpendicular. Figure 5 depicts the chain of PolyBot G1V5 modules used for the concertina gait in an XXYXXXYXX configuration. The X modules perform the concertina motion. The Y modules allow out of plane motion and conform to the up/down bending of a tunnel. A lighted camera module at the front shows images of the inside of the tunnels. In addition, for the bending trials a triangular wire “nose” is mounted onto the camera housing in the front of the module to help passively guide the robot.

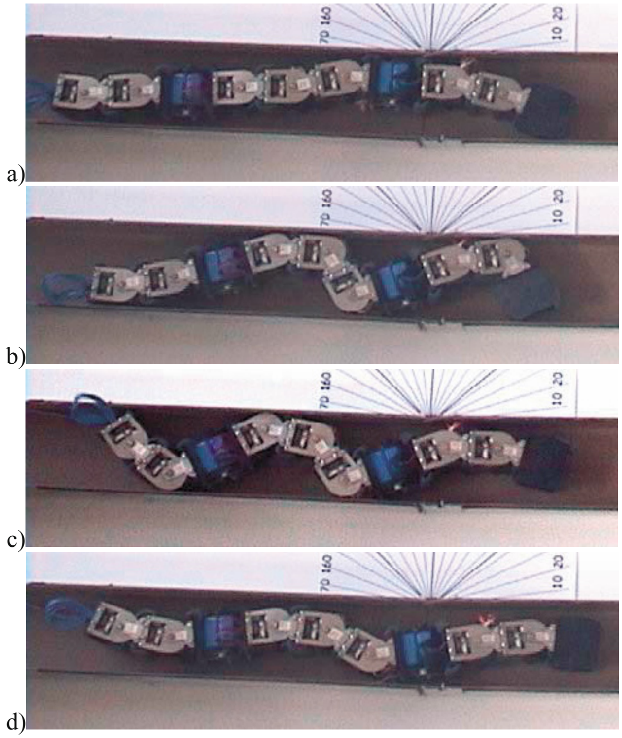


Fig. 5: Nine PolyBot G1v5 modules comprise the concertina gait configuration. a) the starting position, b) bending starting from first the front part of the body, c) bending the tail last with the front unbending first, d) unbending continues.

A gait control table is used to generate the concertina motion pattern. Gait tables are described in detail in [2]. This table assigns a sequence of target angles to each module. The one difference in the gait table implemented in this paper is that each move in the table is a guarded move, rather than an open-loop linearly interpolated motion.

The sequence of motions start with the robot straight and all modules compliant (see Figure 5).

- The front two modules bend to the right until contact is made or until the modules form an arc of 140 degrees which ever comes first (as in Figure 5b).
- Then the next two modules do the same, but bend to the left (as in Figure 5b).
- The next module bends to the right
- And finally the last two modules bend to the left again (as in Figure 5c).
- The modules then straighten in the same order that they bent (front first).

We call this sequence a *right bend cycle* of the gait. A l*eft hand cycle* starts with the front modules bending to the left and alternating right and left similarly in a mirror image to the right bend cycle. During the concertina gait through a straight tunnel, the system is shown alternating between right bend and left bend cycles.

The principle here is that the front end bends first grabbing the side walls so as the rest of the body bends the body gets pulled towards the front. Once fully compressed, the modules sequentially unbend with the back end as the last part to unbend. Thus, the back end remains in contact with the walls as the front end extends out forward again making progress in the desired direction.

By allowing portions of the robot to be compliant in addition to the guarded moves, the robot can explore non-straight paths while using one fixed gait table. That is the robot's global shape can conform to the turns in a path without explicitly controlling the robot to do so, just relying on guarded move to stop the modules at the appropriate time.

3 Experiments

A variety of experiments were implemented for both the conforming loop gait and the concertina gait. Obstacles of varying height and width composed the test environment for the conforming loop while paths with varying bend angles composed the tests for the concertina gait.

Conforming Loop Gait

One measure of performance of the conforming loop is the ability of the leading edge to handle different obstacle configurations. There are two classes of obstacles in the conformal loop tests: a *step up* and *step down*. A variety of objects such as boxes and books make up the step up and step down obstacles of differing heights. These obstacles were broad enough that the center of gravity of the robot is on top of the obstacle before the front end starts to climb down the "step down" part of the obstacle.

In addition several other obstacles were tested. Figure 6 shows a platter that has 4cm high walls that are several millimeter thick. The robot had no problems climbing over these walls the thin walls served more as a feature to grab than as obstacles. In addition, a composition of step up, step down, and ramp obstacles was shown and reported in [8].

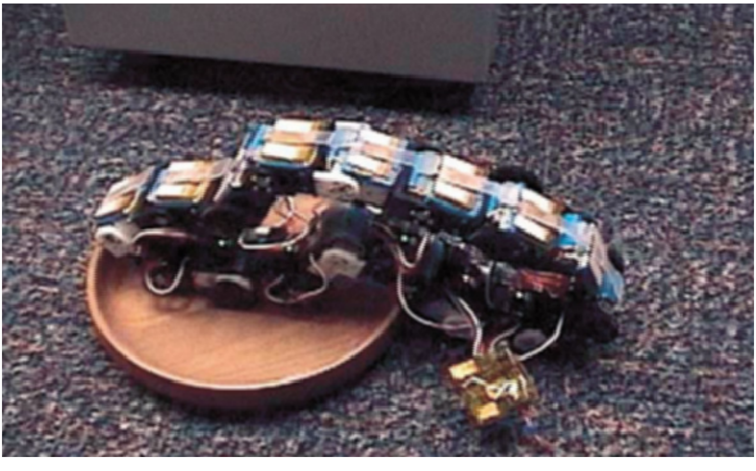


Fig. 6: Conforming loop crossing over a platter

Concertina Gait

The test environment to see how the concertina gait progresses through different bends in the path consists of a straight cardboard u-beam with a partial cut through the middle as shown in Figure 7. The channel is approximately double the width of the modules. Tests are performed with different bends in the beam with additional pieces of cardboard attached at the seam to ease the transition at the bend.

In addition to measuring success or failure, the time to move through the bend is recorded. Timers are started with the head of the robot at the bend, and stopped when the tail of the robot just moves past the bend.

For the bend tests, a slightly different pattern of motion is used than the straight channel. Two more cycles are added to the original left and right bend cycles and then rotated among the four types of cycles. These new cycles are similar to the two mentioned earlier, except that the front three modules bend one way rather than the front two. One called *extended left bend cycle* has the front three bend left and one called *extended right bend cycle* has the front three bend right. This allows for the nose of the module to bend more to insert and follow a bending path. The rotating among the four cycles performs similarly for the straight tunnels as the rotation among the previous two cycles.

Just like the conformal loop, the tests run with the concertina gait are not always consistent, although multiple trials are not completed at this time. Failure is recorded if the system ever enters the same state multiple times (i.e. no progress is being made).

The grounds outside of the laboratories at PARC where multiple ground squirrel burrows pose an annoyance to the local grounds keepers is shown in Figure 8. A hole serves as a test path. Features of this irregular tunnel include an ovoidal shape about 15cm × 10cm, a curve downward at the beginning, then a curve upward and to the left.

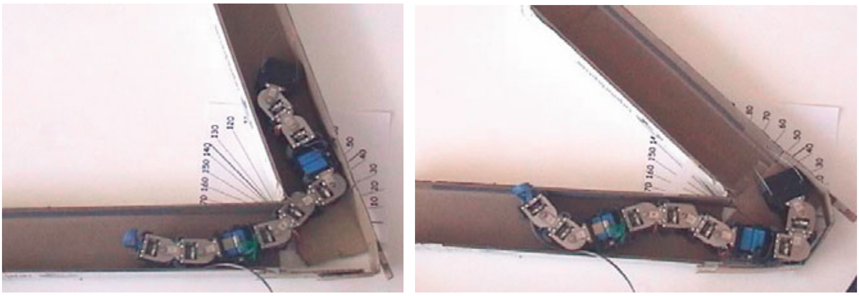


Fig. 7: a) Concertina gait test path with a bend of 110º. **b)** test path with 145º.

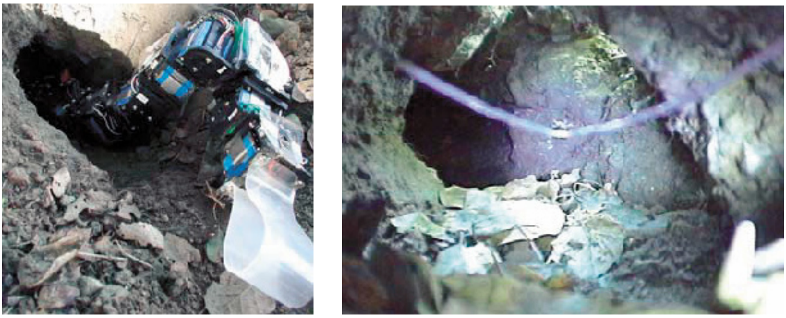


Fig. 8: a) Concertina gait moving through ground squirrel hole. **b)** Inside of the hole as seen by robot’s camera

4 Results

Closed loop conforming loop

To surmount a step with the conforming loop gait, the system must reach the top of the step as well as transfer its center of gravity to the point over the step. Experimentally, the maximum step that was climbed with a 16 module loop was 13cm in height which is approximately one sixth of the length of 16 modules end-to-end, about one third the length of the loop. Real snakes are purported to be able to reach up one third of their length.

Table 1 shows the robot's performance over varying heights for step up obstacles. The 13 cm obstacle appears to be the borderline maximum height achievable with this method. The 18 cm obstacle was not achieved despite multiple attempts. The failure mode for this gait while attempting to climb over a too high obstacle occurs with the robot falling over backwards. The front end starts to build up (as in Figure 4a) until the base of the robot is too small and the robot tumbles.

Table 1: Test runs for conforming loop over obstacles.

Height	Relative height	# success / # trials
9 cm	1.0	3 / 3
13 cm	1.3	2 / 3
18 cm	2.0	0 / 5

For the step down obstacles, the robot seemed to perform adequately. It did not necessarily conform to the corner accurately, however for traversing the step down obstacles, it is only important to make contact with a support surface before the robot falls.

Open loop conforming loop

A goal for the conforming loop would be to climb stairs which are typically 18 to 21 cm tall. In [2], a loop was shown to climb stairs with a rise of 18 cm. The difference was that the system used hand tweaked open loop gait to achieve a consistent robust stair climbing behavior for one set of stairs. In addition it used 17 instead of 16 modules.

It is possible that the addition of a single module makes the difference between success and failure for stair climbing, however it is just one module that contributes to a 6% increase of the circumference of the loop. In any case, the open loop stair climbing is an empirical upper bound that is an achievable maximum height for conforming loop gait.

One reason that the conforming loop does not perform as well as the hand-tweaked stair climber is that the conformed shape does not match as well as the hand tweaked one. For example in Figure 4a, the bottom part of the configuration can be seen to be "wrinkled" where as the analogous part of the stair climber in Figure 9 shows a more precise shape.

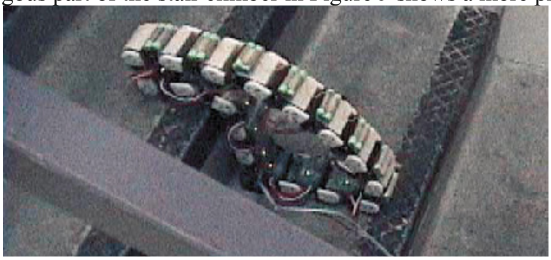


Fig. 9: Stair climbing robot, run open loop.

Closed loop concertina

Table 2 shows the robots performance through a variety of bends and the time it took for the rear of the module to pass the bend. Motions through paths that have bends less than 90º would to be expected to succeed as any compliant forward motion would work (imagine pushing a wire through a tube). However, motions through angles greater than 90º are not as obvious.

The system worked well at 100º and even 115º, moving even quicker than through the 85º bend. In the case of the sharp angles, certain cycle motions would catch the corner of the bend on a module and then propel the snake nearly 4 cm down the tunnel with the one cycle. Normally each bend cycle makes approximately 2.5 cm of progress taking 10 seconds to execute. The robot executes approximately 20 to 40 cycles for each trial.

The system failed at the 145º bend shown in Figure 7b. The front of the modules bunched up in the corner and did not get out of it. It is possible that an increased bend like the extended bend cycles with 4 instead of 3 modules would work, but this was not tried.

During execution of the gait through a bend, there are 3 critical points where a motion cycle may get stuck or slow progress: the front end in the initial bend, and each Y module as it passes the corner. If the configuration consisted of all X orientations, two of the critical points would be missing and the speed through the bends would likely be higher and more consistent. However, the system would likely not perform well in non-planar tunnels such as those constructed by rodents.

The concertina gait also successfully traversed into the ground squirrel hole approximately two meters. A video camera and lights in the nose of the robot showed the progress into the hole, though no ground squirrels were found.

Table 2: Test runs for concertina gait through a single bend in tunnels

Angle in degrees	Time in min’ sec”
0	3’25”
45	3’55”
85	6’30”
100	4’00”
115	4’30”
145	failure

The concertina gait is a notoriously slow and inefficient gait as used in nature and appears to be so for robots as well. A combined concertina, rectilinear (inchworm-like) motion was also tested with a XYXYXYXY configuration. A cycle of rectilinear motion was introduced between right and left bend cycles. The rectilinear motion increased the speed of the locomotion and improved the “catching” of the corner for moving through bends. Figure 10 shows the path successfully traversed by this gait.

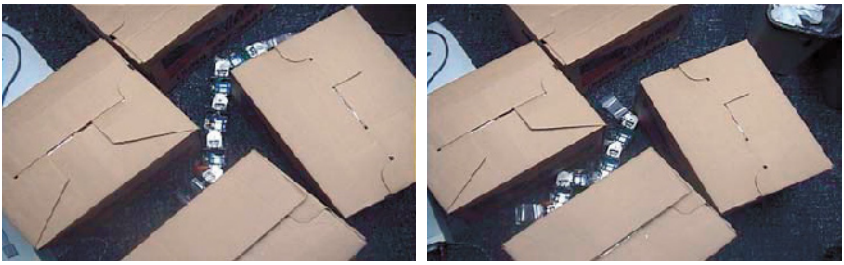


Fig. 10: A zigzag path with an XYXYXYXY configuration using a combination gait.

Open loop concertina

An open loop experiment (without using concertina motion) also uses the path in Figure 9. It uses a rectilinear inchworm-like forward motion with compliant lateral modules in an XYXYXYXY configuration. This control run shows that passive compliance alone does not work. The system gets stuck at the first right angle bend. Even with manual assistance to move the compliant modules to move past the first bend, the system gets stuck at the next right angle bend.

5 Conclusions

Two limbless conforming gaits are presented that have the ability to move through unstructured terrain. One is a conforming loop gait that is low profile (can crawl under 11 cm gaps) yet still climb over obstacles taller than itself. Experiments with the conforming loop gait show the gait is good for climbing over obstacles up to 13 cm, though there is still some room for improving the maximum height as demonstrated by the open loop stair climbing. The conforming loop gait as implemented cannot move out of plane and thus cannot turn. A turning conforming loop would be more useful for realistic applications.

The concertina gait is shown to be able to locomote through bends as sharp as 115º in structured settings and also to be able to maneuver through unstructured holes such as those formed by ground squirrels. This is a first demonstration of such a capability.

Future work includes implementing a conforming loop gait that can turn. Varying the number of modules in a bend may have be more effective over different channel widths. It may be interesting to demonstrate a single robot configuration that can do both the concertina gait through winding tunnels as well as the turning conforming loop.

References

[1] Hirose, S., Biologically Inspired Robots: Snake-like Locomotors and Manipulators, Oxford University Press, 1993, ISBN 0 19 856261 6. (translation of 1987 Japanese edition)

[2] Yim, M., Homans, S. and Roufas, K., "Climbing with Snake-like Robots", *Proc. of IFAC Workshop on Mobile Robot Technology*, Jejudo, Korea, May 2001

[3] Paap, K.L., Dehlwisch, M., Klaassen, B., "GMD-Snake: A Semi-Autonomous Snake-like Robot," *3rd International Symposium on Distributed Autonomous Robot Systems* (DARS 96), October, 29-31, 1996, RIKEN, Saitama, Japan.

[4] Dowling, K. *Limbless Locomotion: Learning to Crawl with a Snake Robot* doctoral dissertation, tech. report CMU-RI-TR-97-48, Robotics Institute, Carnegie Mellon University, Dec., 1997.

[5] Burdick, J., Radford, J., Chirikjian, G., "A 'sidewinding' locomotion gait for hyper-redundant robots," *Advanced Robotics*, Vol. 9, No. 3, pp.195-216, 1995.

[6] Yim, M., Roufas, K., Duff. D., Zhang, Y., Eldershaw, C. and Homans, S. "Modular Reconfigurable Robots in Space Applications", *Autonomous Robot Journal, Special Issue on Robots in Space,* Vol.14, No,2/3 March/May 2003

[7] Yim, M., Duff, D., and Roufas, K., "PolyBot: a Modular Reconfigurable Robot", *Proc. of IEEE Intl. Conf. on Robotics and Automation* (ICRA), San Francisco, CA, April 2000

[8] Greene, H. W. *Snakes: The Evolution of Mystery in Nature*. Berkeley, University of California Press., 1997

XII. Vision

Vision remains as the most verbose sensor for robots with the potential and providing the most information. This chapter presents different methods of integrating vision for the navigation of mobile vehicles.

The goal for the first paper in this chapter, by Pradalier, Bessiere and Laugier, is to build a robust outdoor autonomous vehicle. The vehicle is to navigate outdoor near built up area, so it needs to rely on on-board sensors than the GPS. In the second paper, Usher and co-workers at CSIRO present a catalog of different navigation capabilities of the robot and a method to select between them. Vision-based competencies form the major part of the catalog.

The third paper by McCarthy and Barnes focuses on the study of *Optical Flow* method in the navigation of mobile robots. The *optical flow* is estimated by spatio-temporal filters and the paper presents the comparison of the effects of three different filters. The case study in this paper is that of docking, where the optical flow is not constant, as opposed to that of corridor centering (following) where it is constant.

The last paper in the chapter by Rao, Taylor and Kumar made an interesting study to control unmanned ground vehicles with overhead cameras, with further goal of using camera mounted on aerial vehicles. The idea presents a convenient way of controlling ground vehicle in an unstructured environment. In order to relate the objectives defined in the image plane to the velocity of the vehicle, the relationship between the vehicle's velocity and its motion projection in the image plane was defined. The focus of the paper was the experiment and the validation of the scheme.

Vision-Based Navigational Competencies for a Car-like Vehicle

Kane Usher, Jonathan Roberts, Peter Corke, and Elliot Duff

CSIRO ICT Centre
PO Box 883, Kenmore, Qld 4069, AUSTRALIA
Email: firstname.lastname@csiro.au

Abstract. If mobile robots are to perform useful tasks in the real-world they will require a catalog of fundamental navigation competencies and a means to select between them. In this paper we describe our work on strongly vision-based competencies: road-following, person or vehicle following, pose and position stabilization. Results from experiments on an outdoor autonomous tractor, a car-like vehicle, are presented.

1 Introduction

Reliable navigation of outdoor mobile robots is still a significant challenge. The existence proof of vision-based animal navigation is an inspiration, but an elusive one for engineers. Many researchers are still concentrating on navigating in indoor environments with well defined walls, floors and doors. The outdoor scenario poses many more challenges including a less-structured environment and generally higher vehicle speeds. The particular outdoor environment that we are interested in is the environment around buildings (Figure 1). In this environment, the use of GPS as a

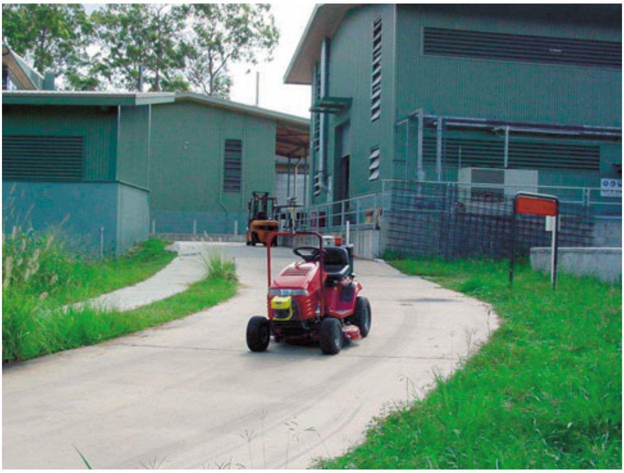


Fig. 1. The application environment with roads in proximity to buildings.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 471–480, 2006.

primary navigation sensor is hindered by the presence of buildings and other items which lie close to desired vehicle paths (e.g. trees near roadways). The ultimate goal of our work is to develop navigation technique(s) that allow a vehicle to autonomously drive around an industrial compound or building site type environment. We also use a vehicle with common car-like kinematics, as opposed to an omnidirectional base.

Our approach is to develop a catalog of fundamental navigation competencies and a means to select between them. In this paper we describe vision- and laser-based road-following, person or vehicle following, pose and position stabilization.

To validate the model of the vehicle and to demonstrate the applicability of the controller, experiments were conducted on our land navigation research platform the Autonomous Tractor and is described in the next section.

2 System Architecture

The CSIRO Autonomous Tractor, (Figure 1), is a ride-on mower which has been retro-fitted with an array of actuators, sensors, and a computer system enabling the implementation and testing of control and navigation algorithms.

The vehicle operates in three modes: *manual*, *remote*, or *automatic*. *Manual* operation is the tractor's original mode of operation in which the vehicle is driven by an operator. *Remote* mode allows a user to control the vehicle from a hand-held radio-transmitter. *Automatic* mode allows the on-board computer to control the vehicle. Six axes of control: speed pedal, throttle lever, brake pedal, park brake lever, steering wheel and steering engage are implemented using HC12-based microcontrollers which are commanded by the main computer. This is a PC104 format Crusoe 800 CPU with a solid-state disk, frame-grabber and WLAN running the Linux operating system. All applications run in our standard robotic control environment [4, 11].

Navigation sensors currently include:

- A color omni-directional camera (EyeSee 360) mounted over the front wheels.
- Crossbow high speed orientation sensor (3 axis accelerometer + 3 axis magnetometer).
- SICK PLS laser scanner, which also serves its original purpose as a proximity barrier and can cut the engine.
- Vehicle speed and steering angle.
- RTK GPS (here used for a ground-truth only).

3 Vision-based competencies

3.1 Road Following

In our previous work, we successfully automated an underground mining vehicle using reactive navigation and opportunistic localization [9, 12]. Central to the reactive navigation was a technique based upon radially constrained active contours [8]

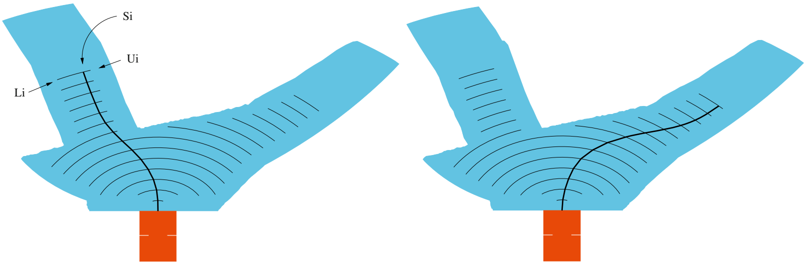


Fig. 2. Radially constrained snake for left or right turn.

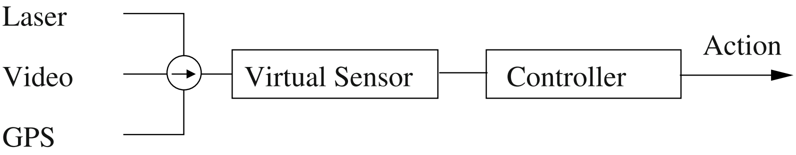


Fig. 3. Switching between different sensors.

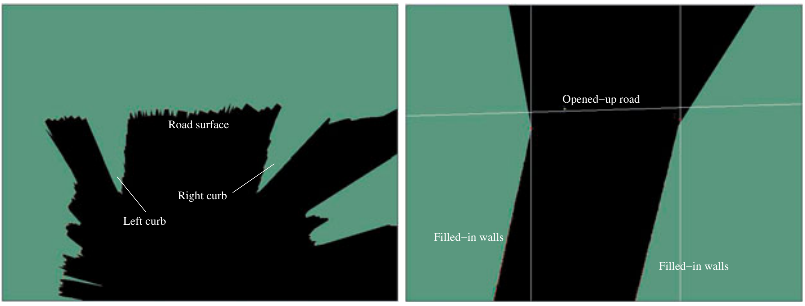


Fig. 4. Raw laser data of road and curbs (left), virtual laser output (right).

(a.k.a snake) inspired by the active contour of [3]. An example, is shown in Figure 2. The primary sensor for the snake was a 2D laser ranger which was mounted on the front of the vehicle so that its scanning plane was parallel to the roadway. This configuration provides a plan-view of the tunnel ahead to the snake. The snake is then used to create a path that avoids the tunnel walls. Over the last few years it has been tested in 100kms of tunnels and on vehicles ranging in size from 30 to 60 tonnes.

In this section we describe how we use a down-tilted laser and an omnidirectional camera to generate laser-scanner like data that we can pass to our existing snake navigation module.

We take the output of different sensors, see Figure 3, and create a virtual tunnel in which to navigate using the method just described. For the controller to cope with the different modes of data, the controller reads its data from a virtual sensor with a predefined data format.

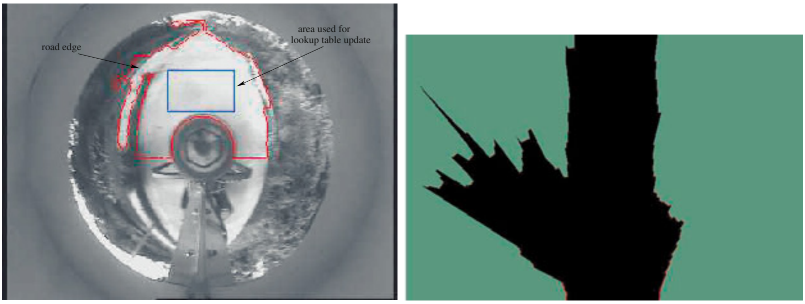


Fig. 5. Segmented omnidirectional image (left), virtual laser output (right).

Laser-based curb following The laser scanner is tilted downwards so the the scanning plane intersects the road at a point approximately 2m ahead of the tractor. Figure 4(left) shows the raw laser scan of a narrow road with a curb on either side. The shallow laser tilt angle accentuates the height of the curbs. The first step is to identify the road itself in the scan. We currently assume the central laser segment (the zero degree or straight ahead point) is on the road (a more sophisticated method is being investigated). The final step is to generate data for a virtual range sensor that is used by the constrained active contour navigation system to generate steering commands. We construct a simple tunnel based on the road-edge points, and opening-up the road ahead (Figure 4(right)). Details are given in [16].

In practice the vehicle can navigate reliably while maintaining a distance of between 2cm and 10cm from the curb.

Robust Vision-Based Road Following Roads come in many shapes and forms and may not always have edges which can be sensed with a laser scanner. Here we use the omnidirectional camera to create a virtual tunnel.

Vision-based road following vehicles are certainly not new with a myriad of existing systems including those of the NAVLAB series of robots at Carnegie Mellon University (see e.g. [13]) and the work of Dickmann's [7]. Similar to the early work in road following of Crisman et al. [6], we use colour histogramming techniques for road/non-road segmentation. However our system is currently much less sophisticated and is probably more related to the carpet tracking type techniques used by some indoor robots (see e.g. [1]). We operate directly on the omnidirectional image (i.e. it is *not* unwarped) and use the YCrCb video signal provided by the framegrabber. More details are provided in [16].

Figure 5 (left) illustrates the results of image segmentation and 'road' identification, while Figure 5 (right) illustrates the resulting conversion to a 'laser scanner' format.

We use a lookup table to segment the color pixels. The lookup table is initialized from a training image in which the user selects the appropriate 'road' regions. The Cr and Cb values of the selected pixels are then added to the colour lookup histogram. This initial histogram acts as a seed for the on-line table generation.

The table is then updated on-line to accommodate changing lighting conditions. Of course, such an on-line update technique has to be used carefully. If the vehicle isn't initially well positioned on the road, it could end up adding the non-road regions to the lookup table which has obvious ramifications. On the other hand, although the techniques used here are relatively simple, they have been found to be quite powerful. Current research is looking towards alternative color models and methods of suitably 'aging' the histogram to allow transitions in road type to occur more smoothly.

Having segmented the 'road regions', we then use chain-code techniques to determine the image position of the road edges with respect to the centre of the omnidirectional image. The next step is the conversion from image polar coordinates to ground-plane polar coordinates using knowledge of the camera-mirror optics and an assumption that the ground is flat and that the road edges lie in the ground-plane [10] (see [14] for an explanation involving equiangular mirror optics as used here). These assumptions are not strict, the main requirement is that the camera-mirror optic axis is perpendicular to the ground-plane, otherwise the conversion skews the resulting ground-plane coordinates.

Finally, resampling of the ground-plane range and bearing from the omnidirectional image merely involves binning the range and bearing measurements from the omnidirectional camera into a laser scanner format. In this case, we simulate a SICK PLS which provides a 180° range scan of the area forward of the vehicle, ensuring that the minimum range for a particular bearing is used.

Experiments The vision-based road following system was tested on a 200 m section of concrete road, bordered by grass and dirt. The vehicle successfully navigated along the centre of the road, negotiating a series of left and right-hand corners and a significant S-bend. The concrete road contains quite a number of 'dirty patches' and shadows, for which the colour of the road significantly changes. Despite the significant difference in road appearance, the vehicle continued to negotiate the road. On reaching the end of the concrete road, the vehicle halts as it has no free space in which to drive.

3.2 Following

A useful behaviour for a tractor type vehicle is that of target following, which could be used for following either another vehicle (convoying) or a person (perhaps for a teach-by-show task). For example, a target following behaviour can be used to guide the tractor around the environment in areas where its other behaviours are unsuitable or when they are known to fail. A target following vision module was developed for our tractor that finds the largest red object in an image.

The tractor was moved around our test site by a person holding a red traffic cone (Figure 6(left)). The tractor continuously steered towards the traffic cone and maintained the area of the target in the image plane at a certain size, which ensures constant-distance following. The tractor was maneuvered in a figure-of-eight path around the test site. Figure 6(right) shows the path of the tractor (measured by GPS).

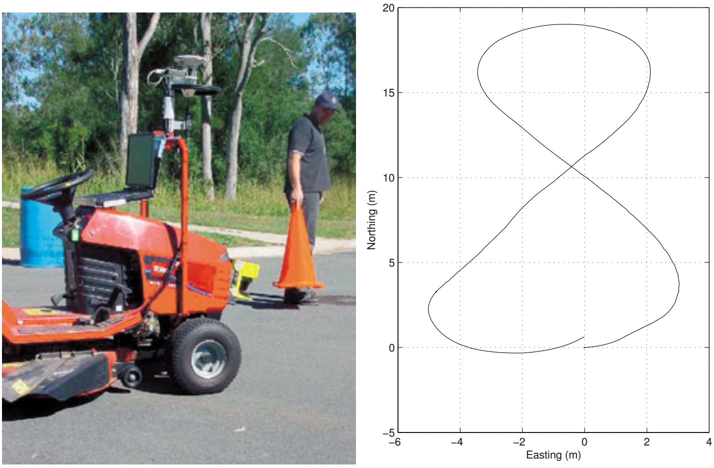


Fig. 6. The tractor following a person with a cone (left). The figure-of-eight path taken by the tractor as it followed the person (right).

3.3 Position and Pose Stabilization

Our previous work has addressed the pose stabilization problem for a car-like vehicle [14] using techniques adapted from the biologically-inspired visual homing literature. We used visually estimated range and bearing to a single landmark and absolute heading direction from a magnetic compass, see [15] for an explanation of the sensing used for pose estimation).

Pose stabilization The pose stabilization strategy relies on a multistage or switching control law in which different control laws are invoked depending on the current state of the vehicle. In summary, pose stabilization proceeds as follows (for more details refer to [14]):

- Stabilize the vehicle to within some tolerance of the goal region (for our vehicle this is within a radius of approximately 5 m of the goal).
- Stabilize the vehicle to the *x*-axis. In reality, this is done to within some tolerance which is empirically found.
- Stabilize the vehicle to $x = 0$. Again, this is done to within some tolerance which is empirically found.

Position stabilization Another useful competency for a mobile robot is position stabilization. For this behaviour, a similar switching control strategy can be used. The kinematic equations for a car-like vehicle in Cartesian space are:

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

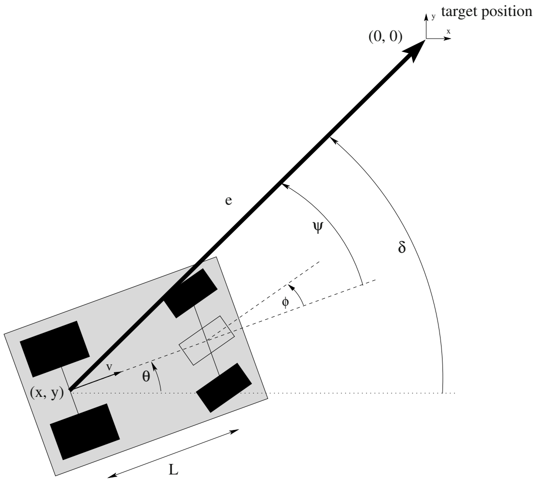


Fig. 7. Coordinate system for control.

where the demands for the system are v and ϕ, and x, y and θ are defined with reference to the target position and the x-axis as shown in Figure 7.

Transforming the Cartesian representation into a vehicle relative target representation, see Figure 7, leads to the following vehicle state equations:

$$\begin{bmatrix} \dot{e} \\ \dot{\psi} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} -v\cos\psi \\ \frac{v\sin\psi}{e} - \omega \\ \frac{v\sin\psi}{e} \end{bmatrix} \tag{2}$$

where e is the distance to the target position, $\psi \in (-\pi, \pi]$ is the bearing to the target position relative to the vehicle's longitudinal axis, $\omega = v\frac{tan\phi}{L}$ and $\delta \in (-\pi, \pi]$ is the bearing to the target position relative to the x-axis. These equations are quite a 'natural' form for a vehicle moving through a workspace and are commonly found in the literature on the pose control of unicycle-like vehicles (see e.g. [2]).

As noted earlier, the control strategy used here is to break the state-space into sub-manifolds, each of which is then sequentially stabilized. To stabilize a car-like vehicle to a point, control proceeds as follows:

- First stage (bearing stabilization): $\psi \to 0$
- Second stage (homing): $\psi \to 0$ and $e \to 0$

To achieve this, the individual controllers of each stage are designed using suitably chosen Lyapunov functions, based upon the well-known Lyapunov stability theory.

Experiments Figure 8 (left) shows a plot of a typical pose stabilization run in which the vehicle stabilizes to a pre-learnt*pose*. This plot shows position as estimated from vision overlayed with results from a simulated run. The pose stabilization behaviour

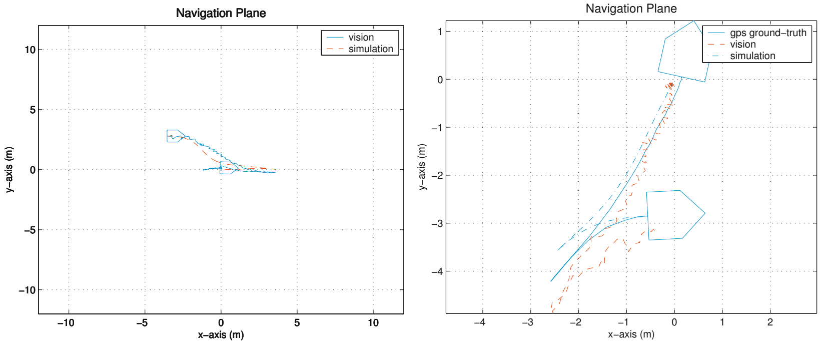


Fig. 8. Left: the tractor stabilizing to a pose. Right: the tractor stabilizing to a position.

is quite similar to that which would be obtained by a human performing a similar task (think of parallel parking of a car).

Figure 8 (right) shows some typical results of a position stabilization run in which the vehicle is stabilized to a pre-learnt *position*. This plot shows position as estimated from vision overlayed with results from a simulated run. Also shown is a GPS ground-truth. Despite the difference between the apparent position (from vision) and the true position (from GPS) when the vehicle is away from the 'home' position, the system still stabilizes to within an acceptable tolerance of the home position. Of course when using such simple sensing strategies, the system is drastically affected if landmarks are moved in the environment or if there are significant misalignments between the camera-mirror optical axis and the ground-plane. This latter matter is of some concern, particularly over undulating terrain, and we are currently investigating more robust sensing strategies.

4 Behaviour switching

The target detection behaviour was combined with the road-following behaviours. Here a red traffic cone was placed behind a pile of timber halfway down a concrete roadway. The tractor automatically followed the roadway until it detected the traffic cone. It then switched behaviours and steered towards the cone until it was close (determined by its area in the image plane) and stopped (Figure 9(left)). The path of the tractor (measured by GPS) is shown in Figure 9(right).

5 Conclusions

We have demonstrated a number of vision-based competencies which form the basis of our navigation competency catalog. We have also demonstrated some simple behaviour switching based on approximate navigation and feature detection. Unlike many laboratory robots this vehicle moves at a reasonable speed (8km/hr) and has car-like kinematics which introduces complexity for tasks such as pose stabilization.

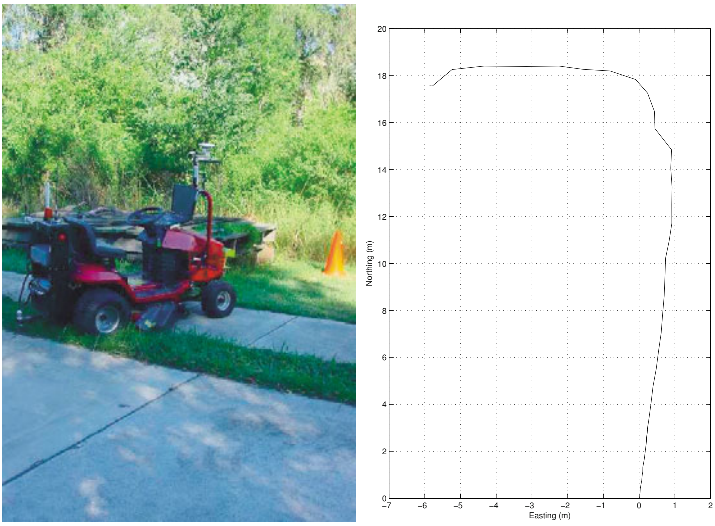


Fig. 9. The tractor turning towards the cone (left). The path taken by the tractor (right).

The system described here is a work in progress, and we are using our busy campus-style laboratory as the test environment. The robot has a high-level of reliability and uptime, clocking many hours of operation per week around the site.

Future work will include the integration with wireless sensor network devices (along the line of [5]), more robust and adaptive colour-based scene segmentation, robust natural feature extraction, more sophisticated behaviour switching rules and more convenient means of expressing the task to the robot.

Acknowledgment

The authors would like to thank the rest of the CSIRO project team: Graeme Winstanley, Matthew Dunbabin, Leslie Overs, Stephen Brosnan, Peter Hynes, Pavan Sikka and Craig Worthington. Thanks also to Jonathon O'Brien of The University of New South Wales for the loan of the Toro ride-on mower.

References

1. Iwan Ulrich abd Illah Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, Texas, USA, 2000.
2. 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.
3. A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.

4. Peter Corke, Pavan Sikka, Jonathan Roberts, and Elliot Duff. Ddx: A distributed architecture for robot control. In *Submitted to 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, October 2004.
5. P.I. Corke. Mobile robot navigation as a planar visual servoing problem. In R.A. Jarvis and A. Zelinsky, editors, *Preprints of 10th Int. Symp. Robotics Research*, pages 217–223, Lorne, December 2001. IFRR.
6. J. Crisman and Charles Thorpe. Scarf: A color vision system that tracks roads and intersections. *IEEE Trans. on Robotics and Automation*, 9(1):49 – 58, February 1993.
7. E. D. Dickmanns, R. Behringer, R. Dickmanns, T. Hildebrant, M. Mauer, F. Thomanek, and J. Shielhlen. The seeing passenger car 'VaMoRs-P'. In *Proceedings of the International Symposium on Intelligent Vehicles*, Paris, 1994.
8. E.S. Duff and J.M. Roberts. Wall-Following with Constrained Active Contours. In *Proceedings of the 4th Internation Conference on Field and Service Robotics*, 2003.
9. E.S. Duff, J.M. Roberts, and P.I. Corke. Automation of an Underground Mining Vehicle using Reactive Navigation and Opportunistic Localization. In *International Conference on Intelligent Robots and Systems*, Las Vegas, US, 2003.
10. I. Horswill. Polly: A vision-based artificial agent. In *Proceedings of the eleventh national conference on artificial intelligence (AAAI'93)*, Washington DC, USA, July 1993. MIT Press.
11. J.M. Roberts, P.I. Corke, R.J. Kirkham, F. Pennerath, and G.J. Winstanley. A real-time software architecture for robotics and automation. In *International Conference on Robotics and Automation*, pages 1158–1163, Detroit, Michigan, 1999. IEEE.
12. J.M. Roberts, E.S. Duff, P.I. Corke, P. Sikka, G.J. Winstanley, and J. Cunningham. Autonomous control of underground mining vehicles using reactive navigation. In *Proceedings of IEEE Int. Conf. on Robotics and Automation*, pages 3790–3795, San Francisco, USA, 2000.
13. Charles E. Thorpe, editor. *Vision and Navigation: The Carnegie Mellon Navlab*. Kluwer Academic Publishers, 1990.
14. Kane Usher, Peter Corke, and Peter Ridley. Landmark-based visual homing: Experiments with a non-holonomic vehicle. In *Proceedings of the Conference on Field and Service Robots*, Tsukuba, Japan, July 2003.
15. Kane Usher, Matthew Dunbabin, Peter Corke, and Peter Ridley. Robust pose estimation for a car-like vehicle. In *International Conference on Robotics and Automation*, New Orleans, USA, 2004. IEEE. to appear.
16. Kane Usher, Jonathan Roberts, Elliot Duff, Matthew Dunbabin, Peter Corke, and Graeme Winstanley. Road following using virtual range sensing and radially constrained active contours. In *Submitted to 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, October 2004.

Comparison of Temporal Filters for Optical Flow Estimation in Continuous Mobile Robot Navigation

Chris McCarthy[1] and Nick Barnes[2]

[1] Dept. Computer Science and Software Engineering
The University of Melbourne
cdmcc@cs.mu.oz.au
http://www.cs.mu.oz.au

[2] Autonomous Systems and Sensing Technologies Programme
National ICT Australia
nick.barnes@nicta.com.au
http://www.nicta.com.au

Abstract. We present our complete study involving comparisons of three spatio-temporal filters used in the estimation of optical flow for continuous mobile robot navigation. Previous comparisons of optical flow and associated techniques have compared performance in terms of accuracy and/or efficiency, and only in isolation. These comparisons are inadequate for addressing applicability to continuous, real-time operation as part of a robot control loop. In recent work [11], we presented a comparison of optical flow techniques for corridor navigation through two biologically inspired behaviours: corridor centring and visual odometry. In that study, flow was predominantly constant. In this paper we give new results from comparisons for flow-divergence based docking, where flow is non-constant. Results for traditionally used Gaussian filters indicate that long latencies significantly impede performance for real-time tasks in the control loop.

1 Introduction

Biologically-inspired visual behaviours such as corridor centring [5], visual odometry[17] and docking[14] have all been demonstrated using visual motion for closed loop control of a mobile robot. Despite encouraging results, mobile robot research has not broadly adopted this paradigm. A perceived lack of robustness, and the absence of any defined systematic approach to the implementation of such behaviours are likely reasons for this. The choice of optical flow method is perhaps the most important case in point. While visual motion is central to these behaviours, no systematic choice of optical flow technique currently exists.

Past comparisons have primarily assessed flow methods on accuracy and/or efficiency, and only in isolation [2,8,10]. These studies have not considered performance when embedded in a system, performing real-time tasks. Quantitative comparisons of accuracy and efficiency alone do not provide sufficient information on which to base a choice for mobile robot navigation. We emphasise the importance of in-system evaluation when comparing optical flow techniques for such an application.

In recent work [11], we conducted a comparison of optical flow techniques for corridor navigation. It was found that the choice of spatio-temporal filter applied

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 481–490, 2006.

with gradient-based methods significantly effected in-system performance. Filters are used to reduce image signal noise before estimating image gradients. Temporal filters differ in required temporal support, latency and accuracy. It is important to consider temporal filters when evaluating flow methods for robot navigation.

In this paper we present our complete study in the comparison of temporal filters for robot navigation, including new results obtained for docking. We focus only on behaviours involving continuous motion, thus excluding such behaviours as hovering where motion changes are necessarily sharp and discontinuous. Corridor navigation comparisons examine performances when flow is constant or near constant. Docking comparisons examine performances when flow is not constant, but exhibiting increasing levels of divergence. The temporal filters for comparison are: Gaussian filtering with central differencing, Simoncelli's matched-pair derivative filters [15], and Fleet and Langley's recursive temporal filter [7]. These are applied with Lucas and Kanade's gradient-based optical flow method [9], chosen on the basis of strong performances in [11]. We give an overview and theoretical comparison of these techniques before setting out our methodology for comparison. We then present results from on and off-board comparisons and then our conclusions.

2 Theoretical Overview

In this section we introduce all techniques and provide theoretical comparisons of the filters for robot navigation. Refer to the cited references for their full details.

Optical Flow Method: Lucas and Kanade[9] This method applies a model of constant velocity on small local neighbourhoods (ω) of the image by minimising:

$$\sum_{\mathbf{x}\in\omega} W^2(\mathbf{x})((\nabla I(\mathbf{x},t)\cdot\mathbf{v}) + I_t(\mathbf{x},t))^2, \tag{1}$$

where $W(\mathbf{x})$ denotes a window function. Thresholding eigenvalues of the least-squares matrix can improve accuracy, however, this was not applied.

Gaussian Filtering Gaussian filtering (std dev 1.5) and central differencing are the traditionally used filters for optical flow estimation. An isotropic Gaussian filter is applied in convolution for spatio-temporal pre-smoothing. A central differencing kernel (typically size 5) is then applied to estimate derivatives. We include two Gaussian filters with standard deviations 1.5 (Gaussian 1.5) and 0.5 (Gaussian 0.5). Central differencing (size 5) is then applied.

Simoncelli's Matched-Pair Filters[15] Simoncelli proposed a filter design for obtaining accurate multi-dimensional derivative estimates using a small low-pass filter and derivative filter. These are related by their simultaneous design and applied as a matched pair through convolution. The implementation used here employs a size three pre-filter before applying the 5-tap matched-pair filters.

Recursive Temporal Filter[7] Fleet and Langley proposed a recursively applied causal temporal filter. Images are filtered via a cascaded implementation of an order n filter, where n is the number of cascades used. A time constant, τ^{-1}, gives the duration of temporal support. We use an order three filter (n=3, τ^{-1}=1.25).

2.1 Theoretical Comparisons

To assist discussion of experimental results, theoretical comparisons are given below.

Accuracy: The Simoncelli filter by its design, is the strongest filter for angular flow accuracy [15]. Gaussian 0.5 accuracy will be low due to increased noise levels. The recursive filter is known to be less accurate on synthetic image sequences than the Gaussian 1.5 filter[7]. Of interest is how accuracy effects on-board performance.

Efficiency: For a 192x144 pixel image sequence, Table 1 shows computation times, storage requirements and latencies for all filters. Of the temporal filters, the Gaussian 1.5 requires the largest explicit frame support and frame delay. Frame delay is likely to influence in-system results significantly.

Robustness: Sensitivity to changing conditions is important, however, robustness to small fluctuations due to robot ego-motion is also desirable. The relatively large temporal support of the Gaussian 1.5, and the implicit support given to the recursive filter, suggests both should be robust to such noise. Simoncelli and Gaussian 0.5 may exhibit higher sensitivity due to their reduced temporal support. Increased noise levels with the Gaussian 0.5 are likely to further impede robustness.

Responsiveness: The recursive filter's large implicit temporal support may inhibit responsiveness to changes. Large frame delay with the Gaussian 1.5 will also impede responsiveness. Simoncelli and Gaussian 0.5 should exhibit high responsiveness given reduced frame support and therefore increased weighting on the current frame.

3 Methodology

In this section we present three navigational behaviours implemented for this comparison: corridor centring, visual odometry and docking. We outline the methodology and performance indicators used in our comparison of temporal filters.

Table 1. Efficiency data for temporal filters. Times taken on an Intel x86 866 MHz machine

Filter	time (ms)	support (frames)	latency (frames)
Gauss 0.5	116	9	4
Gauss 1.5	170	15	7
Recursive	110	3	3
Simoncelli	106	7	3

3.1 Corridor Centring

Corridor centring, inspired by honeybee behaviour [16], is achieved by differencing average flow in the outer thirds of an image from a forward facing camera:

$$\theta = \tau_l - \tau_r, \tag{2}$$

where τ_l and τ_r are the average flow magnitudes in the left and right peripheral views respectively. θ can be directly used for directional control.

Given constant motion and a straight corridor, the flow field response should exhibit consistent average flow magnitude. The robot should be free of short period directional oscillation resulting from noise introduced through the robot's ego-motion. Frequent and current flow updates are needed to maintain behaviour stability. Long period directional oscillation through reduced responsiveness is the likely side effect of such latencies. Off-board comparisons over an image sequence depicting near constant motion of the camera can examine temporal cohesion. On-board trials in a static corridor can demonstrate the level of stability in robot directional control.

3.2 Visual Odometry

Distance travelled can be estimated by accumulating flow in the peripheral regions of the image. At a discrete time t, the visual odometer, d_t, is given by [17]:

$$d_t = \sum^{t} \frac{4}{[\frac{1}{\tau_l} + \frac{1}{\tau_r}]}. \tag{3}$$

Odometry estimates will vary in different environments for the same distance. In the same environment, however, the estimate should be repeatable. To compare methods, variance in average distances travelled can be examined for multiple on-board trials in the same environment. On-board trials are subject to oscillatory directional control, lateral drift, and environmental changes. To account for such in-system influences, off-board performances can also be examined.

A real image sequence with ground truth allows a quantitative comparison. If the distance measure d_t is repeatable, we expect it to differ by only a scale factor, s, from a ground truth visual odometer g_t. Under constant motion, this scale factor should remain approximately constant over time such that:

$$s = \frac{g_t}{d_t} = \frac{g_{t-1}}{d_{t-1}} = .. = \frac{g_1}{d_1}. \tag{4}$$

3.3 Docking

Flow divergence is the measure of image expansion, given by:

$$div = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}, \tag{5}$$

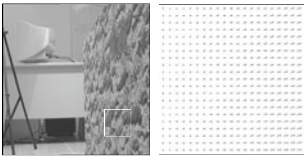


Fig. 1. Sample frame (boxed) and flow field from side wall sequence.

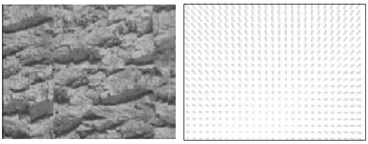


Fig. 2. Sample frame and flow field from looming wall sequence.

where u and v are components of a flow vector in the x and y directions respectively. It has been used extensively in the literature for obtaining time-to-contact estimates[12,4,1]. Docking can be achieved by reducing forward velocity in inverse proportion to increasing flow divergence. To maintain constant flow divergence on approach, forward speed is proportionally reduced.

Typically, divergence is calculated in one or more patches applied at the same image location over time. This assumes the looming surface is fronto-parallel, where in such a case divergence is constant across the image. During docking, minor rotations of the robot will lead to a lateral shift of the focus of expansion (FOE), which would considerably disturb the calculation of divergence if not accounted for. We therefore place divergence patches on a set radius originating from the FOE. The patches are above the FOE, at 45 degrees on either side. An approximately fronto-parallel docking surface is still assumed, however FOE tracking allows for small shifts giving more accurate divergence estimates.

Temporal filters must provide adequate responsiveness to rapidly increasing flow divergence as the wall approaches. Robustness to noise is also important. This can be assessed in off-board comparisons using a looming wall image sequence. On-board docking trials will assess system responsiveness, stability and reliability.

4 Off-board Comparisons and Results

For off-board comparisons, two real, heavily textured image sequences were constructed as shown in Figure 1 and 2. Figure 1 shows a side wall image sequence, depicting the motion of a wall moving 5mm per frame in a near parallel direction to the optical axis of the camera. Figure 2 shows a looming wall sequence. It was constructed by moving a camera 15mm per frame towards an approximately fronto-parallel wall. In both sequences, the visual motion with respect to the camera is subject to small frame-to-frame fluctuations. Ground truth flow fields were generated for all images in both sequences using a calibrated camera and a projective warping technique described in [10]. The model fitting technique RANSAC [6] was used to exclude outlying image correlation points. Comparisons are presented below.

4.1 Corridor Centring

Figure 3 shows average flow magnitudes obtained for each filter across the sequence. No latencies are accounted for, allowing a direct comparison of flow magnitude con-

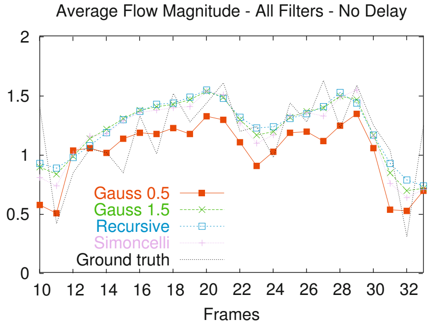


Fig. 3. Average flow magnitudes.

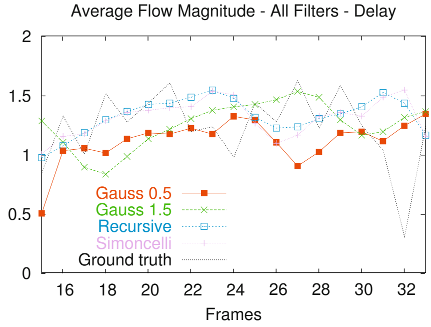


Fig. 4. Average flow magnitudes with latencies.

sistency across the sequence. The recursive and Gaussian 1.5 filters show near identical results. The Simoncelli filter performs slightly worse with sharper fluctuations most evident between frames 19 and 20, and frames 28 to 30. The Gaussian 0.5 exhibited the sharpest fluctuation between frames. Further distinctions are made when their respective frame delays are considered as shown in Figure 4. The larger temporal delay of the Gaussian 1.5 response is clearly evident.

4.2 Visual Odometry

Table 2 shows average scale factor errors ($av(\mathbf{s})$, where $\mathbf{s}$ is defined in (4)) and its variance (σ) for each filter against ground truth. Scale factor errors were calculated using the value of the ground truth visual odometer at each corresponding odometer update. Given constant motion, $\mathbf{s}$ should ideally remain constant over time.

All filters exhibit similar levels of deviation. The last column in Table 2 shows variances of scale factor error when calculated from the straight line approximation to the ground truth visual odometer ($\mathbf{s_{av}}$). The recursive filter shows marginally less variance in $\mathbf{s}$ and $\mathbf{s_{av}}$. The Gaussian 0.5 exhibits the most deviation on both metrics.

4.3 Docking

For off-board docking comparisons, the looming wall sequence was used. Divergence was calculated in two 40x40 pixel patches, centred on a 25 pixel radius from the FOE. Divergence was calculated for each pixel in both patches, then averaged to produce a final output value. Figure 5 shows divergences calculated at each frame, for all filters. All filters, except Gaussian 0.5, produce similar divergence growth.

4.4 Discussion

Across the side wall sequence, all filters except the Gaussian 0.5, exhibited similar consistency. Frame latency is likely to effect Gaussian 1.5 in-system performance. Large temporal support, however, appears advantageous under near constant motion.

Table 2. Odometry error analysis.

Filter	$av(\mathbf{s})$	$\sigma(\mathbf{s})$	$\sigma(\mathbf{s_{av}})$
Gauss 1.5	1.06	0.10	0.12
Gauss 0.5	0.87	0.14	0.14
Recursive	1.07	0.09	0.11
Simoncelli	1.03	0.11	0.13

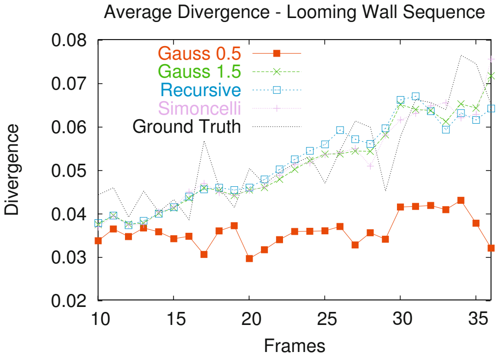


Fig. 5. Divergence over looming sequence.

Visual odometry results give little distinction between filters. All filters appear likely to be reasonable. Some advantage may exist in having large temporal support (i.e. recursive and Gaussian 1.5), thereby reducing sensitivity to noise during motion.

In divergence comparisons, results indicated no lack of responsiveness to divergence increases in the Gaussian 1.5 or recursive filter, despite their large temporal support. With the exception of Gaussian 0.5, differences only become apparent from frame 28 onwards. Consistency appears to diminish in all filters at this point, particularly with the Simoncelli filter. At this point it is likely that flow is beginning to exceed measurable levels.

5 On-board Comparisons and Results

All techniques were integrated into the robot control software, running on an Intel x86 866MHz PC with radio link to a mobile robot. A forward-facing on-board camera was tethered to the PC. Frames were sub-sampled to 192x144 pixels, at a rate of 12.5 frames/sec. Robot tracking was achieved using a calibrated overhead camera. In all comparisons, surfaces were textured as in off-board comparisons.

5.1 Corridor Centring

Trials were conducted for each filter using a straight corridor, 2.5 metres in length, and 0.6 metres wide. Forward velocity was kept constant at $0.15m/s$ and rotation governed by a proportional control scheme. A proportional gain K_p, was chosen for each filter based on multiple trials in a straight and curved corridor. Due to varying filter latencies, a single K_p for all filters was deemed inadequate to accurately gauge relative performances. Figure 6 shows a sample on-board frame and flow field.

Table 3 shows average deviations from corridor center (centring error). The recursive filter achieves the lowest centring error. The Gaussian 1.5 achieves the

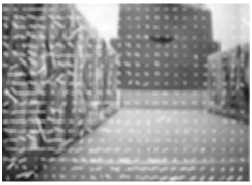


Fig. 6. Sample on-board frame and flow.

Table 3. Centring and odometry results.

Filter	flow updates	centring error	std dev (stop pos)
Gauss 0.5	38	4cm	7cm
Gauss 1.5	35	6cm	10cm
Simoncelli	38	5cm	6cm
Recursive	45	3cm	3cm

highest error. Notably, both Gaussian filters recorded collisions during trials. Figure 7 shows typical path plots for all filters.

5.2 Visual Odometry

On-board visual odometry trials were conducted using the same straight corridor and centring behaviour. For each filter, five trials were conducted. The robot started from the same position each trial, and moved down the corridor until the accumulated visual motion exceeded a preset threshold. The final column of Table 3 shows the variance of stopping positions recorded for each filter. The recursive filter shows significantly lower variance than all others.

5.3 Docking

In docking trials, the robot approached an approximately fronto-parallel wall, attempting to decelerate and safely dock with the wall. Travel distance was 1 metre at an initial speed of 0.4 m/s. Forward velocity control was achieved using:

$$v_t = v_{t-1} + K_p(div_{ref} - div_t), \tag{6}$$

where v_t is the current velocity, K_p is a proportional gain, div_t is the current flow divergence and div_{ref} is the desired divergence which was set to 0.03 for all trials. For each filter, K_p was chosen to be the smallest value for which safe docking could be achieved four times consecutively.

Figure 8 shows typical velocities recorded with all but the Gaussian 0.5 filter for which docking was never achieved. A wide range of K_p values were used to verify this. Stopping distances were found to be similar for all filters. The Gaussian 1.5 exhibits a delayed response compared with other filters. The Simoncelli filter exhibited rapid deceleration, with brief periods of near constant velocity. Notably, no K_p value could be found for which the Simoncelli filter succeeded four times consecutively[1]. The recursive filter shows early response and consistent deceleration.

5.4 Discussion

On-board results suggest the recursive filter is best suited to use in the control loop for continuous motion navigation. This appears a result of large temporal support and low frame delay.

[1] K_p = 0.04 achieved docking three times consecutively and so is shown in Figure 8.

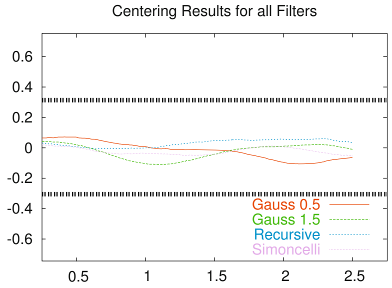


Fig. 7. Best straight corridor results.

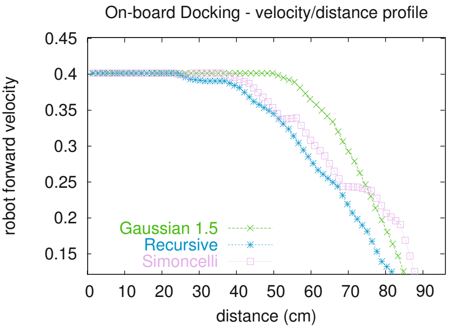


Fig. 8. On-board docking results.

In centring trials, the Gaussian 1.5 filters large frame delay seen in off-board results, appears to cause long phase oscillatory motion when centring. The Gaussian 0.5 centring error results appear reasonable, however several failed attempts were observed. Systematic noise and unresponsiveness appear the main causes of this.

Visual odometry trials exhibited more variation in results than off-board comparisons showed. The recursive filters stability in centring control, and high update frequency are likely reasons for its superior performance. The lack of stability in centring appears the likely reason for the poor performance of the Gaussian 1.5.

Divergence-based docking was reliably achieved using the recursive and Gaussian 1.5 filters. The Simoncelli filter exhibits less stable deceleration. This is possibly due to flow exceeding measurable levels as suggested in off-board results. Reduced temporal support will also heighten sensitivity to noise from large, unmeasurable flow. Future work will investigate this issue further. The Simoncelli filter maybe better suited to navigation under non-continuous motion, such as in hovering. Frame latency in the Gaussian 1.5 appears to have caused delay in divergence response. Where more controlled docking is required, it is likely the Gaussian 1.5 would perform worse than the recursive filter.

6 Conclusion

In this paper, we have presented on and off-board comparisons of temporal filters for gradient-based optical flow estimation in continuous mobile robot navigation. We have emphasised the need for in-system comparisons of vision techniques.

Over all comparisons conducted, the strongest performances were achieved using the recursive filter. Low frame delay and large implicit temporal support appear to be the main reasons for this. This is an encouraging result for the use of gradient-based optical flow in real-time, real-world conditions. Strong performances in corridor centring were also achieved using the Simoncelli filter. The Simoncelli filter appears better suited to navigation under non-continuous motion, such as hovering. Off and on-board results suggested the traditionally used Gaussian 1.5 filter is impeded by high frame delay, causing instability in centring and delayed response in docking.

Acknowledgement

The authors would like to thank Professor John Barron at the University of Western Ontario, for providing the implementation of Simoncelli's, matched-pair 5-tap filters.

References

1. N. Ancona and T. Poggio, "Optical flow from 1D Correlation: Application to a simple time-to-crash detector," *International Journal of Computer Vision*, 14(2):131–146, 1995.
2. J L. Barron, D J. Fleet and S S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, 12(1):43–77, 1994.
3. M. Bober and J. Kittler, "Robust motion Analysis," *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 947–952, 1994.
4. D. Coombs, M. Herman, T. Hong and M. Nashman, "Real-time obstacle avoidance using central flow divergence and peripheral flow," *IEEE Transactions on Robotics and Automation*, 14(1):49–59, 1998.
5. D. Coombs and K. Roberts, "Centering behavior using peripheral vision," *1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 440–445, New York, NY, USA, 1993.
6. M A. Fischler and R C. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography," *Communications of the ACM*, 24(6):381–395, 1981.
7. D J. Fleet and K. Langley, "Recursive filters for optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):61–67,1995.
8. H. Liu, T H. Hong, M. Herman and R. Chellappa, "Accuracy vs. efficiency trade-offs in optical flow algorithms," *Computer Vision and Image Understanding*, pp 271–286, 1998.
9. B. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision," *Proceedings of DARPA Image Understanding Workshop*, pp 121–130, 1984.
10. B. McCane, K. Novins, D Crannitch and B. Galvin, "On benchmarking optical flow," *Computer Vision and Image Understanding*, 84(1):126–143, 2001.
11. C. McCarthy and N. Barnes, "Performance of optical flow techniques for indoor navigation with a mobile robot," *Proceedings of IEEE International Conference on Robotics and Automation*, pp 5093–5098, 2004.
12. R. Nelson and Y. Aloimonos, "Obstacle avoidance using flow field divergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1102-1106, 1989.
13. M. Otte and H. Nagel, "Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-Interlaced image sequences," *Artificial Intelligence*, 78(1):5–43, 1995.
14. J. Santos-Victor and G. Sandini, "Visual Behaviours for Docking," *Computer Vision and Image Understanding*, 67(3):223–238, 1997.
15. E P. Simoncelli, "Design of multi-dimensional derivative filters," *Proceedings of 1st International Conference on Image Processing*, pp 790–794, Austin TX USA, 1994.
16. M V. Srinivasan and S. Zhang, "Visual navigation in flying insects," *International Review of Neurobiology*, pp 44:67–92, 2000.
17. K. Weber, S. Venkatesh and M V. Srinivasan, "Insect inspired behaviours for the autonomous control of mobile robots," *Proceedings of the 13th International Conference on Pattern Recognition*, pp 1:156–160, Vienna, Austria, 1996.

Experiments in Robot Control from Uncalibrated Overhead Imagery

Rahul Rao[1], Camillo Taylor[1], and Vijay Kumar[1]

GRASP Laboratory, University of Pennsylvania
(rahulrao, cjtaylor, kumar)@grasp.cis.upenn.edu
http://www.grasp.upenn.edu

Abstract. In this work we address the problem of controlling a ground robot based on aerial image feedback in real time. We present an analysis of the relationship between a robot's ground velocity and the velocity of it's projection in the image. We also show how a subset of the parameters of the homography relating the ground plane and the image plane can be extracted from motion correspondences and subsequently used in a sample control task. Further, we extend our analysis to the case of robot control from a moving overhead camera using image feedback. The experimental implementation and validation of the schemes we have presented is one of the central goals of this paper.

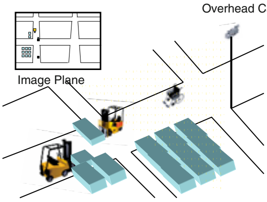


Fig. 1. Control of Vehicles in a Warehouse Using an Overhead Camera

1 Introduction

We are interested in applications in automation and robotics where overhead cameras can provide information that can be used to guide mobile robots. In a factory, a network of overhead cameras can be used to sense positions of robots, guide them around obstacles, and coordinate their motions toward their destinations. Outdoors, aerial vehicles with cameras can be used to guide ground vehicles toward targets. In the first example, it is possible to calibrate overhead cameras and instrument the robots and the environment. However, in general, we would like to be able to consider an *ad-hoc* network of uncalibrated cameras in an unstructured environment. This is particularly relevant for field robotics where the aerial vehicle's position and orientation will change over time, and it is difficult to instrument and structure the outdoor environment.

The goal of our research is to develop a theoretical formulation as well as a real time implementation scheme for the control of an unmanned ground vehicle (UGV) from an unmanned aerial vehicle (UAV) or an overhead camera. A simple

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 491–500, 2006.

situation where such an approach to automation will be useful is shown in Figure 1 which shows forklifts transporting objects in a warehouse while being monitored by an overhead camera. Such an approach can enable the use of personal robots in unstructured environments and AGVs and automated fork-lifts on factory floors without structuring or instrumenting the environment with landmarks, beacons or radio tags.

It is natural to ask what the motivation for such a *distributed sensing system* is. Such a configuration offers some advantages over onboard sensing. An onboard sensing system has *line of sight* sensing only and will thus rely on partial path planning strategies that will need to be updated on a regular basis in order to get the robot from its current position to a desired final position. Further, data fusion over space and time can be done effectively using distributed sensing. For example, if another camera were added to the existing system in Figure 1, the two cameras could be networked in order to provide a wider view of the environment as compared to a single overhead camera or a single onboard camera.

Our goal is the development of motion planning strategies and control of ground robots using feedback from a single, uncalibrated, overhead camera. The intention is to transfer the control problem from the real world to the image plane. Once the control problem is solved in the image plane there is a need to relate these quantities to control quantities (typically translational and angular velocities) used to guide the robot on the ground. In this paper we present a novel analysis of the relationship between the ground velocity of a vehicle and the velocity of its projection in the image. This is particularly useful in situations where robot velocity needs to be regulated, as mentioned above. We present in this work a metric relationship between image plane and ground plane velocities obtained by motion correspondences.

This paper is organized as follows- Section 2 presents an overview of related research in the use of visual feedback for control tasks. Section 3 presents an analysis and the metric relationship between the robot velocity on the ground plane and the velocity of it's projection in the image plane. Section 4 presents the use of the calibration procedure described in Section 3 to control the point to point motion of a robot from image feedback. Section 5 presents experimental validation of the ideas presented previously, followed by some concluding remarks in Section 6.

2 Previous Research

Robust feedback control schemes have been proposed ([1], [4], [5]) that are capable of accurately regulating a system to a desired configuration even when parameters relating the two frames are not exactly known [3]. Overhead imagery has been used in the past where the relationships between the two frames is *a priori* known. It has also been used in the situation where the image plane is parallel to the ground plane [2], thereby reducing the problem to finding a similarity transformation (4 parameters) relating the two planes rather than the more general projective transformation (8 parameters). Our work is novel in the sense that no knowledge is assumed of the intrinsic parameters of the camera or about the relative pose of the image plane with respect to the ground plane.

As far as robot manipulation tasks are concerned, research in visual servoing has been classified broadly into one of two approaches—*position based* and *image based* control systems [11]. In a *position based* (or *3D visual servoing*) control system features are extracted from the image and used in conjunction with a geometric model of the target and the known camera model to estimate the target's pose with respect to the camera. Control inputs are computed based on the errors in the estimated pose space thus making camera calibration necessary for reliable control. In the *image based* (or *2D visual servoing*) approach, controls are computed directly on the basis of image features, thus reducing errors due to sensor modeling and camera calibration. Taking advantage of *2D* and *3D* visual servoing techniques is the approach called *2-1/2 D* visual servoing. This approach is claimed to be more robust with respect to calibration errors, though it is also more sensitive to image noise if image features are used to compute control inputs [12].

In contrast to the papers mentioned above which mostly deal with kinematic models, visual feedback can also be combined with systems with second order dynamics. Ma, Košeçká and Sastry [9] used visual servoing techniques to control the motion of a car based on information obtained from a camera mounted on the vehicle. Zhang and Ostrowski [8] used visual servoing techniques for the control of an unmanned blimp. Cowan and Koditschek [13] used navigation functions on the image plane to design image-based servo algorithms to guide a planar convex rigid body to a static goal for all initial conditions within the camera's workspace. Current efforts [14] explore air-ground collaboration for feature localization with the air and ground vehicles complementing each others observations and sensor platform characteristics.

3 Relating the Robot's Ground and Image Plane Velocities

In this section we investigate the relationship between a vehicles velocity on the ground plane and the velocity of its projection in a fixed overhead camera. We will assume some means for measuring the velocity of the ground vehicle with respect to a fixed frame. For example, an unmanned ground vehicle equipped with a compass and an odometry system would be able to measure it's heading and speed with respect to a fixed magnetic reference frame. We expect, however, that these velocity measurements will contain errors which make it impractical to deduce the displacement of the robot by integrating velocity readings over time.

We perceive a situation where an overhead camera has sight of a ground robot and the desired trajectory. Let $\mathbf{w} \equiv (x, y, 1)^T$ denote the homogeneous coordinates of a point on the ground plane and $\mathbf{c} = (u, v, 1)^T$ denote the coordinates of the projection of $\mathbf{w}$ in the image. It is easy to show that $\mathbf{w}$ and $\mathbf{c}$ are related by a projective transformation $\mathbf{G}$. This can be expressed as

$$\mathbf{c} \propto \mathbf{G}\mathbf{w}, \ \mathbf{G} \in GL(3) \tag{1}$$

$$\Rightarrow \mathbf{w} \propto \mathbf{H}\mathbf{c} \tag{2}$$

where $\mathbf{H} = \mathbf{G^{-1}}$.

For clarity, the matrices $\mathbf{G}$ and $\mathbf{H}$ be represented in terms of their columns as $\mathbf{G} = \left(G^1 \; G^2 \; G^3\right)$ and $\mathbf{H} = \left(H^1 \; H^2 \; H^3\right)$ respectively. Similarly, they can be represented in terms of their rows as $\left(G_1 \; G_2 \; G_3\right)^T$ and $\left(H_1 \; H_2 \; H_3\right)^T$ respectively. Note that superscripts and subscripts are used to distinguish between matrix columns and rows. Replacing the proportionality sign in (2) by an equality we get

$$\mathbf{w} = \lambda \mathbf{H}\mathbf{c} \tag{3}$$

where $\lambda = \frac{1}{H_3.\mathbf{c}}$. Similarly, the image coordinates, $\left(u \; v\right)$, can be expressed in terms of the homogeneous ground plane coordinates, $\mathbf{w}$, as follows:

$$u = \frac{G_1.\mathbf{w}}{G_3.\mathbf{w}} \quad (4), \qquad v = \frac{G_2.\mathbf{w}}{G_3.\mathbf{w}} \quad (5)$$

Differentiating (4) with respect to time yields

$$\dot{u} = \frac{(G_3.\mathbf{w})(G_1.\dot{\mathbf{w}}) - (G_1.\mathbf{w})(G_3.\dot{\mathbf{w}})}{(G_3.\mathbf{w})^2}. \tag{6}$$

Using the expression for $\mathbf{w}$ from (3) we get

$$\dot{u} = \frac{(G_3.(\lambda\mathbf{H}\mathbf{c}))(G_1.\dot{\mathbf{w}}) - (G_1.(\lambda\mathbf{H}\mathbf{c}))(G_3.\dot{\mathbf{w}})}{(G_3.(\lambda\mathbf{H}\mathbf{c}))^2}. \tag{7}$$

Also, $G \cdot H = I$.
Using these facts, and the expression for λ, we can simplify (7) to yield

$$\dot{u} = (H_3 \cdot \mathbf{c})[G_1.\dot{\mathbf{w}} - u(G_3.\dot{\mathbf{w}})] \tag{8}$$

$$\dot{v} = (H_3 \cdot \mathbf{c})[G_2.\dot{\mathbf{w}} - v(G_3.\dot{\mathbf{w}})] \tag{9}$$

These expressions can be written compactly as

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = (H_3 \cdot \mathbf{c}) \begin{pmatrix} G_1.\dot{\mathbf{w}} - u(G_3.\dot{\mathbf{w}}) \\ G_2.\dot{\mathbf{w}} - v(G_3.\dot{\mathbf{w}}) \end{pmatrix} = (H_3 \cdot \mathbf{c}) \begin{pmatrix} 1 \; 0 \; -u \\ 0 \; 1 \; -v \end{pmatrix} \mathbf{G}\dot{\mathbf{w}} \tag{10}$$

where $\dot{\mathbf{w}} = (\dot{x}, \dot{y}, 0)^T$. This can be rewritten as

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = (H_3 \cdot \mathbf{c}) \begin{pmatrix} 1 \; 0 \; -u \\ 0 \; 1 \; -v \end{pmatrix} \left(G^1 \; G^2\right) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \tag{11}$$

Note that H_3, the third row of $\mathbf{H}$, can be expressed in terms of the columns of $\mathbf{G}$ as

$$H_3 = \frac{G^1 \times G^2}{(G^3) \cdot (G^1 \times G^2)} = \frac{G^1 \times G^2}{det(\mathbf{G})} \tag{12}$$

Since the matrix $\mathbf{G}$ represents a projective transformation, it's scale is immaterial which means that we can, without loss of generality, restrict $\mathbf{G}$ to be a matrix with unit determinant. Alternatively, one can note that scaling the matrix $\mathbf{G}$ by a constant does not affect equation (11). With this restriction, equation (11) becomes:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = ((G^1 \times G^2) \cdot \mathbf{c}) \begin{pmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{pmatrix} \begin{pmatrix} G^1 & G^2 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \tag{13}$$

This equation has a number of notable features. It provides a metric relationship between the velocity of the robot on the ground plane $(\dot{x}, \dot{y})$ and the velocity of its projection in the image $(\dot{u}, \dot{v})$. Strangely, it does this without requiring a normalizing division such as the ones implied by the projective relationships given in Equations (1) and (2). The expression reveals that the observed image velocity depends linearly on the ground plane velocity, $(\dot{x}, \dot{y})$, and quadratically on the vectors G^1, G^2 and $\mathbf{c}$.

Note that this expression only involves the first two columns of $\mathbf{G}$ which means that we cannot recover information about the third column of $\mathbf{G}$ solely from measurements of vehicle and image velocities. We can only hope to recover information about 6 of the 8 degrees of freedom that define the homography $\mathbf{G}$. The missing two degrees of freedom can be accounted for by noting that the origin of the ground planes reference frame can be chosen arbitrarily.

3.1 Camera Calibration from Motion Correspondences

We can recover the vectors G^1 and G^2 from image measurements by observing that the right hand side of equation (11) must be perpendicular to the vector $\left(-\dot{v}\ \dot{u}\right))$. This translates to:

$$\begin{pmatrix} -\dot{v} & \dot{u} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{pmatrix} \begin{pmatrix} G^1 & G^2 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = 0 \tag{14}$$

This homogeneous equation is linear in the unknown vectors G^1 and G^2. Given five separate measurements of a vehicles position in the image (u, v), it's instantaneous image velocity, $(\dot{u}, \dot{v})$, and its ground plane velocity, $(\dot{x}, \dot{y})$, we can construct a system of linear homogeneous equations which allows us to recover G^1 and G^2 up to an unknown scale factor, that is $\left(G^1\ G^2\right) = \alpha \left(\hat{G}^1\ \hat{G}^2\right)$. This scale parameter, α, can be resolved by substituting the scaled versions of G^1 and G^2 into equation (13) and enforcing equality as follows:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \alpha^3((\hat{G}^1 \times \hat{G}^2) \cdot \mathbf{c}) \begin{pmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{pmatrix} \begin{pmatrix} \hat{G}^1 & \hat{G}^2 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \tag{15}$$

4 An Image Based Feedback Linearizing Controller

We now describe a simple feedback linearizing controller that is built in the image plane and enables the robot to navigate from a certain starting state in the image to a desired destination, also in the image. Under the assumption that the motion of the robot is limited to a plane, this translates to motion from a starting point in the real to a desired destination in the real world. However, building the controller in the image plane makes it possible to do so without any real time knowledge or feedback of the position of the robot in the real world (i.e. (x, y)). All we need to know is the position of the robot in the image (i.e. (u, v)) and its orientation relative to a starting orientation, and that enables us to build such a controller. The controller has been experimentally tested in real time along with the calibration system that has been outlined in Section 3.1. Details of the experiments are presented in Section 5.2.

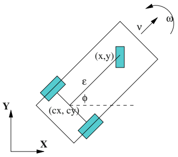


Fig. 2. A representation of the robot model used for image- based control tasks.

Let (u, v) be the image coordinates of a point (x, y) that lies a distance ϵ **off** the center of the rear axle, (c_x, c_y), of the robot, as shown in Figure 2. This means

$$\begin{aligned} x &= c_x + \epsilon \cos\phi \\ y &= c_y + \epsilon \sin\phi \end{aligned} \tag{16}$$

From equation (11), we obtain

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = (H_3 \cdot \mathbf{c}) \begin{pmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{pmatrix} \begin{pmatrix} G^1 & G^2 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \tag{17}$$

$$= \mathbf{J} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \tag{18}$$

Differentiating equation (16) once yields

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \cos\phi & -\epsilon\sin\phi \\ \sin\phi & \epsilon\cos\phi \end{pmatrix} \begin{pmatrix} \nu \\ \omega \end{pmatrix} = \mathbf{ZV} \tag{19}$$

where $\mathbf{V} = (\nu, \omega)$ represents the forward and angular speeds of the robot respectively. Thus

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \mathbf{JZV} \tag{20}$$

Now, if we desire to move the robot to the desired image coordinates given by (u^D, v^D), then we can choose, as a simple example, a proportional feedback controller of the type

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = -\mathbf{K} \begin{pmatrix} u - u^D \\ v - v^D \end{pmatrix} \tag{21}$$

$\mathbf{K}$ being a gain matrix with appropriately chosen gains. Thus the desired controls that need to be applied to the robot to get it to its desired position in the image can be obtained as

$$\mathbf{V}^D = -(\mathbf{JZ})^{-1}\mathbf{K}\begin{pmatrix} u - u^D \\ v - v^D \end{pmatrix} \tag{22}$$

We note here that since we have chosen a reference point that is not on the center of the axle (i.e. $\epsilon \neq 0$), the inverse of the matrix represented above always exists and there are no singularities.

5 Experimental Results

In order to investigate the efficacy as well as the feasibility of the proposed technique experiments were performed with an actual robotic platform. The experimental tasks involved extracting a subset of the parameters of the homography using the methods presented in the previous sections followed by the motion of the robot to a desired destination chosen in the camera image, in real time. Further, the overhead camera was subjected to random disturbances to observe the performance of the simple image stabilization technique using fixed fiducials on the ground plane that were tracked continuously during the robot's motion.

As has been outlined in Section 3.1 the vectors G^1 and G^2 can be recovered from at least five distinct measurements of $(u, v, \dot{u}, \dot{v}, \dot{x}, \dot{y})$ as the robot moves on the ground. Our tests gave us an idea of the accuracy of the results returned by our online estimation procedure.

5.1 Camera Calibration from Motion Correspondences

We conducted a series of experiments with the ER-1 mobile robot from Evolution Robotics and an overhead Dragonfly firewire camera with a Sony ICX204 sensor. The ER-1 is a three-wheeled, Hilare-like robot and is shown in Figure 3.

As mentioned in Section 3.1, the robot was moved in increments of 30° for 8 segments. The path of the robot along with a camera view of the robot as it moves during calibration are shown in Figure 3. It was continuously tracked, thus obtaining image information as well as real time ground velocity information. The parameters of the limited homography were then computed as outlined in Section 3.1. We conducted 20 experiments with the actual robot and recovered the values of the homography for each case.

We compared the results that were obtained from our online calibration procedure to obtain the first 2 columns of the scaled ground plane homography, $\hat{\mathbf{g}}$, with an estimate for the ground plane homography computed from point correspondences, $\tilde{\mathbf{g}}$- i.e. a 3×2 matrix representing the first two columns of the homography $\mathbf{G}$ obtained from point correspondences, appropriately scaled. We defined a benchmark to judge the accuracy of the recovered parameters with regard to their values in an ideal case. Specifically, we computed the quantity, $n_e = \frac{\|\tilde{\mathbf{g}} - \hat{\mathbf{g}}\|_F}{\|\tilde{\mathbf{g}}\|_F}$.

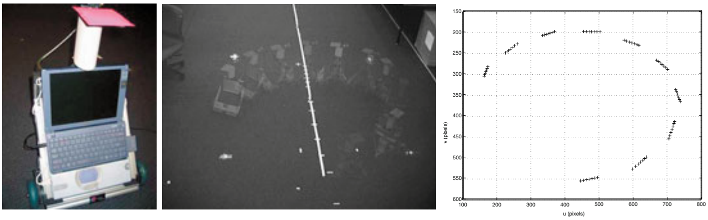


Fig. 3. The ER1 robot used for experimental runs (left) as seen by the overhead camera (center) and segments of the path traced by it (right) during a calibration run.

Over the 20 experiments tests that were conducted, the mean and median values of n_e were determined to be 0.0563 and 0.0540 respectively. A typical set of values for the recovered homography parameters as compared with the parameters recovered from point correspondences (referred to as *ideal*) are presented in Table 1.

Table 1. A Sample Comparison of Ideal and Experimental Values of the Limited Homography

$\hat{G}^1(ideal)$	$\hat{G}^1(exp)$	$\hat{G}^2(ideal)$	$\hat{G}^2(exp)$
-8.8347	-8.9873	-2.0486	-2.1911
-0.2190	-0.4488	3.3972	3.3299
-0.00064	-0.0008	-0.0048	-0.0049

5.2 Point to Point Motion of a Mobile Robot Using an Image Based Controller

During experiments that were performed in real time, the camera was first calibrated using motion correspondences and then the control scheme described in Section 4 was applied to get the robot to randomly chosen destinations in the image plane. A typical path followed by the robot is shown in Figure 4. The path taken by the robot during the calibration procedure is represented by dotted lines while the solid arcs are the path taken by the robot as it moved from one intermediate destination to another chosen in the overhead image.

5.3 Extension to a Moving Overhead Observer

So far it has been assumed that the overhead camera remained stationary during the motion of the ground vehicle. This restriction can be removed if we posit the existence of a set of 4 or more fixed points on the ground plane which can be identified and tracked over time. These point correspondences can be used to compute collineations that effectively fixate the ground plane in the image, a simple form of image stabilization. Note that we do not require any information about the inertial locations of the tracked points on the ground plane since they are only being used to compute collineations between images. This stabilization procedure returns us to the realm where the stationary camera analysis can be applied.

This procedure can be better understood by looking at the two images shown in Figure 5. These are views of the robot and the features on the ground from two

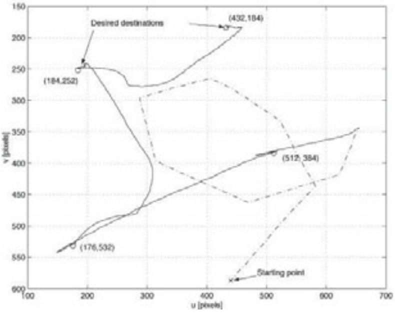


Fig. 4. The robot's path during calibration (dotted lines) and during point to point navigation (solid arcs). Intermediate destination points chosen randomly by the user are indicated by circles.

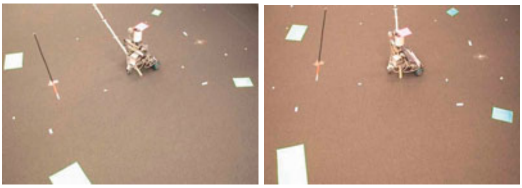


Fig. 5. The robot and fiducials as seen by the overhead camera from two viewpoints- a reference (left) and an intermediate (right) during an experiment.

different camera viewpoints, as might be seen during observations from a mobile overhead camera. Assuming that the frame on the left is the reference frame (F_{ref}) and the frame on the right is an intermediate frame, F_i, a collineation can be computed that would uniquely relate points in F_i to points in F_{ref}. Thus, at every frame, the *current* position of the robot as well as it's desired destination as seen in F_i can be mapped to F_{ref}, thereby transforming the entire problem of controlling the robot to frame F_{ref}. This simple form of image stabilization has also been tested on the experimental platform in real time.

6 Conclusions

This paper considers the problem of controlling the motion of a vehicle moving on a ground plane based on the imagery acquired with an overhead camera, an important special case of the visual servoing problem. One of the primary motivations for considering this situation is the fact that overhead imagery provides a convenient context for defining motion objectives for the robot. In order to relate objectives defined in the image plane to the velocity commands passed to the vehicle we must first recover the relationship between the vehicle's velocity and the motion of its projection in the image. To this end, we propose a novel analysis of this relationship which yields some interesting insights. The analysis shows that the image velocity depends linearly upon its ground plane velocity, $(\dot{x}, \dot{y})$, and quadratically upon the location of its projection in the image, (u, v), and a subset of the parameters of the homography relating the ground and image planes, (G^1, G^2). More importantly, the ideas basic presented here have been implemented on an robot in 2-D experiments and we encouraged by the results which show that the schemes perform well even in the presence of measurement errors.

Our future in this area will address the development of motion and trajectory planning techniques in the image plane, and the extension to multiple uncalibrated cameras. Our ultimate goal is to be able to plan and control multiple robots in the image space.

Acknowledgment: This work was supported by NSF grants IIS02-22927 and IIS00-83240.

References

1. Ronen Basri, E. Rivlin and I. Shimshoni, “Visual Homing: surfing on the epipoles", *International Conference on Computer Vision, 1998*.
2. W.E. Dixon, D.M. Dawson, E. Zergeroglu, and A. Behal, “Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system”, *SMC-B*, 31(3):341—352, June 2001.
3. Gregory D. Hager, “Calibration-free visual control using projective invariance”. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 1009—1015, 1995.
4. Nicholas Hollinghurst and Roberto Cipolla, “Uncalibrated stereo hand-eye coordination”. *Image and Vision Computing*, 12(3):187—192, April 1994.
5. S. Soatto and P. Perona, “Structure-independent visual motion control on the essential manifold”. In *Proc. of the IFAC Symposium on Robot Control (SYROCO)*, Capri, Italy, pages 869—876, Sept 1994.
6. G. S. Hornby and S. Takamura and J. Yokono and O. Hanagata and T. Yamamoto and M. Fujita, “Evolving Robust Gaits with AIBO", In *Proc. 2000 Int. Conf. Robotics and Automation*, San Francisco, CA, pages 3040- 3045.
7. C. J. Taylor, J. P. Ostrowski and Sang Hack Jung, “Robust Visual Servoing based on Relative Orientation", In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Fort Collins, CO, pages 574-580.
8. Hong Zhang and James P. Ostrowski, “Visual servoing with dynamics: Control of an unmanned blimp", In *Proc. IEEE Int. Conf. Robotics and Automation*, 1999, Detroit, MI, pages 618- 623.
9. Yi Ma, Jana Košecká and Shankar Sastry, “Vision guided navigation for a nonholonomic mobile robot", *IEEE* Trans. on Robotics and Automation, 1999, 15(3), pages 521-536.
10. Gideon P. Stein, Ofer Mano and Amnon Shashua, “A Robust Method for Computing Vehicle Ego-motion", In *IEEE Intelligent Vehicles Symposium* (IV2000), Dearborn, MI.
11. S. Hutchinson, G. Hager and P. Corke, “A Tutorial on Visual Servo Control", *IEEE* Trans. on Robotics and Automation, May 1996.
12. Ezio Malis, François Chaumette and Sylvia Boudet, “2 1/2 D Visual Servoing", *IEEE* Trans. on Robotics and Automation, 1999, 15(2), pages 238-250.
13. N. J. Cowan and D. E. Koditschek, “Planar image based visual servoing as a navigation problem", In *Proc. IEEE Int. Conf. Robotics and Automation*, 1999, pages 611-617.
14. Ben Grocholsky, Selcuk Bayraktar, Vijay Kumar, Camillo J. Taylor and George Pappas, “Synergies in Feature Localization by Air-Ground Robot Teams", International Symposium on Experimental Robotics, Singapore, June 2004.

XIII. Human-Robot Interactions, Haptics and Augmented Reality

Haptics and augmented reality provide a responsive media of interaction between human and robots. These interactions are intuitive to the way humans interact with the environment and are therefore user friendly as the user immediately feels the responses provided by the robot. Recent efforts have made headways in implementing haptics and augmented reality into many existing technologies, such as providing kinaesthetic feedback to minimally invasive surgery, teleoperation of robots in hazardous areas and interaction with computer models for various purposes.

In the first article, Iba, Paredis and Khosla present a new and friendly way that humans could interact with robots. The goal is a communication medium that does not require expert knowledge of robotic programming, but one that could easily come with a household product such as a robotic vacuum cleaner. The media chosen is multi-modal, utilizing verbal instructions and hand gestures. The system would recognize the verbal instruction and hand gestures, interpret the commands and plan and execute the list of primitives.

The next article by Butler and co-workers provides an example that humans may not be the only ones interacting with machines. In an interesting experiment, the authors attempted to control cow herds by creating a virtual fence drawn across the grazing land by unseen coordinates. The cows are fitted with collars that are tracked by the GPS and emit loud noise when the cows go across *the fence*. The loud noise was to herd the cow back into the designated area.

The third article by Ueberle and Buss provides the design and analysis of a kinematically redundant haptic device. Kinematic redundancy was introduced to obtain large workspace and singularity robustness. Another approach is taken by Barbagli and co-workers in the fourth paper. They study the possibility of creating a very-large-workspace haptic device by mounting the haptic feedback system on a mobile robot. The paper presents various methods of evaluating the mechanical impedance that can be produced by such systems.

A robust teleoperation of a mobile manipulator with haptic feedback is demonstrated by Park and Khatib in the last paper of the chapter. It is an experiment that takes into account all the necessary aspects of teleoperations to form a truly robust system. It deals with uncertainties and time varying parameters, time delay through the wireless LAN network, high degree of redundancy of the slave system through decoupling of task and posture space. A virtual spring system between slave and master with local force feedback compensates for the dynamics of the slave.

Interactive Multi-Modal Robot Programming

Soshi Iba[1], Christiaan J. J. Paredis[3], Pradeep K. Khosla[1,2]

[1] The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
[2] Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
[3] Systems Realization Laboratory, G. W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA

Abstract. The goal of the Interactive Multi-Modal Robot Programming system is a comprehensive human-machine interface that allows non-experts to compose robot programs conveniently. Two key characteristics of this novel programming approach are that the user can provide feedback interactively at any time through an intuitive interface and that the system infers the user's intent to support interaction. The framework takes a three-step approach to the problem: multi-modal recognition, intention interpretation, and prioritized task execution. The system is demonstrated by interactively controlling and programming a mobile vacuum cleaning robot. The demonstrations are used to exemplify the interactive programming and plan recognition aspects of the research.

1 Introduction

The number of vacuum cleaning robots being manufactured and sold to household consumers is rising as they become increasingly popular as technical gadgets [1]. Depending on the price range, these robots can navigate, avoid obstacles, localize themselves, and cover an area autonomously or through manual control by the user. However, none of them offers a novice-friendly interface to control and program a robot for particular tasks. As robots enter the human environment and come in contact with inexperienced users, they need to be able to interact with users in a multi-modal fashion - keyboard and mouse are no longer acceptable as the only input modalities.

This paper introduces a novel approach for programming a robot interactively through a multi-modal interface. The key elements behind this novice-friendly system are intuitive interfaces based on speech and hand gesture recognition, intention modeling and recognition, and interaction capabilities that allow the user to take over the control of the robot at any given time. Such interaction capabilities give a sense of assurance to users and help them in dealing with a robot by including a human in the control loop.

Designing and building such a system involves multiple problems. The system needs to infer underlying robot commands from a sequence of multi-modal user inputs to formulate a robot program. The system also needs to allow preemptive interaction between the user and the robot, which involves suspension, arbitration,

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 503–512, 2006.

and resumption of a robot task. Furthermore, it is desirable to reduce user interaction with a graphical user interface in order to allow direct interaction between the user and the robot.

There have been many approaches to improving task automation. The current state-of-the-art in user-friendly task automation is based on iconic programming [2] and/or programming by human demonstration [3]. The goal of these paradigms is to translate the burden of programming robotic systems from robot experts to task experts. The problem arises in these approaches when the programmed system is diverted from its intended task, and the user is required to re-execute, or even worse, to reprogram the task all over again. Our approach adds an intuitive multi-modal interface and interactive programming and execution capability so that the user can take over and correct the system with ease. The use of multi-modal interfaces in human-robot interaction is an active field with numerous applications such as teleoperation [4] and navigation [5]. We use voice and hand gestures to convey a combination of symbolic and parametric information to the robot.

2 System Design

The framework is composed of three functional modules, as illustrated in Figure 1. The first module (multi-modal recognition) translates hand gestures and spontaneous speech into a structured symbolic data stream. The second module

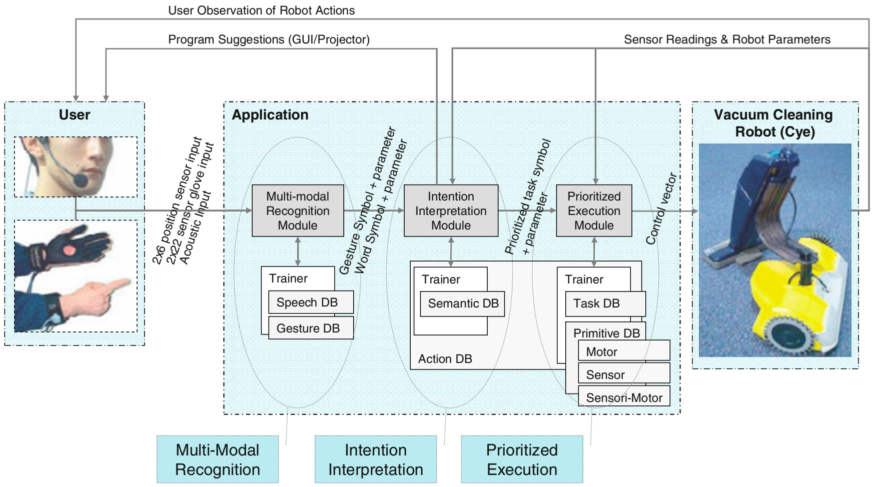

User Observation of Robot Actions
Program Suggestions (GUI/Projector)
Sensor Readings & Robot Parameters
User
Application
Vacuum Cleaning Robot (Cye)
2x6 position sensor input
2x22 sensor glove input
Acoustic Input
Multi-modal Recognition Module
Gesture Symbol + parameter
Word Symbol + parameter
Intention Interpretation Module
Prioritized task symbol + parameter
Prioritized Execution Module
Control vector
Trainer
Speech DB
Gesture DB
Trainer
Semantic DB
Action DB
Trainer
Task DB
Primitive DB
Motor
Sensor
Sensori-Motor
Multi-Modal Recognition
Intention Interpretation
Prioritized Execution

Figure 1. The framework is roughly divided into three modules: multi-modal recognition, intention interpretation, and prioritized execution, and each roughly corresponds to providing an intuitive interface, robot programming and suggestion, and an interactive control capability of the overall system. **514**

(intention interpretation) selects the appropriate set of primitives based on the user input, current state, and robot sensor data. Finally, the third module (prioritized execution) selects and executes primitives based on the current state, sensor inputs, and the task given by the previous step.

The vacuum cleaning robot used in our experiments is Cye, a 10” by 16” two-wheeled robot which carries a portable vacuum cleaner on its tail. It uses dead reckoning to localize itself with respect to the starting position, and is subject to cumulative error as it navigates. The robot comes with a graphical user interface that allows users to control Cye using a mouse, and to program it using an iconic programming framework. We added the three modules on top of the graphical user interface to provide features such as: (1) A hand gesture and spontaneous speech recognition interface; (2) A robot simulator for users to control and execute a program in the virtual environment; (3) Capability to preemptively interrupt a set of instructions; (4) Capability to create and adjust the program on-the-fly; (5) Capability to suggest a program that the user may want to execute based on the partial sequence of robot trajectory.

3 Multi-Modal Recognition

The system is capable of recognizing two different modalities: hand-gestures and spontaneous speech. The primary motivation for multi-modality is that no single mode provides a highly competent human-robot interface. Verbal cues are most appropriate when either party needs to convey symbolic information with an unambiguous context, such as “stop”, “move forward”, “turn right”, etc. Motion cues are an essential supplement when deictic elements are involved, as in the verbal commands “go there” and “move this way”. Such instructions are ambiguous without accompanying gestures, which are inherently more suitable to express position and geometry.

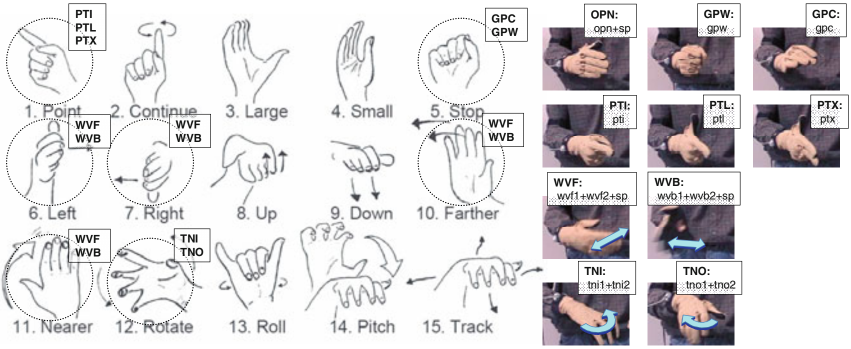


Figure 2. Left (drawings): The list of gesture vocabulary by Quek [6]. Gestures used for our system are circled with the corresponding three-letter gesture symbols (in upper-case). Right (pictures): Gesture symbols recognized by our HTK-based gesture recognition module, along with the gesture phonemes (in lower-case).

The list of hand gestures was selected based on work by Quek [6], who compiled the list of gestures necessary to express spatial information in 3D space (Figure 2). The gesture recognition module is implemented using the Hidden Markov Model Toolkit (HTK) [7] customized to recognize gestures at 60Hz from temporal data streams from two 22-sensor CyberGloves [8], each with a Polhemus 6DOF inductive position sensor. Using HTK, which was primarily developed for speech recognition research, we were able to treat hand gestures as words, and a sequence of hand gestures as a sentence. HTK offers versatile tools and the capability to build HMMs for recognition purposes. Using model adaptation techniques and triphone modeling (e.g.: *gpw-tno1+tno2*) strategy to capture inter-word transitions as well as intra-word transitions provided by the HTK, the gesture recognition module is able to achieve an average accuracy of 92% for previously unseen users.

The spontaneous speech recognition is implemented on a public domain large-volume speech recognition engine. In our implementation, spontaneous speech is translated into words using Microsoft Speech SDK [9], an off-the-shelf speech recognition package. It is responsible for recognition, adaptation, and grammatical parsing of the spoken words. The package's TTS (Text-To-Speech) capability is used for recognition acknowledgments and making user suggestions, as described in Section 5. List of a basic speech vocabulary consists of motion, deictic, name, attribute and programmable command terms [10]. The choice of words is task-dependent and experience-based. Readers interested in vocabulary selection for speech-based robot interactions should refer to [11], which deals with a vocabulary-based human-robot instruction system.

The temporal streams of results from hand gesture recognition and spontaneous speech recognition are combined to generate a semantically correct interpretation to control and program the robot. Results from both recognition processes are streamed into a buffer that gives a one-second window to decrease ambiguity in the speech recognition result (e.g. "this", "that") by grounding proper parameters from the gesture recognition result. Interpretation module uses the semantics database, which is described in [10] to interpret multi-modal recognition results to intended robot task symbols. It is implemented as a lookup table of candidate task symbols and their priorities from input symbols from the multi-modal recognition module.

4 Interactive Robot Control and Sequential Programming

Preemptive execution is crucial in providing the user real-time interaction to control and program a robot on-the-fly. Tasks are prioritized according to a pre-defined rule, and sequential robot actions in the tasks are executed and sometimes overridden based on the arbitration policy. This allows the user to handle situations such as making an emergency stop or avoiding an obstacle during the execution of other tasks. A robot program (task) is stored during interactive robot

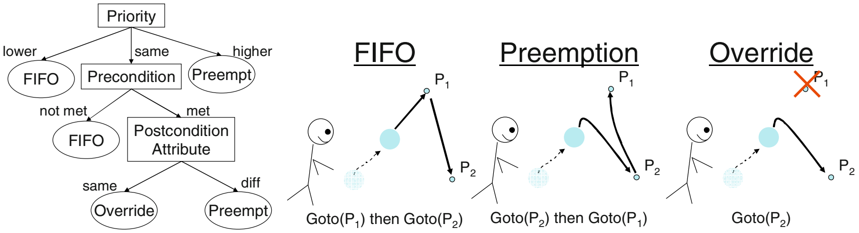


Figure 3. Arbitration Policy Tree and the possible actions for the sequence of instructions Goto(P_1) followed by Goto(P_2). Based on the policy tree, *Override* will be the correct action for the given example.

control. The user sets the module to a learning mode and executes primitives sequentially; the system remembers the sequence as a task.

An illustrative example of task arbitration is given in Figure 3. Imagine that the user first points to a particular position, P_1, and asks the robot to "go there", then points to a different position, P_2, and asks the robot to "go there" while the robot is moving toward P_1. There are three potential outcomes to the above sequential instructions: FIFO, Preemption, and Override. The outcome is decided by the arbitration policy tree described in Figure 3 by using features such as priority, precondition, and post-condition attribute. For the given example with two consecutive "go there" instructions, the outcome is an Override, since they have no preconditions and their priorities (given by the Semantic Database) and post-condition attributes (position) are the same. On the other hand, if the second instruction were "go this way" for a given direction, the outcome would be Preemption, since the post-condition attributes are different (position vs. direction).

To generate a sequential robot program, the user should execute the instructions in the intended order. When an action is overridden during the programming phase, the action up until the point of the override is programmed. The user can edit the program by interrupting its execution and showing an alternative action. In order to compose a non-sequential program which contains loops or if-then statements, the user may edit the program on the iconic programming interface, which is more suitable to display and edit the program flow.

5 Intention Awareness

The system's intention awareness is composed of two capabilities: intention recognition and adaptation. Instead of merely mapping the sequence of multi-modal recognition results to the set of actions using the semantics database, the intention-aware system should suggest which task the user may want to execute based on an incomplete sequence of instructions executed by the user. The recognition ability is similar to the auto-completion ability in a text-editing program. It is especially helpful when there is a large number of programs, and explicitly searching for any particular program may be time-consuming. In order to perform such recognition in the real world, we represent tasks in a probabilistic framework rather than as a discrete sequence of commands. A Hidden Markov Model (HMM) provides a way to model the task in a probabilistic framework, where both state transitions and observations can be expressed stochastically. Sets of tasks represented in HMMs are organized and compared to the current observation sequence to detect which task, if any, the user may want to execute.

For each program, a continuous-density HMM representation is generated from a discrete sequence of actions and observations collected during the programming phase. All program HMMs are integrated into a single HMM network, λ_{net}, which is then used to recognize the user's intended program based on the new observation sequence (Figure 4). When a robot is programmed interactively, the system collects an observation sequence $O_n = \{o_{0,n}\, o_{1,n} \ldots\, o_{t,n}\}$ for program action n, where $o_t = \{x_t,\ y_t,\ \theta_t\}$ corresponds to the robot position at time t. The sequence O is the collection of observations O_n resulting from program actions 1 to N. The robot program is then converted into an HMM by assigning one action per state, and its transition and observation probabilities are calculated from $O_{n=1..N}$. HMMs

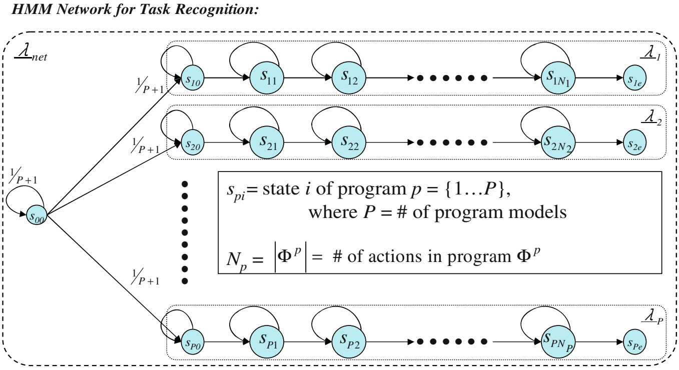


Figure 4. HMM network used for task recognition

```
Initialization:
  Assign a token with value of 1 to the initial shared state $s_{00}$.
  Assign a token with value of 0 to all other states.
  For all arcs not originating from state $s_{00}$, compute and store the value $\psi_{ij}$.
Algorithm:
  for each time $t$ do
    for each state $i \neq s_{00}$ do
      Compute and store the Mahalanobis distance between $o_t$ and $\mu_{ij}$;
      Pass a copy of the token in state $i$ to each connecting state $j$, multiplying its value by $a_{ij}b_{ij}(o_t)$.
      If the new token value underflows to 0, let the value be ε;
    end;
    Pass a copy of the token in state $s_{00}$ to each connecting state $j$, multiplying its value by $a_{ij}\bullet\psi_{mn}$. Choose
    the $\psi_{mn}$ for which the Mahalanobis distance between $\mu_{mn}$ and $o_t$ is the smallest;
    Discard the original tokens;
    for each state $i$ do
      Find the token in state $i$ with the largest value and discard the others;
    end;
    Normalize all tokens such that their sum equals 1;
    Find the state $q_t$ with the largest token value;
  end;
```

Figure 5. Viterbi Algorithm with Dynamic Garbage Collection

describing different programs are connected in a network forming a single HMM network, λ_{net}.

During recognition, the current observation sequence is evaluated and compared to the HMM network. To find the single most likely state out of all states in the shared HMM network for the current observation sequence, we use a modified Viterbi Algorithm described in Figure 5. The Viterbi algorithm, based on the Token Passing paradigm [12], has been modified by adding dynamic garbage collection, that is, recognizing the state s_{00} of the HMM network in which none of the programs is being executed. It is dynamic, in the sense that the state needs no prior training and the evaluation of whether or not the observation sequence is garbage depends entirely on the rest of the HMMs. A model update of each HMM provides online seamless adjustments of the statistics that describe the robot program. The update is performed after execution of the corresponding action associated with the state. The update does not require the entire sequence of previous observations, but rather it is updated using the previous statistics and the most recent observation sequence.

6 Demonstrations

We have conducted two demonstrations to exemplify the interactive programming and plan recognition aspects of the research. The first demonstration is to verify the operation of the overall system through sequential programming and adjustment of a mobile vacuum cleaning robot. The second is to demonstrate intention-awareness by letting the system detect the most likely program the user wants to execute, and having the intention model adapt to the current observations.

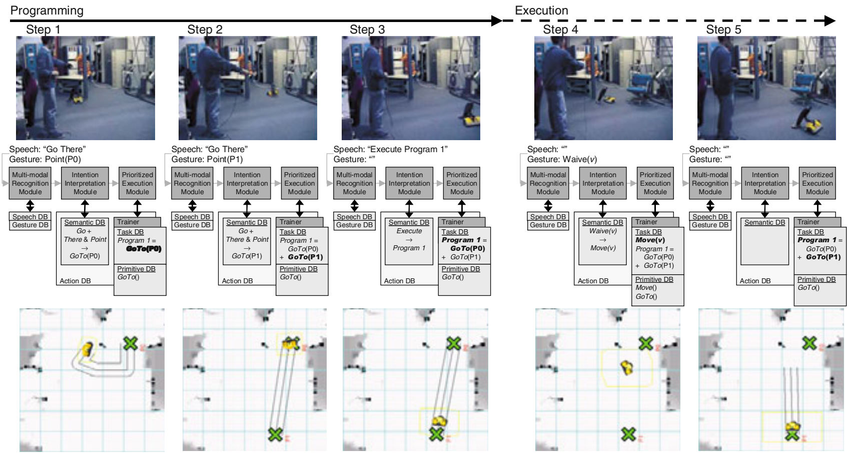


Figure 6: Task level sequential robot programming demonstration

6.1 Task Level Sequential Robot Programming

We have considered an interactive programming scenario, which has a user register numerous via-points to which the robot should navigate using its path planning capability. The robot can accept the user's preemptive speech and hand gesture commands to deal with unforeseen events. Figure 6 illustrates the sequences of the scenario including a sequence of camera snapshots with the corresponding conceptual illustrations of the framework, and the cropped images of the GUI.

In this scenario, the user first verbally commands that the subsequent actions be stored as "Program One". The user then executes the Goto primitive by combining the voice command "Go There" with the gestural command "Point" to indicate the destination (step 1). In general, deictic terms such as "This", "That", and "There" must be accompanied by a referential gesture to specify the corresponding task parameters. For the Goto primitive, the Cartesian coordinates are extracted from the intersection between the extension of the index finger and the ground. In step 2, the user enters another Goto primitive, but with a different end-position. After having saved these two primitives in "Program One" with the "Complete" command, the user can re-execute the program with the voice command "Execute Program One". However, in step 4, when the robot navigates to the second position from the first, it encounters an unknown obstacle. At this point, the user gestures the "Wave" command, which has a higher task priority and can be used to control the robot around the obstacle. When the obstacle has been cleared and the user stops waving, the robot returns to the execution of "Program One" (step 5).

6.2 Making Suggestions based on the Intention Awareness

This demonstration was conducted to verify the system's intention awareness. Assume that the database of robot programs contains three test programs:

Φ^1 = {Goto(P_1), Vacuum(On), AreaCoverage(P_2, P_3), Vacuum(Off), GoHome()}

Φ^2 = {Vacuum(On), Goto(S_1), Goto(S_2), AreaCoverage(S_3, S_4), GoHome()}

Φ^3 = {Goto(T_1), Goto(T_2), Goto(T_3)}

where P_i, S_i, T_i all represent positions on the map in (x, y). Their trajectories and the combined probability distributions of the converted HMMs are shown in the first and second columns in Figure 7. The last three columns in Figure 7 describe the output of the task recognition results from three test observation sequences τ^1, τ^2, and τ^3. Scores in the figure show that for the first test observation sequence, τ^1, the most probable state sequence follows that of the first program, Φ^1. This makes sense, because the trajectory of τ^1 was generated from Φ^1. The second observation sequence, τ^2, was generated by traversing regions covered by both the second and third programs, Φ^2 and Φ^3. The score in the figure also shows that is indeed the case. The third test observation sequence, τ^3, is a random traversal, which is captured by the dynamic garbage collection state, s_{00}, as "non-program".

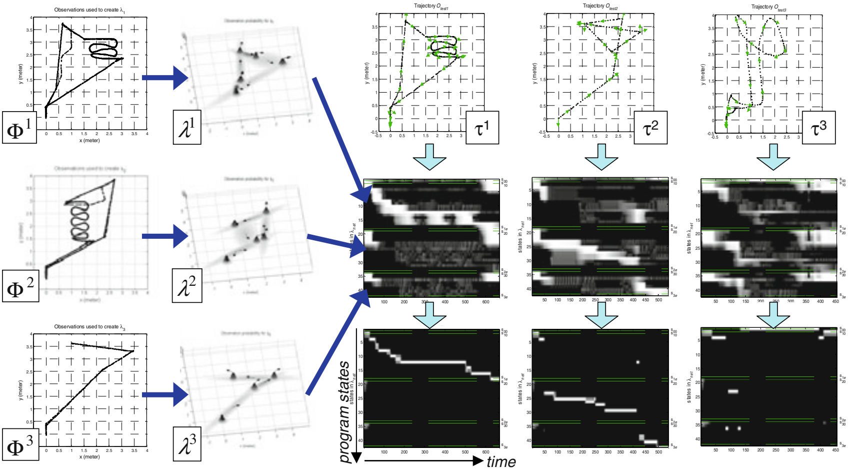


Figure 7. Task recognition results on three test observation sequences τ^1, τ^2, and τ^3. The images describe the state likelihood at any given time.

In each case, suggestions to the user are made through the text-to-speech system and the GUI projected on the wall. When the algorithm determines that there is a most probable state other than the dynamic garbage collection state, the application announces to the user through the text-to-speech system that there is a suggestion to be made. The suggested program is then executed on the simulated robot starting from the instruction associated with the most probable state. Based on the suggestion displayed by the simulated program, the user may or may not choose to execute the suggested program. It is important to note that the user is free to stop or alter the suggested program at any time, so the suggestion does not need to be a perfect match.

7 Conclusion and Future Work

In this paper, we have described an overall framework for interactive multi-modal robot programming and have illustrated the framework using two demonstrations. The programming approach offers, through an intuitive interface using hand gestures and speech recognition, the ability to provide interactive feedback to the robot to coach it throughout the programming and execution phases. The user's intent is captured in the form of a sequential robot program, and the flexibility given to the user by the framework through real-time interaction and an intuitive interface allows the captured intent to be closer to the user's true intent.

The first demonstration verified interactive multi-modal programming and execution, including the capability to interrupt commands preemptively. The second demonstrated that the system can determine the most likely high-level goal the user is trying to achieve, given a limited, initial sequence of task primitives.

To attain a comprehensive multi-modal interactive robot programming system, several elements still need to be added in the future. Although the programs generated by the current system can be re-executed, they are limited to fixed task sequences. To expand the generality of the paradigm, we need to add the ability to define non-sequential flow structures such as conditional branching and looping. Also needed is the ability to learn new primitives from demonstrations. The current implementation assumes that the given primitives cover the entire task space, and it would be convenient to be able to use the current programming paradigm to create new primitives. Discerning user preferences is also an important issue. Currently all programs in the HMM network are equally preferred prior to incoming observations.

Acknowledgements

This research was funded in part by DARPA under contract DAAD19-02-1-0389 and ABB under contract 1010068. Additional support was provided by the Robotics Institute at Carnegie Mellon University.

References

[1] Musser, G. (2003). "Robots That Suck." Scientific American, 288(2), 84-6.

[2] Gertz, M. W., Stewart, D. B., and Khosla, P. K. (1994). "A human machine interface for distributed virtual laboratories." IEEE Robotics & Automation Magazine, 1(4), 5-13.

[3] Ikeuchi, K., and Suehiro, T. (1994). "Toward an Assembly Plan from Observation, Part I: Task Recognition with Polyhedral Objects." IEEE Transactions Robotics and Automation, 10(3), 368-385.

[4] Fong, T., Conti, F., Grange, S., and Baur, C. (2000). "Novel Interfaces for Remote Driving: Gesture, Haptic and PDA." SPIE Telemanipulator and Telepresence Technologies VII, Boston, MA.

[5] Perzanowski, D., Schultz, A. C., Adams, W., Marsh, E., and Bugajska, M. (2001). "Building a multimodal human-robot interface." IEEE Intelligent Systems, 16(1), 16-21.

[6] Quek, F. (1994). "Toward a Vision-Based Hand Gesture Interface." Virtual Reality System Technology Conference, Singapore, 17-29.

[7] Young, S. J., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (2000). HTK: Hidden Markov Model Toolkit V3.0, Microsoft Corporation, Redmond, Washington, USA.

[8] CyberGlove Reference Manual (1998). Virtual Technologies Inc., Palo Alto, CA.

[9] Microsoft Speech SDK ver. 5.1. (http://www.microsoft.com/speech/dev/)

[10] Iba, S., Paredis, C. J. J., and Khosla, P. K. (2002). "Interactive Multi-Modal Robot Programming." International Conf. on Robotics and Automations, Washington, D.C., 161-168.

[11] Lauria, S., Bugmann, G., Kyriacou, T., Bos, J., and Klein, A. (2001). "Training personal robots using natural language instruction." IEEE Intelligent Systems, 16(5), 38-45.

[12] Young, S. J., Russell, N. H., and Thornton, J. H. S. (1989). "Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems." Cambridge University Engineering Dept.

Dynamic Virtual Fences for Controlling Cows

Z. Butler[1], P. Corke[2], R. Peterson[1], and D. Rus[1,3]

[1] Dartmouth Computer Science Department
Hanover, NH 03755 USA
{zackb,rapjr,rus}@cs.dartmouth.edu

[2] CSIRO ICT Centre
Brisbane, Australia
peter.corke@csiro.au

[3] Computer Science and Artificial Intelligence Laboratory, MIT
Cambridge MA 02139, USA
rus@csail.mit.edu

Abstract. A virtual fence is created by applying an aversive stimulus to an animal when it approaches a predefined boundary. It is implemented by a small animal-borne computer system with a GPS receiver. This approach allows the implementation of virtual paddocks inside a normal physically-fenced paddock. Since the fence lines are virtual they can be moved by programming to meet the needs of animal or land management. This approach enables us to consider animals as agents with natural mobility that are controllable and to apply a vast body of theory in motion planning. In this paper we describe a herd-animal simulator and physical experiments conducted on a small herd of 10 animals using a Smart Collar. The Smart Collar consists of a GPS, PDA, wireless networking and a sound amplifier. We describe a motion planning algorithm that can move a virtual paddock subject to landscape constraints which is suitable for mustering cows. We present simulation results and data from experiments with 8 cows equipped with Smart Collars.

1 Introduction

Our goal is to develop computational approaches for studying groups of agents with natural mobility and social interactions. Such systems differ in many ways from engineered mobile systems because their agents' motion is due to complex natural behaviors influenced by the environment (for example moving toward a food or water source). We wish to generate models of such systems using observed physical data and to use these models to synthesize controllers for the movement of the mobile agents. Our main motivation and application is in the agricultural domain. Herds of animals such as cattle are complex systems with interesting interactions between individuals, such as friendship, kinship, group formation, leading and following. There are complex interactions with the environment, such as looking for a water source in a new paddock by perimeter tracing along the fence and random walking within the perimeter. Such behaviors are well known to farmers but not so well documented. Unlike more familiar robot control problems, the animal state (stress, hunger, desire) is only partially observable and only limited control over motion can be exerted.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 513–522, 2006.

In this work we combine robotics, networking and animal behavior to create a fence-less approach to herding cows called *control by virtual fences*.

There are two fundamentally different approaches to controlling animal position: a physical agent such as a sheepdog or robot, and a stimulation device worn by the animal. In the first category there is the pioneering work of Vaughan [8] who demonstrated a mobile robot that was able to herd a flock of ducks to a desired location within a circular pen. In the second category there are a number of commercial products used to control domestic pets such as dogs. These typically employ a simple collar which provides an electric shock when it is in close proximity to a buried perimeter wire. The application of smart collars to manually control cattle is discussed in detail by Tiedemann and Quigley [5, 7]. The idea of using GPS to automate the generation of stimuli is discussed in [2, 4].

In [3] we describe our first experiments in controlling a herd of cows with a single static virtual fence using an approach that relies on ad-hoc networking. This paper extends [3] on the algorithmic side, by introducing motion planning for computing dynamic virtual fences whose goal is to muster the herd to a new location.

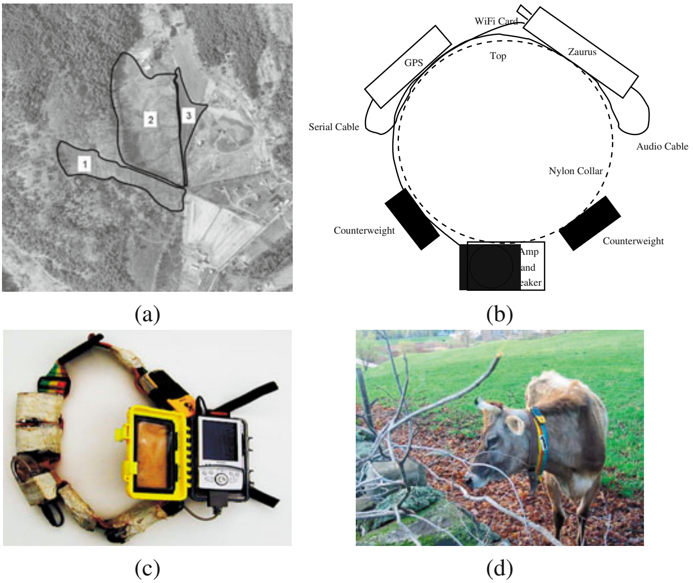
2 Approach

Our virtual fences combine GPS localization, wireless networking, and motion planning to create a fence-less approach to herding animals (see Figure 1). Each animal is fitted with a smart collar consisting of a GPS unit, a Zaurus PDA, wireless networking, and a sound amplifier. The collar is given the boundary of a polygonal virtual paddock in the form of a set of linear fences specified by their coordinates. The location of the animal is periodically checked against this polygon using the collar GPS. When in the neighborhood of a fence, the animal is given a sound stimulus whose volume is proportional to the distance from the boundary, designed to keep the animal within boundaries.

Each virtual fence is defined by a point F_p and a normal vector F_n. This representation allows for an easy computation of the distance d that the cow is behind (or in front of) the fence. If d is positive, the cow is in the desired region. Several fences can be combined to represent an enclosed boundary. We consider the virtual fence to be a repulsive potential field whose magnitude increases with distance beyond the fence line. Such a graduated stimulus will help the animal better understand the location of the fence [2, 3]. The magnitude of the field is rendered in terms of sound stimulus volume or stimulus rate. A more sophisticated approach is to monitor d over time, and stop the stimulus as soon as the cow begins to move toward the desired region.

Cattle domain experts have suggested using a library of naturally occurring sounds that are scary to the animals (a roaring tiger, a barking dog, a hissing snake) and randomly rotating between the sounds.

A virtual fence can be made dynamic by automatically and gradually shifting its location. A moving fence can be instantiated with a non-zero velocity v_F, in m/s. The point F_p is then moved as a function of time along the normal, $F_p(t) =$



(a) (b)

(c) (d)

Fig. 1. (a) Aerial view of Cobb Hill farm. The fields where experiments were conducted are outlined in black. North is up. The photo displays an area approximately 1 km on a side. (b) The components of the Smart Collar include a Zaurus PDA, WiFi compact flash card, eTrex GPS, protective case for the Zaurus, an audio amplifier with speaker, and various connecting cables. (c) A fully assembled Smart Collar, with PDA case open. (d) A cow with a collar.

$F_p(0) + v_F F_n t$. Several moving fences can be used to muster the herd according to the plans developed by our motion planning algorithm, described in Section 3.

2.1 System hardware

Prototypes of a Smart Collar were constructed using commercial off-the-shelf components that are readily available. Figure 1(b) shows the components of a collar. The computer is a Zaurus PDA with a 206MHz Intel StrongArm processor, 64MB of RAM, with an additional 128MB SD memory card. It runs Embedix Linux with the Qtopia window manager. The Zaurus has a serial port and stereo sound port. A Socket brand 802.11 compact flash card provides a wireless network connection. An eTrex GPS unit is connected to the serial port of the Zaurus. A small Smokey brand guitar amplifier is used to reproduce sounds from the Zaurus audio port. A fully assembled collar is shown in Figure 1(c). Figure 1(d) shows a cow wearing an early version of the collar.

2.2 Software Infrastructure

The components of the software used in the experiments are as follows:

Fences and Sounds. Fences can be added or removed at any time and several of them can be created at once from definitions stored in a file. Several fences can be combined to create convex polygonal shapes. When the GPS readings indicate a cow has crossed a fence a sound is triggered. The sounds are stored in WAV format files and can be selected from a list to be played on the Zaurus audio device. The volume of sounds is controllable on a percentage scale from zero to 100 percent. All fences use the currently selected sound and volume, which can be changed without redefining the fences.

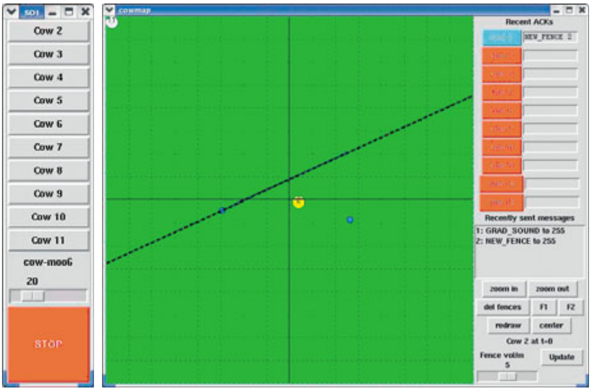
The fence module also reads and interprets the GPS data which arrives every two seconds when the GPS has a good lock on the satellites. It also sends a periodic `Alive` message indicating the collar is functional.

Message Handling. We developed our own adhoc messaging protocol layered over 802.11 sockets. All WiFi messages are multihop, being forwarded once by each collar, to improve range and connectivity within the herd. There are two message channels, one outgoing from a basestation and one incoming to the basestation. The outgoing channel is used for defining fences, manually triggering sounds, setting sound type and volume. The incoming channel carries `Alive` messages indicating a collar is active, and acknowledgment messages for receipt and proper interpretation of messages. Figure 4(a) shows a histogram of the number of hops required for an `Alive` message to reach the laptop basestation during an experiment. Most messages are relayed only once to reach their destination which indicates good connectivity between collars. Dynamic graphs of the message routing have shown us that connectivity among the herd is usually quite good since the cows tend to stay in proximity. Connectivity with the base station was problematic in that there is a tradeoff in staying far enough away to not influence the herd. WiFi networks are essentially line of sight and are blocked completely at times by the cows bodies.

Experiment Control. Both command line and GUI control programs are used to manage the collars in the field. The command line control program can be run on a Zaurus or Linux laptop and allows setting and deleting fences, setting type and volume of sound, and manually triggering a sound. The GUI control programs (see Figure 2) include the functionality of the command line program, and adds buttons for triggering sounds on specific cows, a map display showing current cow locations and status (i.e., relationship to fence boundary and whether a sound is playing), and a status display showing whether `Alive` messages have been received recently from each cow.

3 Planning for Dynamic Fences

An important goal of this work is to automatically move cows from one pasture to another (mustering). To make this possible, we have started to develop a path planning system for virtual fences. While this problem shares some basic features with traditional robot path planning, it has important differences as well.



1: GRAD_SOUND to 255
2: NEW_FENCE to 255

(a) (b)

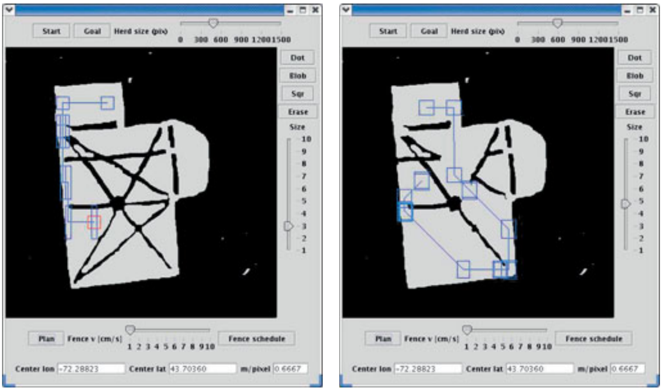
Fig. 2. GUIs used on laptops to monitor field experiments. (a) Sound control GUI. Pressing a button triggers the current sound on a specific cow. Current sound and volume can also be selected. (b) Map control GUI. Shows the last reported position of each cow, whether it is currently playing a sound, and whether an `Alive` message has been received recently. Buttons and text boxes in upper right show recent command acknowledgments from collars.

The planner creates a path from one point to another using a simple occupancy grid and A* search [6]. The object executing the plan is a virtual (polygonal) paddock, with significant extent that may change as it moves[1], as long as its area remains sufficiently large for the herd. Thus, it is easier to perform planning in the workspace than the configuration space. Obstacles can overlap somewhat with the virtual fence edges, changing the effective area of the virtual paddock but not altering the plan. We use planning operators that change the dimensions of the paddock while keeping the amount of free space within the paddock sufficient for the given number of animals. Finally, we would like the motion to consist of a small number of straight-line segments. This type of optimization is necessary because changing the animals' direction is more difficult and confusing than keeping them moving along their current vector, and to limit the number of fences that are downloaded to the collars. This optimization can be implemented by penalizing turns in the A* search.

To create a plan, instead of doing an expensive search in five dimensions (paddock location, width, height and motion direction) we first generate a pair of baseline plans. The first uses a point-sized virtual paddock and the second a constant-sized square paddock. These plans can be computed using the motion operators only. If the two plans are similar in length and complexity then the square paddock plan can be used as is. If not, we then gradually reduce the size of the square paddock until a plan is successful. The successful plan is then used as a base for a path with a variable-sized virtual paddock. The search for this final path is done

[1] Both extent and shape may change.

efficiently in three dimensions (width, height and distance along the base path.) The GUI developed for this planner along with examples of resulting plans is shown in Fig. 3.



Center lon -72.28823 Center lat 43.70360 m/pixel 0.6667

Center lon -72.28823 Center lat 43.70360 m/pixel 0.6667

(a) (b)

Fig. 3. GUI for dynamic fence planning. Environment is based on the Dartmouth Green, with some real-world paths defined to be obstacles. In (a), finding a path from the start (in top portion of free space) to the goal (at lower left) requires a significant change in the size and shape of the virtual paddock. (b) A more complex path — note that small overlap with obstacles is considered acceptable as long as sufficient free space remains in the virtual paddock.

Plans are turned into schedules of fences that are executable by the collar. For each segment of the path, four fences are required to define the paddock. A velocity is set for the fences, which in turn gives each segment a time interval over which its fences are active. This list of fences, with a point, normal vector, speed and relative time interval for each, is then given to the collars. The collars wait for an initialization message which tells them to make the time intervals absolute from that moment and continue to evaluate the fences and make appropriate stimuli for the course of the path.

4 Results

We have implemented both static and dynamic virtual fences in simulation and on 10 smart collars. Simulations were developed to investigate various algorithmic and stimulus methods. Early field experiments were performed on a small herd of cows at Cobb Hill Farm in Vermont. Later experiments were performed with people on the Dartmouth campus.

4.1 Simulation Experiments

To test the various virtual fence techniques, we developed a Matlab simulator that models the behavior of a herd of cows both with and without the virtual fence stimulus. We were inspired by Vaughan's duck simulator [8], but extended the animal model to account for the differences between the species as well as their environments. Most importantly, while we also use potential fields to model the effects of one animal's position on another's motion, we explicitly model the stress of each animal and use this to affect the animal's behavior. The animals have a two-state behavior model, walking and grazing, each with associated speeds and durations. In terms of motion, we use the potential force as a force on the cow, but model the cows as non-holonomic and give them a maximum angular velocity. If the virtual force given by the potential fields is not closely aligned with the cow's current direction, the cow will turn until the force causes it to walk in a reasonable direction.

In the simulation, stress is created by the fence stimulus as well as the nearby presence of other fast-moving animals or isolation from the herd. An animal in a low stress condition will alternate between grazing and walking, choosing a direction of walking randomly but biased toward the direction it is pointing. State duration, walking speed and direction are all stochastic. Unstressed animals exhibit very little herding instinct (as observed in the field) until they get very distant from each other. An animal that is experiencing high stress will move toward other animals, and will not resume grazing until its stress has gone down. The stress level of an animal decays over time.

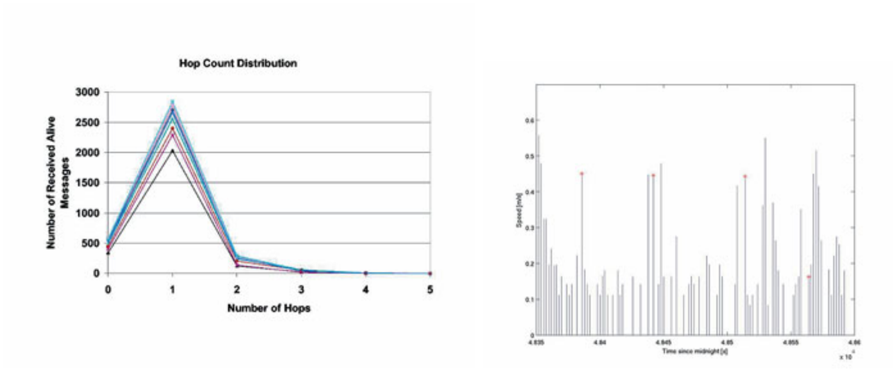
In addition to using stress, the stimulus has an immediate effect on the motion of the animal. We have used two different models, each of which take inspiration from field observations. In the first model, a stimulus causes the animal to quickly turn approximately 90°. This behavior was also observed in [5]. In the second model, the cow walks forward for a short time when stimulated.

To test the algorithms against these models, we ran virtual fences on a simulated herd with widely varying parameters. The overall goal was to move the virtual fence slowly into the herd and test how quickly the herd moved away from the encroaching fence. This was tested with different values for the grazing speed and walking speed of the cows, the level of herd-attraction and the probability that a stimulus would have the desired effect. We found that the parameters affected the overall speed of the herd in front of the fence and the number of stimuli that were applied, but in all cases the herd did move in the desired direction.

4.2 Cow experiments

We performed a series of field experiments at Cobb Hill, targeting four issues: (1) collecting data to create a grazing model for the cows, which is used in the fence control algorithm; (2) collecting connectivity data and information propagation data, which is used to determine the multi-hop routing method for networking the herd; (3) collecting stimulus-response data for individual animals; and (4) collecting response data for the virtual fence on a group of animals.

Our preliminary results are encouraging. Animals respond to artificial potentials of sounds generated by the virtual fence by moving forward if they are on their own, or toward the group if they are in close proximity to the group. Figure 4(b) shows a very strong correlation between when a sound is played, the asterisk, and a velocity spike. The animals responded to sounds (see Figure 4(b)) but habituation to stimuli was a problem. Others [2] have combined the sounds with shocks to avoid habituation.



(a) (b)

Fig. 4. Experimental results. (a) Adhoc hop count data (b) Speed when sounds are played.

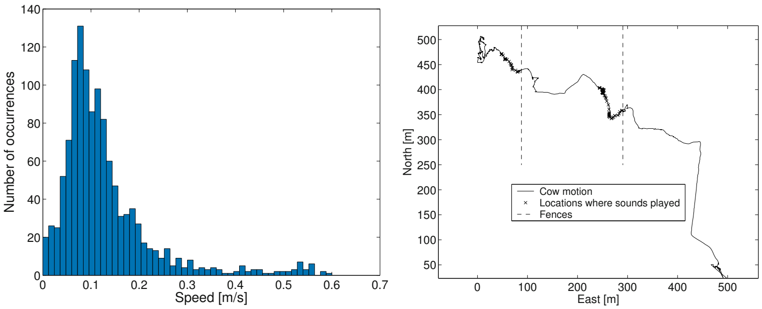
Our later experiments supported the two-state grazing/walking model. Fig. 5(a) is a speed histogram of one animal and clearly shows that it has a wide range of speeds throughout the day. They spend a large amount of their time moving quite slowly, and the rest of the time at higher, but differing, speeds.

We also successfully deployed a static virtual fence. As can be seen in Fig. 5(b), the cow was not deterred from passing through the fence, however it did slow down (to a statistically significant degree) while the sounds were playing.

4.3 Dynamic Fence Experiments

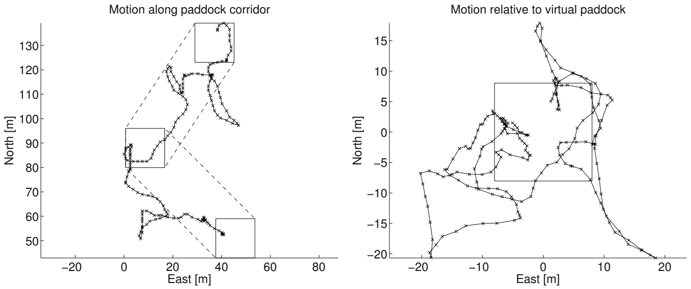
We have tested the dynamic fence algorithm using the smart collars on people[2]. Plans were created in the planning GUI and exported directly to the collars. The results were successful algorithmically, in that the fences appeared in the correct real-world locations at the desired times, but were less successful from a herding standpoint. Motion traces for one person, both absolute and relative to the moving paddock, are shown in Fig. 6. It is interesting to note that the general motion of the people was along the desired corridor, as in Fig. 6(a), despite spending very

[2] We will extend our experiments to animals as soon as the New England weather allows them to return to the pastures.



(a) (b)

Fig. 5. More experimental results. (a) Speed histogram (b) Trace past a fence (dashed lines) with sounds.



(a) (b)

Fig. 6. Dynamic fence experiment results. (a) Absolute and (b) relative motion of one person.

little time in the virtual paddock, as seen in Fig. 6(b). This may be biased by the experimenters walking in the general direction of the virtual paddock motion.

One benefit of this experiment was that the subjects were able to verbalize their feelings about the system. Primarily, they felt that the resolution of the stimulus gradient was insufficient to be of assistance. Also, they tended to be inquisitive, actively exploring the acceptable boundaries of their space. Together with relatively high walking speeds (averaging over 0.5 m/s) they were able to move a large distance between sounds and were unable to "find" their way back inside the virtual paddock. We believe that with a more appropriate stimulus (such as Anderson's stimulus for cows [2]) that this can be overcome.

5 Conclusion

Virtual fencing is a concept that could radically change the way that humans manage farm animals. We have developed a herd-animal simulator based on individual state machines and potential fields which exhibit many characteristics seen in real animal groups. Control techniques developed using the simulator have been tested on a small herd of 10 animals using using off-the-shelf hardware. Our physical experiments show promise for this application but much work remains to be done. The range of sound stimuli we tried were not very effective, although we did observe some reactions. Other researchers [1] have achieved consistent response to sound followed by electric shock stimuli, but this was not an option available to us. There are significant open questions related to stimulus habituation and the need for explicit training which will require close collaboration between roboticists and animal behaviorists.

Acknowledgments

We thank Simon Holmes a Court, Steve Petty, and the staff at Heytesbury Beef Australia for inspiring us to think about this problem and for great technical insights into cows and herding. We especially thank Steven, Kerry and Paul at Cobb Hill for their assistance and enthusiasm and the use of their cows for these experiments. We thank Lorie Loeb for facilitating our work. Tom Temple assembled the cow collars. Jenni Groh provided invaluable advice on the experimental procedure. The experimental part of this work has been done under protocol assurance A3259-01 given by the Institutional Animal Care and Use Committee (IACUC) of Dartmouth College. Zack Butler was supported under Award No. 2000-DC-CX-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security. Points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security.

References

1. D. Anderson. Personal communication on may 3, 2004, 2004.
2. D.M Anderson, C.S. Hale, R. Libeau, and B. Nolen. Managing stocking density in real-time. In N. Allsop, A.R. Palmer, S.J. Milton, K.P. Kirkman, G.LH. Kerley, C.R. Hurt, and C.J. Brown, editors, *Proc VII International Rangelands Conf.*, pages 840–843, Durham, South Africa, August 2003.
3. Z. Butler, P. Corke, R. Peterson, and D. Rus. Virtual fences for controlling cows. In *ICRA*, New Orleans, LA, 2004.
4. R.E. Marsh. Fenceless animal control system using gps location information. Technical Report US Patent 5,868,100, Agritech Electronics, February 1999.
5. T.M. Quigley, H.R. Sanderson, A.R. Tiedemann, and M.K. McInnis. Livestock control with electrical and audio stimulation. *Rangelands*, June 1990.
6. S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentics Hall series on Artificial Intelligene, 2002.
7. A.R. Tiedemann, T.M. Quigley, L.D. White, W.S. Lauritzen, J.W. Thomas, and M.K. McInnis. Electronic (fenceless) control of livestock. Technical Report PNW-RP-510, United States Department of Agriculture, Forest Service, January 1999.
8. R. Vaughan, N. Sumpter, A. Frost, and S. Cameron. Robot sheepdog project achieves automatic flock control. *Proc. Fifth International Conference on the Simulation of Adaptive Behaviour.*, 1998.

Design and Control of a Hyper-Redundant Haptic Interface

Marc Ueberle[1] and Martin Buss[2]

[1] Control Systems Group, Technische Universität Berlin
Einsteinufer 17/EN 11, D-10587 Berlin, Germany
ueberle@rs.tu-berlin.de http://www.rs.tu-berlin.de

[2] Institute of Automatic Control Engineering, Technische Universität München
D-80290 München, Germany
Martin.Buss@ei.tum.de http://www.lsr.ei.tum.de

Abstract. This paper presents the design concept of a hyper-redundant haptic interface with 10 actuated degrees-of-freedom (DOF). The kinematical redundancies allow a significantly larger workspace, while reducing the overall device size. Moreover, an increase in a variety of dexterity measures and a singularity robust redundancy resolution can be achieved. Numerical studies comparing the performance of local redundancy optimisation techniques are presented along with the results of hardware experiments for closed loop performance.

1 Introduction

Haptic interfaces are force feedback devices enabling bidirectional human system interactions via the sense of touch. Being able to exert and react to the operator's motions and interaction forces they are used to mimic the dynamic behaviour of virtual or remote environments in virtual reality and telepresence systems. Although in recent years haptic devices have been successfully implemented in various task domains including e. g. medical and surgical VR systems [1], rehabilitation, tele- and micromanipulation [2], telemaintenance, virtual prototyping, scientific visualisation [3], and education [4], their enormous application potential does not seem to be exhaustively investigated and exploited.

The exploration of novel applications is often impaired by the unavailability of qualified haptic hardware. Scenarios involving operations in large regions are e. g. little studied mostly due to the fact that commercially available haptic devices suffer from comparatively small workspaces. Among the reasons for this is the kinematical design which is almost exclusively non-redundant (equal number of actuated joints and DOF at the end-effector). A well known deficiency of all non-redundant robots are interior singularities in the workspace (positions where the robot looses a DOF). Whereas in common industrial applications it is frequently allowed to drive the robot through such singularities it is necessary to circumvent these locations in haptic systems because of degrading dynamic properties in certain directions. As the device can be moved by the human operator at will motions close to singularities cannot be avoided by trajectory planning methods. Consequently, large areas of the device workspace are not available for haptic interaction. Also, an angular workspace of 360° around each axis is in general foreclosed for non-redundant robots.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 523–532, 2006.

These fundamental limitations are the motivation for the design of the hyper-redundant haptic interface VISHARD10 (**V**irtual **S**cenario **Ha**ptic **R**endering **D**evice with **10** DOF). The redundancies allow for a change of the internal configuration without changing the end-effector position and orientation. This kind of motion is called *null-space movement* or *selfmotion*. A well directed control of the selfmotion may contribute to increase the overall system performance. Besides the increase of workspace the redundant kinematical DOF offer a potential for operator collision avoidance and improvement of the dynamic properties and output capability.

Despite of these attractive features kinematical redundancies are little used for haptic devices. A possible explanation is the increased complexity of the mechanical design. To control the redundant DOF a computational augmentation is unavoidable. Also, the introduction of additional joints seems to be contrary to the objective of low inertia and high stiffness. The potential reduction of the device size relaxes these disadvantages. Moreover, the removal of the interior singularities allows to operate in workspace regions with increased stiffness and output capability. The only kinematically redundant haptic devices known to the authors are exoskeleton constructions, off-the-shelf redundant industrial robots, and the 7 DOF DLR light-weight robots [5] which are, however, not specifically designed for haptic applications.

2 Design of VISHARD10

2.1 Design rationale

The main design objective for the new hyper-redundant haptic interface is to provide a versatile haptic display with distinct advantages compared to existing solutions with respect to applicability for a variety of applications; large workspace free of singularities; high payload to accomodate various application specific end-effectors as e.g. surgical tools like drills [1] or scissors, to mount tactile stimulation actuators for combined kinesthetic and tactile feedback; offer redundancy to avoid user interference; provide dual-arm haptic interaction with full 6 DOF capability (again redundancy facilitates collision avoidance between the two arms).

This versatility is advantageous as it provides a benchmarking testbed for the development and feasibility studies of novel haptic applications. When new applications are developed, which require a certain workspace or force capability, the proposed redundant device can be constrained to these specifications by appropriate controller design; this includes the development of dedicated inverse kinematics algorithms incorporating the specific needs of these applications. Once the new haptic application has been rudimentarily developed using VISHARD10 and the feasibility is verified, a tailored, highly specialised haptic display can be developed.

2.2 Haptic device description

The design is based on the experiences obtained from a non-redundant haptic device VISHARD6 previously developed [6]. Both devices employ a force-torque sensor for active force feedback control to shape the device dynamics. The kinematic structure of the hyper-redundant design with 10 DOF, ViSHaRD10, is depicted in Fig. 1, the link length design is summarised in Tab. 1. It shows the reference configuration with all joint angles θ_i defined to be zero. In Fig. 2 the prototype is shown in a typical operational configuration.

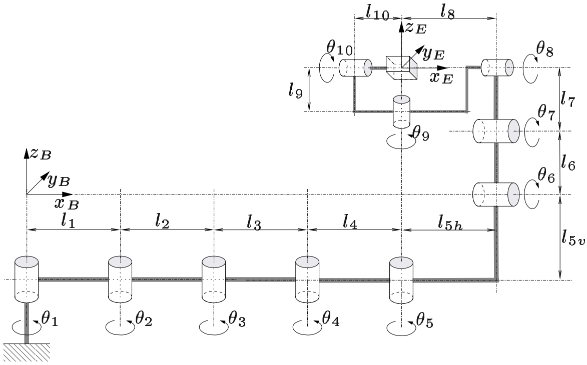


Fig. 1. Kinematical model of ViSHaRD10

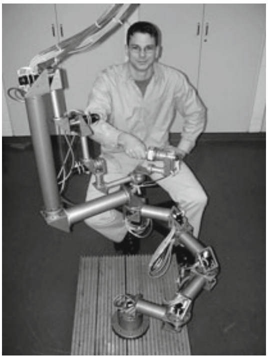


Fig. 2. ViSHaRD10

The first five joints are arranged in a SCARA configuration with vertical axes avoiding the need for an active compensation of gravity. This segment is assigned for the positioning of the end effector in the x-y-plane. Although a SCARA segment with three links is sufficient for the elimination of the singularity in the center of the planar workspace we decided for an arm with 4 revolute joints (4R) for two reasons: First, it is well known that the 4R arm provides improved dexterity compared to the 3R arm [7] and second, the avoidance of user interference is much simpler to achieve. Alternatively, we weighed up the use of a 2 DOF linear axes design which is, however, much larger in physical size.

Joints 6 and 7 are assigned to adjust the height of the end-effector. One simple inverse kinematics solution for them is to imitate a prismatic joint by means of not changing the end-effector position in the x-y-plane.

Joint 5 is used to prevent singular configurations in the wrist formed by joints 8, 9, 10. Despite tending to an increased wrist size we decided for a yaw instead of roll orientation for joint 8 to obtain decoupling of the wrist configuration from the end-effector height. This mechanism has a singularity when the axes of joint 8 and 10 have the same orientation which has to be avoided by a rotation of joint 5.

The axes of joint 5, 8, 9, 10 intersect at one point which is located 5 cm in front of the force-torque sensor (assuming that the motion of joint 6 and 7 is controlled accordingly). The benefit of such a mechanical decoupling of the angular from the translational DOF is twofold. First, it results in reduced natural dynamics of the orientational DOF and second, the torque capability of the actuators can be chosen to match the capability of a human wrist. Coupled designs usually tend to an unnecessary large torque capability requiring additional safety measures.

The mechanical realisation of this design is without joint angle limits and possibility for collision between parts of the structure. The actuation torque is provided by DC-motors coupled with harmonic drive gears offering zero backlash. The moment stiffness of the gears in the SCARA segment is increased by additional bearing support in order to avoid damage due to deflection in the harmonic drive component sets. The motors and the gears have been selected to meet the target specifications summarised in Tab. 2. In order to permit force feedback control the device is equipped with a six-axis JR3 force-torque sensor providing a bandwidth of 8 kHz at a comparatively low noise level. The joint angles are measured by digital MR-encoders with a resolution of 4 096 counts per revolution, resulting in a comparatively high position resolution when multiplied with the gear ratio varying from 50:1 to 160:1.

Table 1. Link length design of VISHARD10

Link i	Length
$l_1 = l_2 = l_3 = l_4$	0.25 m
$l_{5h} = l_8$	0.47 m
l_{5v}	0.71 m
$l_6 = l_7$	0.212 m
l_9	0.15 m
l_{10}	0.15 m

Table 2. Target specifications of VISHARD10

Property	Value
workspace	cylinder ∅1.7 m × 0.6 m 360° for each rotation
peak force	170 N
peak torque	pitch, yaw: 13 Nm roll: 4.8 Nm
translational velocity	> 1 m/s
maximum payload	7 kg
mass of moving parts	≈ 23 kg

3 Control of VISHARD10

3.1 General control scheme

The haptic simulation of a human's bilateral interaction with a virtual environment requires the control of the motion-force relation between operator and robot. This can be achieved by either controlling the interaction force of the device with the operator (impedance display mode) or the device motion (admittance display mode).

Admittance control is particularly well suited for robots with hard nonlinearities and large dynamic properties. In this display mode forces are measured and motion is commanded. The high gain inner control loop closed on motion allows for an effective elimination of nonlinear device dynamics. Contrary to impedance controlled haptic displays it is thus possible to render an isotropic closed-loop dynamic behaviour to provide the operator a "natural feeling". The drawback is the reduced capability for the display of low impedances.

Fig. 3 illustrates the admittance control scheme of VISHARD10. The interaction force $\boldsymbol{f}_{\text{ext}}$ of the operator is measured by a force-torque sensor. The virtual environment, modelled as admittance, relates the measured force to the desired end-effector velocity $\dot{\boldsymbol{x}}_d$. An algorithm for inverse kinematics resolution (see Sec. 3.2) calculates the desired joint velocities $\dot{\boldsymbol{q}}_d$. The desired joint angles $\dot{\boldsymbol{q}}_d$ are then the reference input to a conventional control law, e.g. a computed torque scheme [8].

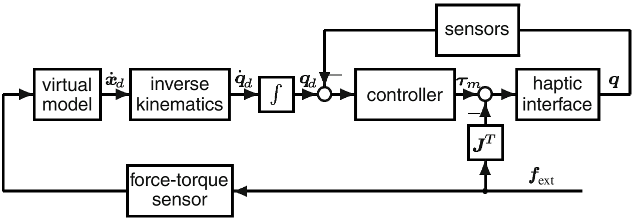


Fig. 3. Admittance control scheme

3.2 Inverse kinematics

The mapping

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{q}} \tag{1}$$

relates the joint velocity $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ to the end-effector velocity $\dot{\boldsymbol{x}} \in \mathbb{R}^m$, where $\boldsymbol{J}$ is the $m \times n$ Jacobian matrix of the manipulator. If $n > m$ the manipulator is said to be redundant with respect to the end-effector task. Then, a solution to the inverse kinematics problem, the calculation of $\dot{\boldsymbol{q}}$ from $\dot{\boldsymbol{x}}$, is not uniquely determinable as there are fewer equations than unknowns.

Pseudoinverse control

A solution to (1) proposed by Liegeois [9] that applies the Moore-Penrose generalised inverse $\boldsymbol{J}^{\#}$ of the Jacobian matrix is

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}\dot{\boldsymbol{x}} + [\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J}]\dot{\boldsymbol{q}}_0 \ , \tag{2}$$

where the first term is the minimum norm joint velocity solution and $[\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J}]\dot{\boldsymbol{q}}_0$ the homogeneous solution of (1) to project an arbitrary joint velocity vector $\dot{\boldsymbol{q}}_0$ onto the nullspace of $\boldsymbol{J}$. The homogeneous solution can be used to improve the device performance when chosing $\dot{\boldsymbol{q}}_0$ to optomise a performance criterion $H(\boldsymbol{q})$, a scalar function of the joint angles. Using gradient projection the redundancy can then be solved by substituting $\dot{\boldsymbol{q}}_0$ with $\alpha\nabla H(\boldsymbol{q})$ resulting in

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}\dot{\boldsymbol{x}} + [\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J}]\alpha\nabla H(\boldsymbol{q}). \tag{3}$$

In order to avoid singular configurations often a performance criterion evaluating the device manipulability is applied. The term manipulability can be understood as a definition of how easily and uniformly the device can move around the workspace. Yoshikawa [10] first introduced a quantitative measure of manipulability as

$$m(\boldsymbol{J}) = \sqrt{\det \boldsymbol{J}\boldsymbol{J}^T}. \tag{4}$$

which can be used as a side criterion $H\left(\boldsymbol{q}\right)$.

A drawback of the manipulability index is that translational and rotational manipulabilities are not distinguishable. Partitioning of the Jacobian $\boldsymbol{J}$ into orientational and translational parts

$$\begin{bmatrix} \boldsymbol{v}_{\mathrm{E}} \\ \boldsymbol{\omega}_{\mathrm{E}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{\mathrm{v}} \\ \boldsymbol{J}_{\omega} \end{bmatrix} \dot{\boldsymbol{q}}, \tag{5}$$

one can define the translational and rotational manipulability index to be

$$m_{\text{trans}} = m(\boldsymbol{J}_{\text{v}}) \quad \text{and} \quad m_{\text{rot}} = m(\boldsymbol{J}_{\omega}) \ . \tag{6}$$

The combination of multiple optimisation criteria is achieved by the formulation of one single composite performance index ψ as e. g. by using the weighted sum

$$\psi = w_1 C_1(\boldsymbol{q}) + w_2 C_2(\boldsymbol{q}) + \ldots + w_n C_n(\boldsymbol{q}), \tag{7}$$

where $w_i > 0$ are the weighting factors and C_i normalised criteria values.

Inverse function

Another approach to solve the redundancy is to define a single inverse function giving the joint coordinates for each point in a specified subset of the end-effector space. In contrast to pseudoinverse control these algorithms are cyclic (every closed path in the end-effector space is tracked only by closed paths in the joint space) avoiding unpredictable joint motions. Whereas it is in general not possible to define such a single inverse function for the whole six DOF end-effector space no matter how much redundant joints are used (see [11] for a proof) this approach may be rewarding in case the application does not require such a large workspace.

A simple inverse function can be defined when controlling the SCARA segment and joint 6 and 7 to mimic the operation of three prismatic joints:

$$\theta_1 = \arcsin\frac{x}{2l_1} \qquad \text{and} \qquad \theta_2 = -2\arcsin\frac{x}{2l_1} \tag{8}$$

$$\theta_3 = \arcsin\frac{y}{2l_3} + \arcsin\frac{x}{2l_1} \qquad \text{and} \qquad \theta_4 = -2\arcsin\frac{y}{2l_3} \tag{9}$$

$$\theta_6 = \arcsin\frac{z}{2l_6} \qquad \text{and} \qquad \theta_7 = -2\arcsin\frac{z}{2l_6}\ , \tag{10}$$

where x, y, z is the end-effector position respective the coordinate system $\{B\}$ defined in Fig 1. Setting joint angle 5 to $\theta_5 = \theta_{5,0} - \sum_{i=1}^{4}\theta_i$ a unique solution to the inverse kinematics problem can be determined. The choice for the constant $\theta_{5,0}$ affects the location of the interior singularity of the angular workspace.

Partitioned inverse kinematics solution

One important goal pursued with the kinematical design of ViSHaRD10 has been to provide the option to partition the inverse kinematics problem into two separate problems: the inverse kinematics for the positioning and the orientation stage. This can decrease the computational power required for the redundancy resolution significantly. The decoupling of the translational from the rotational movement is achieved when controlling joint 6 and 7 according to (10).

Defining $\boldsymbol{q}_{xy}$ to be the vector of the first four joint angles, $\boldsymbol{J}_{xy}$ the 2×4 submatrix of the manipulator Jacobian obtained by removing row 3 to 6 and column 5 to 10, and $\boldsymbol{v}_{xy}$ the vector containing the end-effector position in the x-y-plane, the solution for the joint rates of joint 1 to 4 is

$$\dot{\boldsymbol{q}}_{xy} = \boldsymbol{J}_{xy}^{\#}\boldsymbol{v}_{xy} + [\boldsymbol{I} - \boldsymbol{J}_{xy}^{\#}\boldsymbol{J}_{xy}]\alpha_{xy}\nabla H_{xy}\boldsymbol{q}_{xy}. \tag{11}$$

By setting $\boldsymbol{q}_{\text{rot}}^T = \left[\theta_5^* \; \theta_8^* \; \theta_9 \; \theta_{10}\right]$ with $\theta_5^* = \theta_5 + \sum_{i=1}^{4} \theta_i$ and $\theta_8^* = \theta_8 + \sum_{i=6}^{7} \theta_i$ the inverse kinematics for $\dot{\boldsymbol{q}}_{\text{rot}}$ is decoupled from the translational movement:

$$\dot{\boldsymbol{q}}_{\text{rot}} = \boldsymbol{J}_{\text{rot}}^{\#}\boldsymbol{\omega} + [\boldsymbol{I} - \boldsymbol{J}_{\text{rot}}^{\#}\boldsymbol{J}_{\text{rot}}]\alpha_{\text{rot}}\nabla H_{\text{rot}}\boldsymbol{q}_{\text{rot}}. \tag{12}$$

Here $\boldsymbol{\omega}$ is the rotational velocity command in operational space and $\boldsymbol{J}_{\text{rot}} \in \mathbb{R}^{3\times 4}$ the Jacobian relating $\boldsymbol{q}_{\text{rot}}$ to $\boldsymbol{\omega}$.

4 Experimental results

4.1 Numerical simulation experiments

The goal of the simulation experiments we carried out is the comparison of the full with the partitioned inverse kinematics solution approach. All experiments have been conducted applying Pseudoinverse control optimising the translational and rotational manipulability index (optimised solution), see (3), (4), (6), and (7), and Pseudoinverse control without optimisation (nonoptimised solution). A violation of joint velocity constraints has been avoided by a proper control of the step length α.

It has to be noted that a comparison of the results needs caution since they are highly dependent on the choice of the step length α and the weighting factors w_i. For all experiments we normalised the performance indices by the average of the maximum performance value among a set of end-effector positions and orientations. By experiment we determined suitable weighting factors for the translational and rotational performance with $w_{\text{trans}} = 1$ and $w_{\text{rot}} = 2$. The maximum step length for the full solution not producing selfmotion oscillations has been found by experiment with $\alpha = 2.1$; for the partitioned solution we used $\alpha_{\text{trans}} = \alpha_{\text{rot}} = 0.53$.

The end-effector has been commanded to follow at constant velocity v a circular trajectory with radius 0.4 m in the x-y-plane while keeping the height and orientation fixed. The initial end-effector position is given with $x = 0.8$ m, $y = 0$ m, $z = 0.3$ m such that the end-effector is at the center of the x-y workspace when half of a circle is traversed. The initial configuration of the joint angles is given with $\left[\theta_2 \; \theta_4 \; \theta_9 \; \theta_{10}\right] = \left[0 \; 0 \; -\frac{3\pi}{4} \; 0\right]$. Setting joint angles 6, 7 according to (10), $\theta_5 = \pi - \theta_1$, and $\theta_8 = \theta_6$ the angles of joint 1 and 3 are well defined.

Results

Fig. 4 shows for all solution approaches the progression of the manipulability indices when the end-effector follows the circular trajectory at velocity v =0.1 m/s and $\bar{v}$ =1 m/s (the results for the nonoptimised solutions are independent of the velocity v). The horizontal axis indicates the angular position γ on the trajectory. Please note, that the partitioned solution applies a manipulability index based on $\boldsymbol{J}_{xy}$ to optimise the SCARA selfmotion, see (11), whereas the index to evaluate the translational manipulability at the full solution applies to $\boldsymbol{J}_{xyz}$, see (6).

Looking at the results for *nonoptimised* Pseudoinverse control one can see that the full solution technique drives the system in a rotational singularity. Although the partitioned solution does not result in singular configurations this cannot be guaranteed for increased simulation time due to the noncyclic behaviour of Pseudoinverse control. The results for the nonoptimised partitioned solution reveal the decoupling of the rotational from the translational DOF; the rotational manipulability index is not changing as the commanded orientation of the end-effector is fixed; the rotational stage is decoupled from the selfmotion of the SCARA segment.

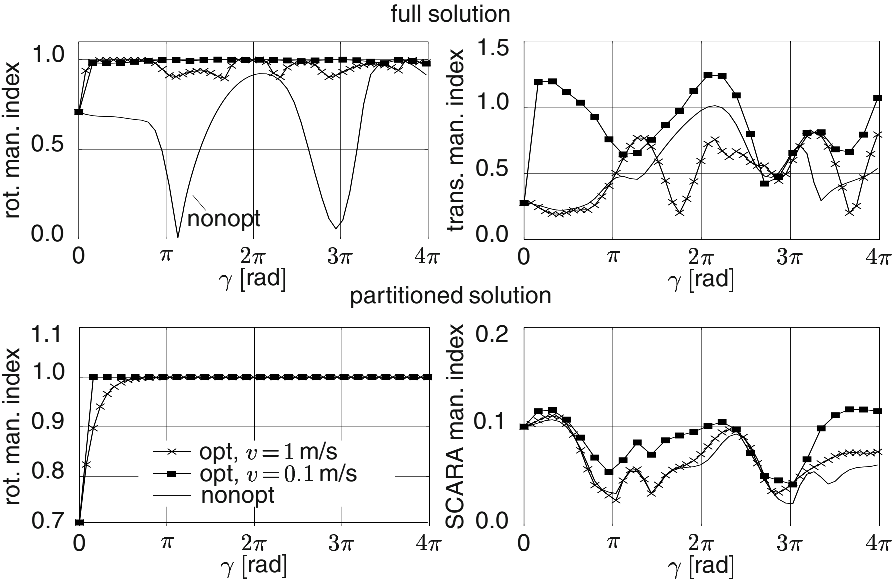


Fig. 4. Results of Pseudoinverse control techniques for different end-effector velocities

The results for the *optimised full solution* show that the optimised device selfmotion can effectively prevent singular configurations. The progression of the translational manipulability at $v = 1$m/s illustrates that optimised Pseudoinverse control is a local inversion technique not necessarily giving results superior to the nonoptimised solution. The benefit of the optimisation is, however, more significant when v is decreased. Both observations also hold for the optimised partitioned solution.

To allow a comparison of the full with the partitioned solution Fig. 5 presents the results of the optimised solutions at $v = 1$ m/s showing for both approaches the translational manipulability index based on $\boldsymbol{J}_{xyz}$. The results indicate a significant potential for performance improvement when solving for all 10 DOF simultaneously. The reason is that at the full solution the movement of joint 6 and 7 is not only used to adjust the end-effector height but also to increase the workspace in the x-y-plane. Moreover, by driving link 6 and 7 in an "outstretched" configuration (i. e. $|\theta_7|$ small) one can obviously increase the maximum velocity in height. For a fair comparison it has to be noted that the performance difference is dependent on the trajectory and the initial device configuration. We observed e. g. similar results for trajectories at an end-effector height close to zero. But it is also true that we never noticed superior performance for the partitioned solution with respect to translational manipulability. The main disadvantages of the full solution seem to be the increased computational effort and the coupling of the translational and rotational selfmotion. The coupled motion may be more difficult to understand and anticipate for the operator.

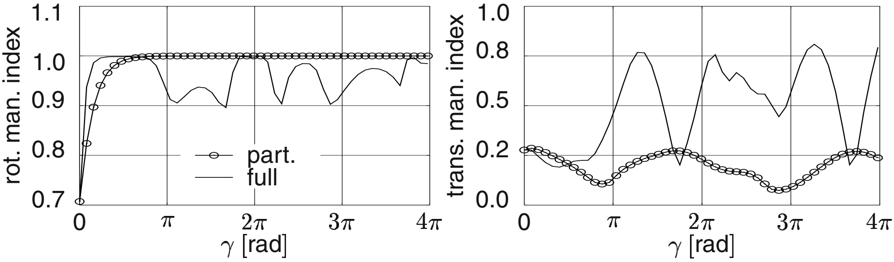


Fig. 5. Comparison of optimised full with partitioned solution at end-effector velocity of 1 m/s

4.2 Hardware experiments

In order to evaluate the device closed loop performance regarding the dynamic range of achievable admittances the maximum controllable stiffness and the minimum mass the device is able to render has been experimentally determined. For all experiments the inverse kinematics has been solved by using the inverse function described in Sec. 3.2. In addition we controlled the device to keep the end-effector orientation fixed constraining the haptic feedback to the three translational DOF.

In the experiments for the performance evaluation regarding the display of high admittances the dynamics of the virtual model in Fig. 3 has been set to

$$\dot{\boldsymbol{x}}_d(t) = \frac{1}{m} \int_0^t \boldsymbol{f} \, \mathrm{d}t \,, \tag{13}$$

rendering the behaviour of a point mass. The minimum mass producing no instabilities during vigorous end-effector motions induced by a human operator was about $m_{\text{min}} = 10$ kg. In order to evaluate the performance of rendering high stiffness a virtual wall has been implemented. The experiment proceeded by increasing the wall stiffness until instability was first detected. The results indicate a realisable cartesian stiffness of 80 000 N/m.

It has to be noted that the position control scheme used in the hardware experiments is a comparatively simple high gain PD control law. Based on our experiences obtained from the 3 DOF haptic interface ViSHaRD3 previously developed we expect a significant reduction of the minimum mass ViSHaRD10 is able to display when implementing advanced control laws based on the dynamic device model.

5 Summary

In this paper we proposed the introduction of actuated kinematical redundancies in kinesthetic (haptic) feedback device design to overcome performance limitations of non-redundant interfaces due to internal singularities. We presented the kinematical design of a highly versatile haptic interface ViSHaRD10 with 10 actuated DOF offering a large workspace free of singularities and high force capability. The goal of this prototype is to provide a benchmarking testbed for the development and feasibility studies of novel haptic applications. We described the basic control strategy of the prototype along with inverse kinematics solution techniques. The kinematical design allows for a decoupling of the translational from the rotational device movement enabling the partitioning of the inverse kinematics problem in two subproblems that can be solved separately to reduce computational effort. A comparison of this partitioned solution approach with the full solution reveals a significant potential for performance improvement when solving for all DOF simultaneously. The simulation results also show the benefit in terms of effective singularity avoidance when applying techniques to optimise the device selfmotion. Finally, experimental results regarding closed loop performance of the device have been shown. Future work will include further in-depth studies of redundancy exploitation techniques specific to the goal of haptic human-device interaction, the implementation of dynamic model control algorithms, the implementation of tactile actuators, and the development of novel haptic applications e. g. dual handed open surgery.

Acknowledgments — This work is part of the TOUCH-HapSys project financially supported by the 5th Framework IST Programme of the European Union, action line IST-2002-6.1.1, contract number IST-2001-38040. For the content of this paper the authors are solely responsible for, it does not necessarily represent the opinion of the European Community. Special thanks to F. Bachmann for his technical mechanical design support of the prototype.

References

1. H. Esen, K. Yano, and M. Buss. A control algorithm and priliminary user studies for a bone drilling medical training system. In *ROMAN*, 2003.
2. M. Buss and G. Schmidt. Control Problems in Multi-Modal Telepresence Systems. In P.M. Frank, editor, *Advances in Control: Highlights of the 5th European Control Conference ECC'99 in Karlsruhe, Germany*, pages 65–101. Springer, 1999.
3. J.D. Brederson, M. Ikits, C.R. Johnson, and C.D. Hansen. The visual haptic workbench. In *Proc. 5th PHANToM Users Group Workshop*, 2000.
4. C. Richard, A.M. Okamura, and M.R. Cutkosky. Getting a feel for dynamics: using haptic interface kits for teaching dynamics and control. In *ASME IMECE 6th Annu. Symp. on Haptic Interfaces*, Dallas, Texas, 1997.
5. G. Hirzinger, N. Sporer, A. Albu-Schäffer, M. Hähnle, R. Krenn, A. Pascucci, and M. Schedl. DLR's torque-controlled light weight robot iii - are we reaching the technological limits now? In *Proc. IEEE Int. Conf. Rob. Automat.*, pages 1710–1716, 2002.
6. M. Ueberle and M. Buss. Design, control, and evaluation of a new 6 DOF haptic device. In *Proc. IEEE/RSJ Int. Conf. on Intellig. Rob. and Syst.*, pages 2949–2954, 2002.
7. F.B.M. Duarte, J.A.T Machado, and L. Horváth. A trajectory planning algorithm for redundant manipulators. In *Proc. IEEE Int. Symp. Ind. Electr.*, pages 1002–1007, 1999.
8. P.K. Khosla and T. Kanade. Treal-time implementation of the computed-torque scheme. *IEEE Trans. Rob. Automat.*, 5(2):245–253, 1989.
9. A. Liégeois. Automatic supervisory control of the configuration and behaviour of multibody mechanisms. *IEEE Tans. on Syst., Man, and Cybern.*, 7(12):868–871, 1977.
10. Tsuneo Yoshikawa. Manipulability of robotic mechanism. *The Int. J. of Robotics Research*, 4(5):3–9, Summer 1985.
11. D.H. Gottlieb. Topology and robots. In *Proc. IEEE Int. Conf. Rob. Automat.*, pages 1689–1691, 1986.

An Experimental Study of the Limitations of Mobile Haptic Interfaces

F. Barbagli[1,2], A. Formaglio[1], M. Franzini[1], A. Giannitrapani[1], and
D. Prattichizzo[1]

(1) Dipartimento di Ingegneria dell'Informazione, University of Siena
(2) Stanford Robotics Lab, Stanford University
[barbagli, formaglio, anto, franzini, prattichizzo]@dii.unisi.it

Abstract. This paper presents various procedures that can be used in order to numerically evaluate what the maximum $Z-$width that can be rendered by a mobile haptic interface will be given few parameters that characterize the haptic device and the mobile platform that make up such interface. Such procedures are applied to the case of two different mobile haptic interfaces. Results are encouraging, even though limitations to the proposed procedure exist.

1 Introduction

The workspace of haptic interfaces varies largely on their design and usage, ranging from few planar centimeters of the Pantograph [2] to several cube meters of the Scaleable Spidar device [3]. Most haptic devices, however, share two main traits: they are grounded and they have limited workspace. While this is not a problem in many applications, it can become one in cases where users need to interact with large virtual environments while navigating inside of them.

A possible solutions for this problem is to use locomotion interfaces, i.e. treadmill-like interfaces that simulate some of the inertial feedback that a user would experience while navigating through a large virtual environment [5]. Another possible approach is to create haptic interfaces featuring unlimited workspace by combining mobile robots and standard grounded force-feedback devices. This type of interface, which in part resembles the *cobot* [4], was introduced by Nitzsche et al [1] and is referred to as *mobile haptic interface*. In this paper we focus our attention on the latter type of interfaces (see Fig. 1 for two examples of mobile haptic interfaces).

Haptic interfaces are often described as mechanical impedance transducers. Mobile haptic interfaces are no exceptions, and are thus required to render impedances ranging from zero, simulating movement in free space, to a maximum value Z_{max} that depends on a multitude of factors [6]. The $Z-$width characterizing a mobile haptic interface depends on factors characterizing its two main constitutive subsystems.

The main contribution of this paper is to attempt to provide some useful guidelines on how to select the right combination of a desktop haptic device and a mobile platform in order to design a mobile haptic interface that will have a certain desired $Z-$width. More specifically we will focus our attention on how to correctly simulate

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 533–542, 2006.

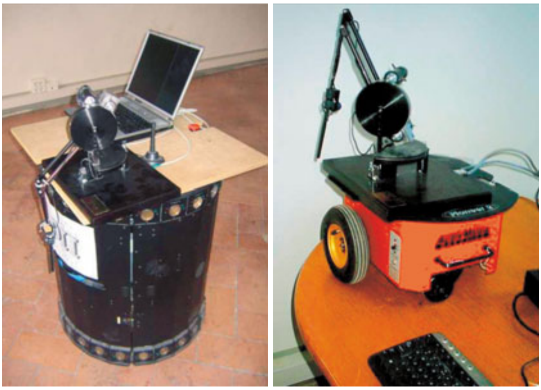


Fig. 1. Two mobile haptic interfaces: a general purpose interface based on the holonomic Nomad mobile robot and a Phantom Premium 1.5; a more limited interface based on a non-holonomic Pioneer mobile robot and a Phantom Premium 1.5

free-space movement, i.e. $Z_{min} \simeq 0$. In order to do so we will first introduce a simplified model of a mobile haptic interface (Section 2); we will then use such model to analyze the response of a mobile haptic interface to a given set of input signals that most closely resemble those of the users (Section 3); finally we will compare such results to a set of experimental results obtained on different mobile haptic interfaces (Section 4). While the simplicity of the proposed model may lead to some error in our analysis, its usage allow the results obtained to be easily interpretable.

2 Modelling mobile haptic interfaces

Mobile haptic interfaces (MHI) are made up of two main components: a mobile platform (MP), such as the holonomic Nomad or the non-holonomic Pioneer mobile robots in Fig. 1; an impedance-type haptic device (HD), such as the Phantom device in Fig. 1, grounded to the mobile platform. The former robots typically feature high inertia and are normally position controlled. The latter normally feature low inertias and high levels of transparency and are normally controlled in force.

In order to transparently render any impedance inside an unlimited environment we propose a simple control algorithm, which mimics the one proposed by Nitzsche et al. in [1]. The basic idea is that the MP should be position controlled to track the movements of the operator, thus allowing the HD to always be in the center of its workspace. This has two main advantages: it allows the HD to render forces on the operator in a configuration of maximum structural stiffness; and it allows the HD to never reach its workspace limits, a situation in which spurious forces would be felt by the operator thus destroying the overall sense of transparency. On the other side, forces are rendered by the HD using standard constrained based methods such as the proxy algorithm [7].

Let Σ_W be the base reference frame which is attached to the world and Σ_M the reference frame attached to the mobile platform base.

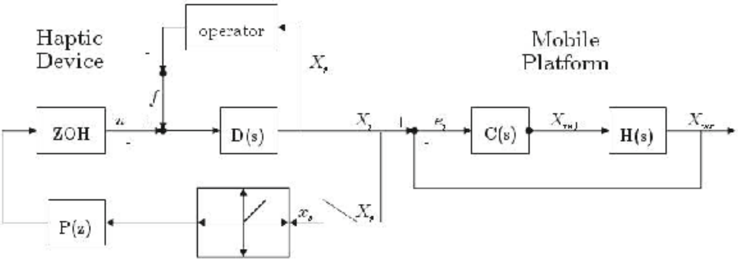


Fig. 2. Control scheme for a mobile haptic interface: the left side of the picture represents the interaction force rendering algorithm while the right side represents the position control algorithms for the mobile platform.

In order to control the position of the MP we propose a simple PD controller as depicted in Fig. 2 (right side), where X_p represents the position of the HD end effector with respect to Σ_W, X_{mr} the position of the MP with respect to Σ_W, e_p is the position of the HD end effector with respect to Σ_M (as well as the tracking error for the mobile platform), $X_{ref}(s) = C(s)e_p(s)$ is the position commanded to the MP, $C(s)$ is the transfer function of the MP control algorithm, and $H(s)$ is the transfer function representing the MP.

In order to control the interaction forces between user and virtual environments we use a classic scheme that was introduced by Colgate in [6], see Fig. 2 (left side) where X_p has been defined above, $D(s)$ is the haptic device transfer function, $P(z)$ is the discrete time transfer function of the virtual object impedance, and ZOH is a zero holder hold. Note that while this simplified model is normally used to represent virtual walls it can be generalized to more complex cases by adding a collision detection block.

The MP can be modelled as a mass-damping system subject to actuator forces, which are modelled using a spring. Thus in the case of the MP we have $F_{mr}(s) = K_m X_{ref}(s) - B_m s X_{mr}(s)$ yielding transfer function

$$H(s) = \frac{K_m}{s(M_m\, s + B_m)} = \frac{X_{mw}}{X_{ref}}. \tag{1}$$

While this is far from being an exact dynamical model of a mobile robot and cannot account for non-linear effects that are present in reality, it has the advantage of being simple and of being characterized by a small set of parameters that can easily be interpreted and identified. Both MPs that were employed during the experimental phase have been modelled using this approach.

A PD controller is chosen in order for the platform to track the position of the user with respect to the world, i.e. to bring e_p to zero. Thus $C(s) = sD_m + P_m$.

The HD can also be modelled as a mass-damper system such that

$$D(s) = \frac{1}{s(M_h s + B_h)}$$

and similarly the impedance model of a virtual object can be chosen as a discrete spring-damper system, yielding $P(z) = D_h \frac{z-1}{zT} + P_h$ where T is the haptic servo-loop period.

3 Performance of a mobile haptic interface

The $Z-$width of a MHI depends on how MP and HD are controlled. Standard HD are typically designed in order to have low reflected inertias and friction, i.e. $Z_{min} \simeq 0$, while Z_{max} normally depends on the servo rate at which the device can be commanded [6]. The addition of a MP, whose inertia is usually fairly large and whose dynamics are normally slower than those of the HD, has little implication on Z_{max} but major implications on Z_{min}. When a user is slowly examining a virtual object by touching it with the HD the MP basically does not move, and thus Z_{max} is determined by the HD alone. When a user moves in free space, on the other side, the MP may lag behind the HD, leading the user to feel spurious forces due to the HD workspace limitations, i.e. experiencing a $Z_{min} > 0$. In such case the control scheme on Fig. 2 ceases to be valid and u is determined by the contact with the workspace limits of the HD.

The goal of this section is to analytically determine which free space motions of the user will result in a correct impedance being rendered by the MHI. Thus we seek to determine X_p such that $e_p < X_{ws}$ with X_{ws} workspace limits of the HD. Such analysis will focus on three main types of inputs that closely resemble an operator's movements: step input, ramp input, and sinusoidal input. It is important to note that our analysis only applies to steady-state behavior of the system. This does not cover all possible scenarios, since during transients the HD could reach its workspace limits, and thus the analytical results proposed in the following sections must be considered as rough indicators of performance.

3.1 Step displacement

What is the maximum step displacement $X_p(t) = A_g$, $(t \geq 0)$, that can be correctly rendered by the MHI? Let us consider an ideal step signal of amplitude A_g. At time $t = 0^+$ we have

$$X_{mr}(0^+) = 0 \quad \text{and} \quad X_p(0^+) = A_g$$

i.e. while the system output is still at zero the input has jumped to A_g. In this case $e_p(0^+) = A_g$, i.e. the haptic device is at a A_g distance from the center of its workspace, and thus it is necessary [1] that the maximum amplitude of a step position signal applied to a MHI is such that $A_g < X_{ws}$.

3.2 Ramp displacement

What is the maximum ramp displacement $X_p = V_R t$ that can be correctly rendered by the MHI? Let us consider an ideal ramp $X_p = V_R t$. The MP model of Fig. 2

[1] but not sufficient due to the lack of knowledge of what happens during transients

follows such input with a finite steady-state error, since $H(s)$ has one pole in the origin. Such error is given by

$$e_p(\infty) = \lim_{s \to 0} \left(\frac{1}{sC(s)H'(s)\frac{1}{s}} \right) V_R = \frac{V_R}{K_v}$$

where $H(s) = H'(s)/s$. Note however that the maximum error between HD and MP positions e_{max} can be reached during the transient and can be found using

$$e_{po} \triangleq \frac{e_{max} - e_p(\infty)}{e_p(\infty)}$$

where e_{po} does not depend on the slope of the ramp but only on H(s) and C(s). Given that e_{max} is proportional to V_R, since $e_{max} = (1 + e_{po})\frac{V_R}{K_v}$, by experimentally determining the value of e_{po} for a given MP, we can then compute e_{max}. In order for $e_{max} < X_{ws}$ to be true, when applying $X_p(t) = V_R t$, V_R must be such that

$$V_R < \frac{X_{ws} K_v}{(1 + e_{po})} \tag{2}$$

3.3 Sinusoidal displacement

What is the maximum amplitude/frequency of $X_p = A_s sin(\omega t)$ that can be correctly rendered by the MHI? Referring to Fig. 2 it is possible to compute $G(s)$ such that $e_p(s) = G(s)X_p(s)$. Given the linearity of the overall system we have

$$e_p = A_s \|G(j\omega)\| sin(\omega t + \angle(G(j\omega)))$$

and thus in order for $e_p \in (-X_{ws}; X_{ws})$, inequality $A_s\|G(j\omega)\| < X_{ws}$ must hold. Since $G(s)$ has high-pass filter behavior, higher-frequency sinusoids must have lower amplitude in order for the MHI to track them and viceversa.

In other terms it is always possible to compute a region of the (ω, A_s) plane, $\mathcal{I} = \{(\omega, A_s)/(\omega, A_s) : A_s\|G(j\omega)\| < X_{ws}\}$, that represents the sinusoidal inputs that can be correctly rendered by a MHI. Moreover curve $\gamma : A_s\|G(j\omega)\| = X_{ws}$ representing the border between $\mathcal{I}$ and the rest of the (ω, A_s) plane can be numerically computed.

4 Experimental results

In order to validate the theoretical results presented in Section 3, several experimental tests have been carried out, using the Phantom Premium 1.5 haptic interface and the mobile platforms Pioneer2 DX and Nomad XR4000. The latter is a fully holonomic robot, featuring high levels of inertia, whereas the former is an agile differential drive (non-holonomic) robot.

A preliminary set of experiments has been conducted to identify the model parameters of each robot. For this purpose, several sets of input-output data $\{X_{ref},\ X_{mr}\}$, corresponding to different kinds of input signals (square waves, ramps, sinusoids) have been collected, and the values of the parameters K_m, M_m, B_m characterizing the transfer function $H(s)$ (see Section 2) have been tuned, by comparing the actual and simulated outputs. Once the robot models were available, a suitable PD controller has been designed to guarantee closed-loop stability and to achieve satisfactory tracking performance.

Two experimental setups have been considered, combining the HD with each MP. All tests have been performed along a single degree of freedom and for each experimental trial, the actual tracking error has been compared to the one predicted by the corresponding model. As pointed out in Section 2, the tracking error e_p corresponds to the displacement of the end-effector w.r.t. the center of its workspace, thus being directly available from the readings of the haptic interface encoders.

In order to verify the ability of the models to predict whether a given input X_p, belonging to one of the signal classes considered in Section 3, would cause the end-effector to reach the limit of its workspace, several signals within each class have been applied to the haptic device end-effector. During all the experimental trials, it was supposed that the maximum desired displacement of the end-effector from the center of its workspace was $X_{ws} = 150\ mm$. In Fig. 4 and 5, the results of typical experimental tests, involving the Pioneer2 DX and the Nomad XR4000 MP, respectively, are shown.

4.1 Step displacement

In a first set of experiments, step displacements $X_p(t) = A_g$, $(t \geq 0)$, with different amplitudes, have been used as reference signals. The ideal step has been generated by fixing the haptic device's end-effector to a given position A_g to the side of the center of the workspace, before the robot starts the tracking. As correctly predicted by the models, the results obtained with both MHI configurations confirmed that as long as the amplitude of the Step displacements is inside the workspace limit, the end-effector will never reach its maximum extension (see Fig. 4(a) and 5(a)).

4.2 Ramp displacement

Secondly, Ramp displacements $X_p(t) = V_R t$ have been considered. In order to move the haptic device's end-effector with a constant velocity V_R, a second mobile robot was used. Thus, while the MHI stood still, another mobile robot was accelerated in order to reach a desired velocity V_R (see Fig. 3) and then hooked up to the MHI end-effector (through a velcro connection), thus exciting the MHI with the desired reference signal. With this kind of inputs, the behavior of the two MHI configurations is significantly different. As far as the Pioneer platform is concerned, it turns out that the maximum error is reached during the transient (see Fig. 4(b)-4(c)), whereas the MHI employing the Nomad robot reaches its maximum tracking error at the steady state (see Fig. 5(b)-5(c)). This is due to the different controller parameters, which make the Pioneer platform exhibit a remarkable overshoot. The maximum tracking error of the Nomad MHI is almost proportional to the input velocity, as predicted by the theoretical analysis, contrariwise to the Pioneer behavior, manifesting nonlinear phenomena.

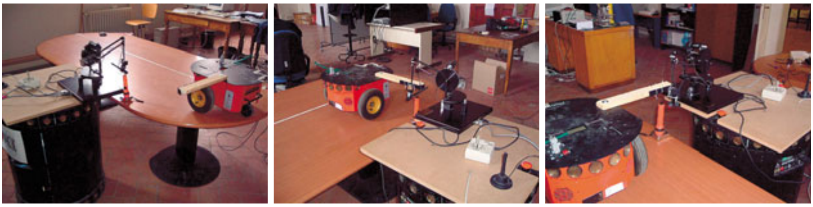


Fig. 3. A mobile robot is used to excite the MHI with a ramp signal: (left) the mobile robot is accelerated to reach V_R velocity; (middle) the mobile robot comes into contact with the MHI end effector; (right) the MHI end effector starts moving.

4.3 Sinusoidal displacement

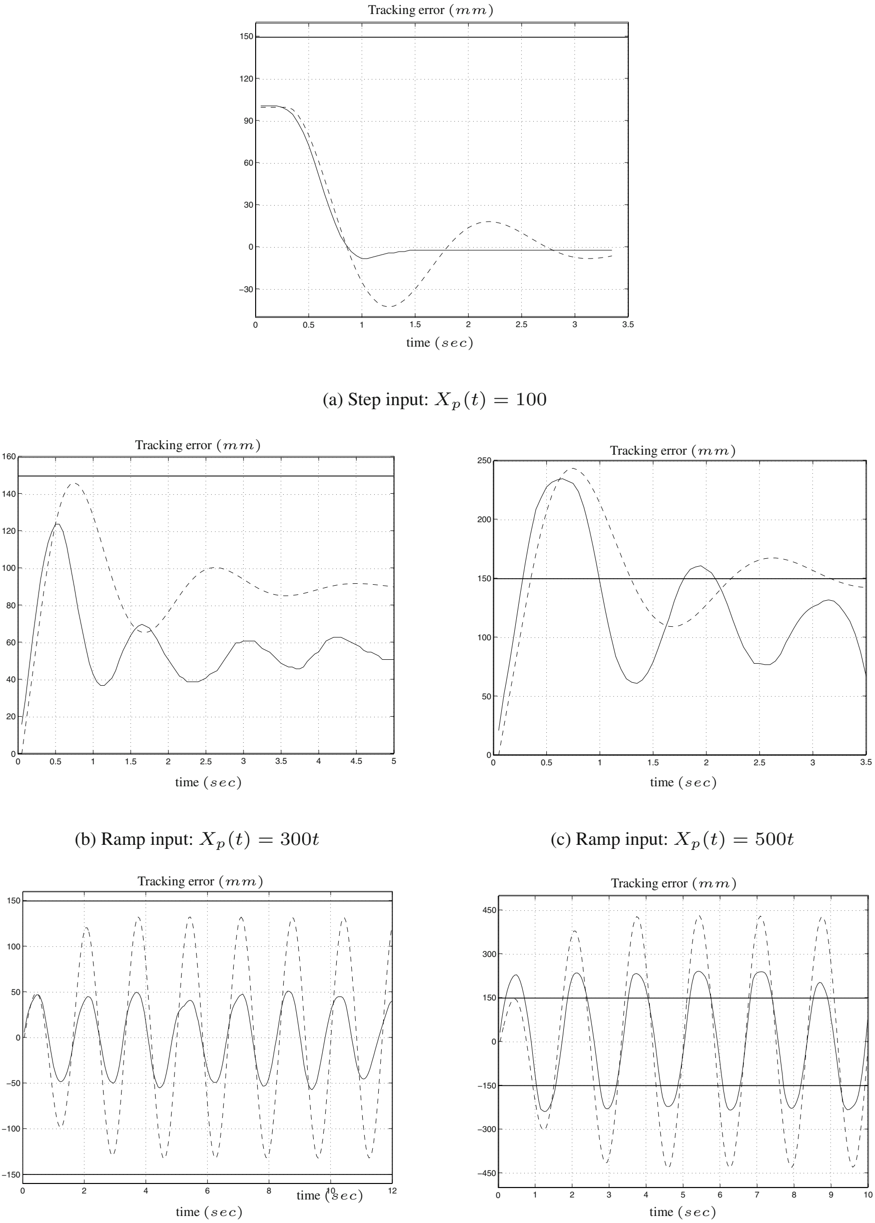
The last set of experiments have been performed with sinusoidal inputs $X_p(t) = A_s \sin(\omega t)$, featuring different amplitudes A_s and frequencies ω . Such reference signals have been generated by a human operator who, with the aid of periodic acoustic and visual stimuli, moved the haptic device's end-effector sideways, approximately describing a time dependent sinusoid.

The experimental results obtained with sinusoidal inputs support the intuition that, in order to confine the tracking error within the workspace limits, the maximum admissible amplitude is roughly inversely proportional to the sinusoid frequency. In this case, the maximum tracking error predicted by the Pioneer MHI is significantly bigger than the actual one, thus resulting in more conservative constraints on the admissible sinusoidal inputs (see Fig. 4(d)-4(e)). This means that a more extensive model identification phase is needed. On the contrary, the maximum predicted and actual error for the Nomad MHI are very close (5(d)-5(e)). Due to the inaccuracy of the man-made sinusoidal references as well as nonlinear phenomena, the actual tracking error may exhibit trends and/or non-zero mean.

In summary, the performed experimental phase confirmed the viability of the presented theoretical analysis, showing that it may serve as a useful tool for the evaluation of MHI's performance limitations. Despite the difficulty to accurately reproduce the time evolution of the tracking error, the procedures proposed in this paper were able to correctly predict the outreach of the HD workspace.

It is worth noticing that, due to the simple structure of the chosen robot model, the resulting theoretical limitations on the trackable input signals may be somewhat conservative. Of course, a noticeable improvement would be achieved by adopting more sophisticated robot motion models and carrying out more extensive parameter identification campaigns.

We conclude this section pointing out some technical problems encountered during the experimental phase.



(a) Step input: $X_p(t) = 100$

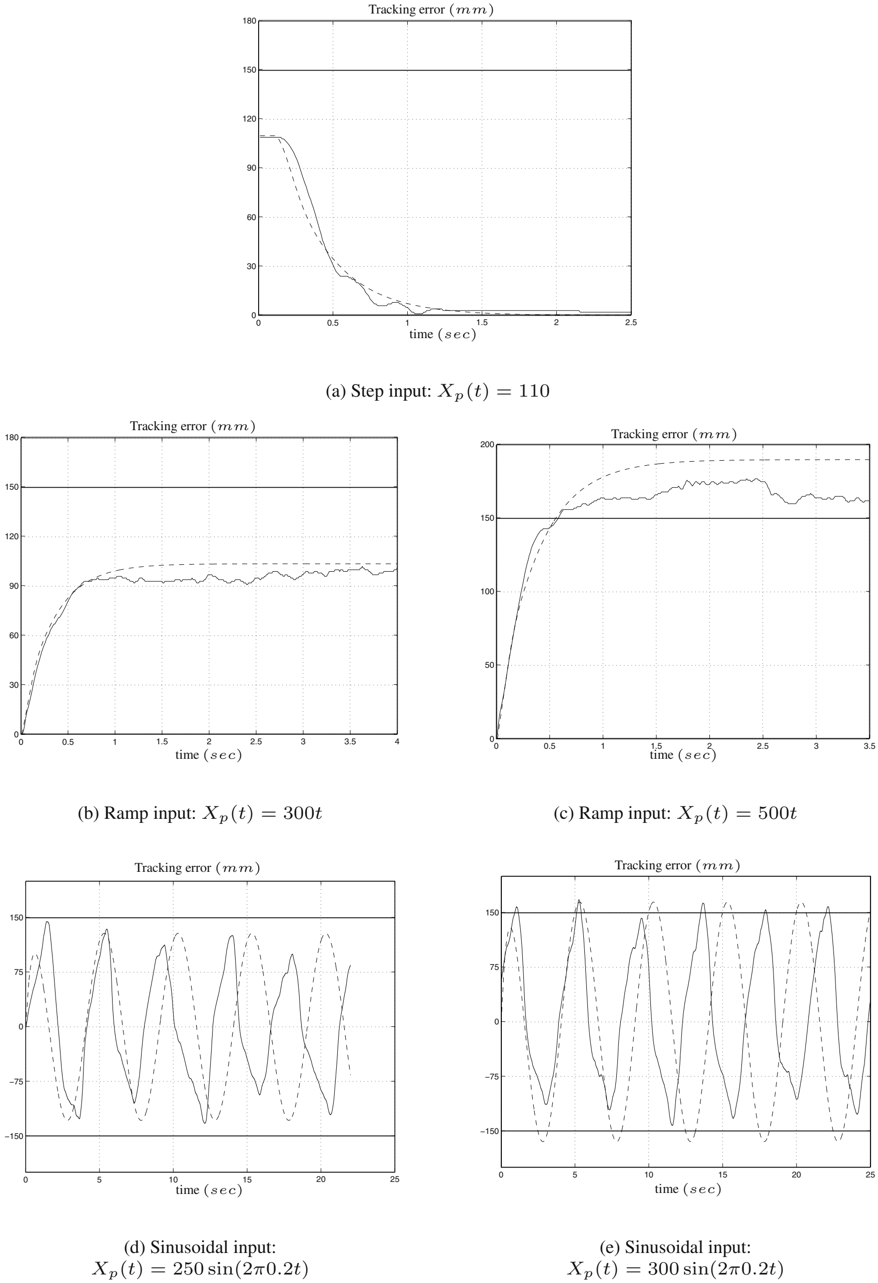
(b) Ramp input: $X_p(t) = 300t$

(c) Ramp input: $X_p(t) = 500t$

(d) Sinusoidal input:
$X_p(t) = 50\sin(2\pi 0.6t)$

(e) Sinusoidal input:
$X_p(t) = 150\sin(2\pi 0.6t)$

Fig. 4. Experimental results with the Pioneer2 DX mobile platform and different input signals: actual (solid line) and predicted (dashed line) tracking error. Thick solid lines represent the desired maximum error, due to workspace limits.



(a) Step input: $X_p(t) = 110$

(b) Ramp input: $X_p(t) = 300t$

(c) Ramp input: $X_p(t) = 500t$

(d) Sinusoidal input:
$X_p(t) = 250\sin(2\pi 0.2t)$

(e) Sinusoidal input:
$X_p(t) = 300\sin(2\pi 0.2t)$

Fig. 5. Experimental results with the Nomad XR4000 mobile platform and different input signals: actual (solid line) and predicted (dashed line) tracking error. Thick solid lines represent the desired maximum error, due to workspace limits.

- The generation of accurate sinusoidal reference signals by means of a MP was not possible due to spurious movements during the inversions of the motion direction. In these situations the Nomad platform suffers from the problem of sudden wheel steering, while the requested acceleration makes the Pioneer robot slip. The same undesirable phenomena are experienced when the MHI tries to track a man-made sinusoidal input.

- The range of testable reference signals is severely limited by a number of factors. First, the movements of the robots are constrained by the length of the Phantom Premium 1.5 cables (roughly 2 meters). Secondly, for safety reasons, special caution must be used when using the Nomad platform, which basically results in limiting the maximum admissible speed of the reference signals. Finally, special care should be devoted for preventing the controller to excite un-modelled dynamics or inevitable nonlinearities (e.g., actuator saturations).

- The communication between the haptic device and the MP occurs via serial link (Pioneer robot) or Ethernet connection (Nomad robot). This introduces significant delays and degrades the tracking performance.

5 Conclusion and future work

This paper presents various procedures that can be used to pre-evaluate if the minimum impedance that can be rendered by a MHI can be made equal to zero. The proposed procedures are shown to match the real behavior of two MHI with different features. However such procedures have limitations, mainly due to the fact that they can only account for steady state behavior and that MHIs are modelled as linear systems. Thus the proposed procedures should only be considered as qualitative indicators of the likely performance of an MHI, and not as exact ones.

References

1. N. Nitzsche, U.D. Hanebeck, and G. Schmidt “Desgin Issues of Mobile haptic Interfaces”, *Journal of Robotic Systems*, vol. 20:9, pp. 549-556, 2003.
2. C. Ramstein and V. Hayward, “The Pantograph: A Large Workspace Haptic Device For A Multi-Modal Human-Computer Interaction”, *CHI’94, Conference on Human Factors in Computing Systems ACM/SIGCHI Companion-4/94*, pp. 57-58, 1994.
3. L. Bouguila, M. Ishii and M. Sato, “Scaleable SPIDAR: A Haptic Interface For Human-Scale Virtual Environments”, *Haptic Human-Computer Interaction*, pp. 182-193, 2000.
4. M. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. Moore, B. Gillespie and P. Akella, “Cobot Architecture”, *IEEE Tr. on Robotics and Automation*, 17(4):377-390, 2001.
5. J. M. Hollerbach et al., “Simulating side slopes on locomotion interfaces using torso forces,” *Haptic Symposium*, pp. 91-98, March 22-23, 2003.
6. J. E. Colgate and J. M. Brown, “Factors Affecting the Z-Width of a Haptic Display”, in *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 3205-10, San Diego, CA, May 1994.
7. D. C. Ruspini, and K. Kolarov and O. Khatib, “The Haptic Display of Complex Graphical Environments”, Siggraph97, pp. 345–352, 1997.

Robust Haptic Teleoperation of a Mobile Manipulation Platform

Jaeheung Park and Oussama Khatib

Stanford AI Laboratory
Stanford University
http://robotics.stanford.edu

Abstract. This paper presents a new teleoperation scheme for mobile manipulation systems. A virtual spring connects a master and slave system where local force feedback on the slave system compensates for the dynamics of the mobile manipulator. This scheme is robust in the presence of time delays and disturbances and provides good tele-presence. The slave manipulator is mounted on a mobile base and controlled within the operational space framework. In this framework, the redundancy of the mobile manipulation system is addressed through a decoupled decomposition of task and posture dynamics.

1 Introduction

Tele-manipulation provides teleprescence by allowing a user to remotely control a slave robot through a master device. These systems offer great potential, however, connecting master/slave stations in a coherent way is a challenging task. While the master station is controlled by the human operator, the slave station often interacts with an unknown and dynamic environment. The nature of such interaction greatly influences overall system performance.

Many teleoperation schemes have been developed to improve telepresence and stability when position and force measurements are available on both the master and slave [7,8,6]. Most of these schemes provide direct contact force feedback to the user for a greater level of telepresence. However, this feedback degrades the stability of the overall system especially if the mass properties of the master and slave differ significantly [3]. This paper proposes a new teleoperation scheme that utilizes local force control while exchanging position information between the master and slave station. Such an approach provides much better stability characteristics, while offering the user a realistic feeling of teleprescence through the use of a virtual spring. This architecture is especially suited for systems where force sensing is limited to the slave robot and when the master device is relatively light and frictionless. Specifically, our setup which uses a Phantom device and PUMA mounted on a mobile base meets this criteria.

The operational space formulation [5] decouples the dynamics of the mobile manipulator into end-effector task dynamics and posture dynamics. Furthermore, each end-effector DOF can be independently controlled. The control of the base can be separately synthesized since its dynamics are decoupled from that of the end-effector. Based on this formulation, our new teleoperation scheme is applied for each end-effector DOF.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 543–554, 2006.

To deal with uncertainties and time-varying parameters (e.g. dynamic environments), the force control on the slave robot uses Active Observers (AOBs) [1], which modify the Kalman estimation structure to achieve model-reference adaptive control. In this case, the stiffness parameter has to be identified on-line to improve robustness and telepresence. The AOB is designed to cover a medium range of stiffness values. However, for large variations, on-line stiffness estimation is necessary [2]. Time delay associated with the wireless LAN network is also analyzed for our system.

2 Control for a Mobile Manipulator

The equations of motion for the end-effector of a robotic manipulator can be described using the operational space formulation [5],

$$\Lambda(q)\dot{\vartheta} + \mu(q,\dot{q}) + p(q) + F_c = F \tag{1}$$

where $\Lambda(q)$, $\mu(q,\dot{q})$, and $p(q)$ are the inertia matrix, the vector of Coriolis/centripetal forces, and the gravity vector in operational space, respectively. The term ϑ denotes the instantaneous velocity in operational space coordinates. The term F_c is the contact force at the end-effector. The control torque is selected as,

$$\Gamma = J^T F + N^T \Gamma_0 \tag{2}$$

$$F = \hat{\Lambda} F^* + \hat{\mu} + \hat{p} + \hat{F}_c \tag{3}$$

where N^T is the dynamically consistent null space projection matrix and F^* is the command to the unit mass system. The $\hat{\cdot}$ indicates an estimate of a particular quantity. The following decoupled equations of motion for the end-effector are obtained.

$$\dot{\vartheta} = F^* \tag{4}$$

The command F^* is composed of force and motion control components that are projected by the selection matrices, Ω_f and Ω_m, respectively.

$$F^* = \Omega_f F_f^* + \Omega_m F_m^* \tag{5}$$

In the experimental setup, force control is used to control only the Cartesian position of the end-effector since the master device does not provide force feedback on the orientation. The selection matrices are

$$\Omega_f = \begin{bmatrix} I_3 & 0_3 \\ 0_3 & 0_3 \end{bmatrix}, \qquad \Omega_m = \begin{bmatrix} 0_3 & 0_3 \\ 0_3 & I_3 \end{bmatrix} \tag{6}$$

where I_3 is the 3×3 identity matrix and 0_3 is the 3×3 zero matrix.

The control of the mobile base is applied to Γ_0 in Equation (2). The dynamically consistent null space projection matrix N^T prevents control of the mobile base from affecting the end-effector control.

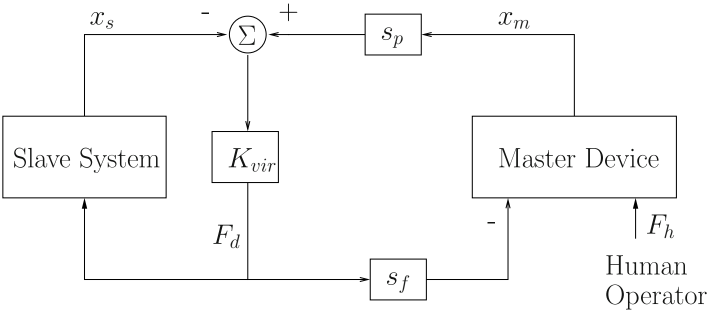


Fig. 1. Teleoperation Scheme. x_m, x_s, s_p and s_f are the master position, slave position, position scaling and force scaling, respectively. s_p and s_f are 2.0 and 0.1 in the experimental setup. K_{vir} is the virtual spring that generates the desired force, F_d.

3 Teleoperation

The teleoperation scheme is developed for each direction in operational space coordinates (i.e. one DOF system) since the control structure in Section 2 enables each Cartesian direction of a manipulator end-effector to be controlled independently. Figure 1 illustrates the proposed teleoperation scheme. The proposed scheme is simply to connect the master and slave system with a virtual spring. Then, force control is used on the slave manipulator to eliminate the dynamics of the slave robot.

The desired force, F_d, for both master and slave systems is generated by the virtual spring K_{vir} due to the position error. The contact force on the slave end-effector is controlled to track the desired force, F_d. The force controller in the slave system is implemented using a modified Kalman Estimator with full state feedback(AOB). However, only feedforward control is used to generate the desired force on the master side since the device is light weight and has low friction.

Stability characteristics of the system are improved by providing the desired contact force to the operator rather than the measured contact force. The direct use of the measured contact force causes a delay in the loop and the stability of the system is greatly dependent upon the mass ratio of the master and slave systems [3].

3.1 Telepresence

The user is always provided with the contact force that is scaled by s_f through a haptic device if the force control in the slave robot tracks the desired contact force well. Moreover, the transfer function, $\frac{X_m(s)}{F_h(s)}$, from the force of a human operator to the master position represents the compliance that the human operator feels at the master device [7]. Telepresence would be realized if the transfer function closely matches the slave system compliance.

In Figure 1, the master device is modeled with a mass and damper system, having the transfer function of $1/(m_m s^2 + c_m s)$. The slave system represents the force controlled robot in contact with environment; thus, the transfer function from the desired force to the slave position, $\frac{X_s(s)}{F_d(s)}$ is represented by $G_{se}(s)$. The equations of motion for the master and slave are

$$(m_m s^2 + c_m s)X_m(s) = F_h(s) - s_f K_{vir}(s_p X_m(s) - X_s(s)) \tag{7}$$
$$G_{se}(s)K_{vir}(s_p X_m(s) - X_s(s)) = X_s(s), \tag{8}$$

where $X_m(s)$ and $X_s(s)$ are the Laplace Transform of x_m and x_s. Moreover, the environment on contact is modeled to have a certain stiffness, K_s,

$$F_c = K_s x_s, \tag{9}$$

where F_c is the contact force with the environment. Therefore, $G_{se}(s)$ can be represented by

$$G_{se}(s) = \frac{X_s(s)}{F_d(s)} = \frac{1}{K_s}\frac{F_c(s)}{F_d(s)}. \tag{10}$$

$G_s(s) = \frac{F_c(s)}{F_d(s)}$ is the closed loop transfer function of the force control in the slave system; thus, $G_{se}(s) \approx \frac{1}{K_s}$ within the bandwidth of the force controller. With (7) and (8), the transfer function $\frac{X_m(s)}{F_h(s)}$ can be derived as

$$\frac{X_m(s)}{F_h(s)} = \frac{K_{vir}+G_{se}(s)^{-1}}{(m_m s^2+c_m s+K_{vir}s_p s_f)(K_{vir}+G_{se}(s)^{-1})-K_{vir}^2 s_p s_f}\,. \tag{11}$$

Equation (11) shows the characteristics of the proposed teleoperation scheme. If $K_{vir} \gg |G_{se}^{-1}|$, at a low frequency range, $|m_m s^2 + c_m s| \ll K_{vir}s_p s_f$, the compliance that a human operator feels will be close to the environment compliance, $\frac{X_m(s)}{F_h(s)} \approx \frac{1}{s_p s_f K_s}$. At a high frequency range, $|m_m s^2 + c_m s| \gg K_{vir}s_p s_f$, it will be $\frac{X_m(s)}{F_h(s)} \approx \frac{1}{m_m s^2 + c_m s}$.

Therefore, the key aspect for telepresence is to maintain $K_{vir} \gg |G_{se}^{-1}|$, i.e. $K_{vir} \gg K_s$, in addition to having a large force control bandwidth. The value of K_{vir} is limited by the stability. To maintain the ratio $\frac{K_{vir}}{K_s}$ as large as possible within this limit, K_{vir} is updated on-line based upon the estimated environment stiffness. That is, K_{vir} is increased with the estimate of K_s. The following equation is used in the experiments.

$$K_{vir} = 2000.0\sigma_d(0.007(\hat{K}_s - 1000.0)) + 1000.0 \tag{12}$$

where $\sigma_d(x) = \frac{1}{1+e^{-x}}$.

3.2 Stability

The characteristic equation Δ of the loop is

$$\Delta = (m_m s^2 + c_m s)G_{se}(s)^{-1} + K_{vir}(m_m s^2 + c_m s + s_p s_f G_{se}(s)^{-1}). \tag{13}$$

The system is stable for any K_{vir} if the model is perfect because $G_{se}(s)$ is a stable minimum system with a constant DC value. However, the feedback gains K_{vir}

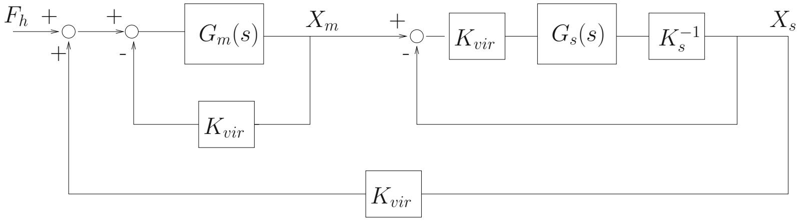


Fig. 2. Re-arranged block diagram. It highlights local feedback for each system in the proposed Teleoperation Scheme. $G_m(s)$ and $G_s(s)$ represent dynamics of the master device and the closed loop force control system of the slave manipulator. i.e. $G_m(s) = \frac{X_m(s)}{F(s)} = (m_m s^2 + c_m s)^{-1}$ and $G_s(s) = \frac{F_c(s)}{F_d(s)}$. X_m and X_s are the position of the master and slave system. F_h is the force from a human operator. F_c and F_d are the contact force and the desired contact force in the slave manipulator. K_s is the environment stiffness.

and K_{vir}/K_s are bounded by the physical limitations of the master device and the slave robot. Figure 2 shows the local feedback systems at the master and slave. Specifically, K_{vir} cannot exceed the maximum stiffness that the master device can produce. Also, $\frac{K_{vir}}{K_s}$ is limited by the motion bandwidth of the slave manipulator. In free space motion, where K_s is small, this results in greater limitation on the magnitude of K_{vir}.

3.3 Time delay

In the presence of time delay Equations (7) and (8) become

$$(m_m s^2 + c_m s)X_m(s) = F_h(s) - s_f K_{vir}(s_p X_m(s) - X_s(s)e^{-T_d s}) \quad (14)$$

$$G_{se}(s)K_{vir}(s_p X_m(s)e^{-T_d s} - X_s(s)) = X_s(s) \quad (15)$$

Now, the transfer function from the force of the human operator to the position of the master device is

$$\frac{X_m(s)}{F_h(s)} = \frac{K_{vir} + G_{se}^{-1}}{(m_m s^2 + c_m s + K_{vir} s_p s_f)(K_{vir} + G_{se}^{-1}) - K_{vir}^2 s_p s_f e^{-2T_d s}} . \quad (16)$$

The effect of time delay on the performance is investigated using Padé approximation for small time delay, $e^{-2T_d s} = \frac{1 - T_d s}{1 + T_d s}$.

$$\left[\frac{F_h(s)}{X_m(s)}\right]_{\text{w delay}} = \left[\frac{F_h(s)}{X_m(s)}\right]_{\text{w/o delay}} + \frac{K_{vir}^2 s_p s_f \frac{2T_d s}{1+T_d s}}{K_{vir} + G_{se}(s)^{-1}} \quad (17)$$

The additional term, $\frac{K_{vir}^2 s_p s_f \frac{2T_d s}{1+T_d s}}{K_{vir} + G_{se}(s)^{-1}}$, can be further approximated as $2sT_d K_{vir} s_p s_f$ at a low frequency range when $K_{vir} \gg |G_{se}^{-1}|$. This shows the damping effect of the time delay. Stability is no longer guaranteed for large T_d.

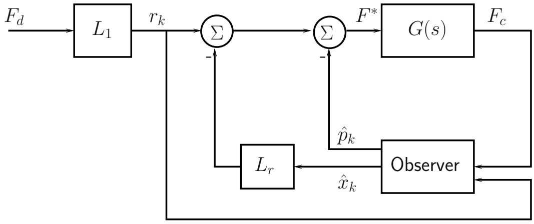


Fig. 3. Force Control Design. $G(s)$ is the system transfer function from the command F^* to the contact force F_c. F_d is the desired contact force. r_k, $\hat{x}_k$, and $\hat{p}_k$ are reference input, state estimate, and input error estimate. L_r and L_1 are a full state feedback gain and a scaling factor to compute reference input r_k.

4 Force Control

The decoupled unit mass system for each Cartesian direction (4) is used for force controller design. With the contact model (9), the equations of motion of contact force for each direction in operational space are,

$$\ddot{F}_c = K_s F^* \tag{18}$$

The system transfer function for contact force control is derived from a decoupled sub-system (18). With an additional damping, $K_v\vartheta$, to F^* for better stability and a system input delay, $T_{input,d}$, the overall system can be approximated by

$$G(s) = \frac{K_s e^{-sT_{input,d}}}{s(s+K_v)}, \tag{19}$$

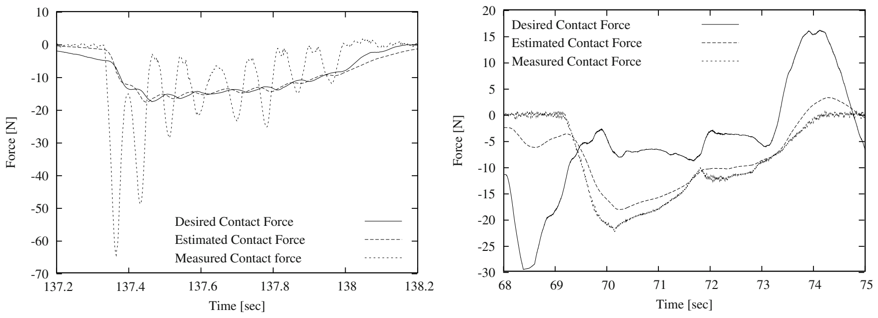
where K_v is a positive scalar. The discretized state space form of Equation (19) is used for discrete Kalman estimation and control.

The overall force control scheme is illustrated in Figure 3. The AOB uses a probabilistic approach to estimate the states and input error. The estimated input error is directly compensated at the input command. A full state feedback is applied with the estimated states. More details on the implementation and robustness analysis can be found in [2].

4.1 Stiffness Adaptation

The slave manipulator in teleoperation experiences contact with different environments. The knowledge of the stiffness, K_s, is important not only for the force control but also for modifying the virtual spring, K_{vir}, to provide better telepresence to an operator. The changes can be abrupt and large in magnitude; thus, a fast on-line stiffness estimation strategy is required to cope with these changes.

The performance of force control without adaptation degrades in the presence of a large mismatch of the environment stiffness, as shown in Figure 4. Also, the different relationships between the desired, measured, and estimated contact force (F_d, F_m,



(a) (b)

Fig. 4. The results of force control without adaptation. (a) Nominal stiffness, $\hat{K}_s$, is 100 [N/m] and K_s changes from free space to 3000 [N/m]. (b) Nominal stiffness, $\hat{K}_s$ is 3000 [N/m] and K_s changes from free space to 300 [N/m].

and F_e) in Figure 4(a) and (b) are noticeable. In Figure 4(a), where $K_s \gg \hat{K}_s$, the difference between F_m and F_e is large while the difference between F_d and F_e is relatively small since the system response starts fluctuating (i.e., F_m oscillates around F_e). In Figure 4(b), where $K_s \ll \hat{K}_s$, the difference between F_d and F_e is larger than between F_m and F_e since the system response is sluggish.

In addition, it is observed that as the larger contact force is applied, the system stiffness increases. With these observations from the experiments, an adaptation law has been proposed [4]. This adaptation law combines the aforementioned two ideas; thus, it has two parts.

$$\hat{K}_s^i = \hat{K}_{s,1}^{f,i} + \hat{K}_{s,2}^{f,i}. \tag{20}$$

where the superscript i indicates the discrete time step and the superscript f indicates the filtered value. The first part of the estimation is based upon the relation between F_d, F_m, and F_e.

$$\hat{K}_{s,1}^i = \hat{K}_{s,1}^{i-1} + \varDelta\hat{K}_{s,1}^i, \tag{21}$$

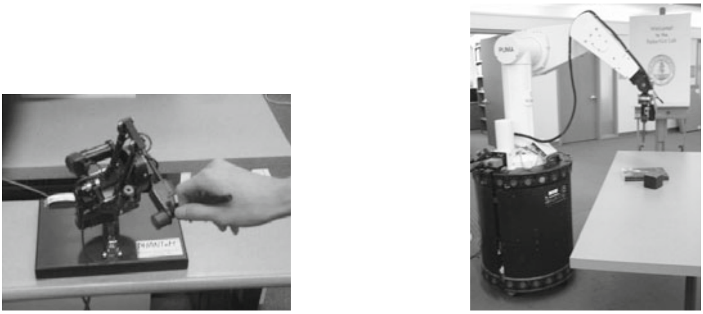
where $\varDelta\hat{K}_{s,1}^i = k_1|F_m - F_e|\,\sigma_d\left(c, \frac{|F_m - F_e|}{|F_e|+a_1} - b_1\right) - k_2|F_d - F_e|\,\sigma_d\left(c, \frac{|F_d - F_e|}{|F_e|+a_2} - b_2\right)$, and $\sigma_d(c,x) = \frac{1}{1+e^{-cx}}$. The terms $k_1, k_2, a_1, a_2, b_1, b_2$ and c are positive parameters. The minimum of $\hat{K}_{s,1}$ is kept to 0.0 [N/m]. The second part of the estimation law (20) is based on the fact that the stiffness increases with the applied force.

$$\hat{K}_{s,2} = K_{min} + k_3\,\sigma_d\left(c_0, |F_m| - F_0\right), \tag{22}$$

where F_0, c_0 and k_3 are positive parameters. K_{min} is set to 100.0 [N/m] in the experiments. Finally, low-pass filters are used to prevent jerking motions due to quick changes in the stiffness estimation. All the parameters are obtained experimentally: a_1, a_2, b_1, b_2, c, and c_0 are 1.0, 0.1, 1.5, 1.0, 5.0, and 0.2 respectively. F_0, k_1, k_2, and k_3 are 20.0 N, 10.0 m^{-1}, 10.0 m^{-1}, and 3000.0 N/m.

5 Experimental Setup

The slave station consists of a PUMA560 manipulator mounted on an XR4000 mobile base. The master station is a Phantom haptic device. Only position of the end-effector is controlled by teleoperation through force control. The end-effector rotation is simply controlled to maintain a fixed orientation. The XR4000 mobile base is controlled to track a trajectory autonomously in the null space. The time delay associated with the current wireless LAN setup is approximately 26 ms in each direction.



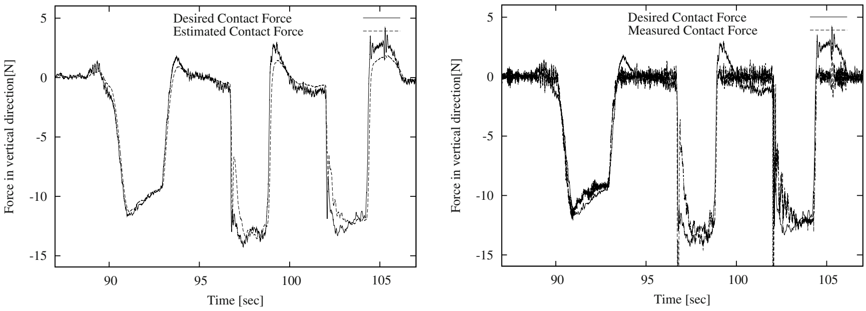
(a) (b)

Fig. 5. System Setup. (a) Phantom$^{\mathrm{TM}}$ device controlled by a human. (b) PUMA robot mounted on XR4000 (ROMEO)

6 Results

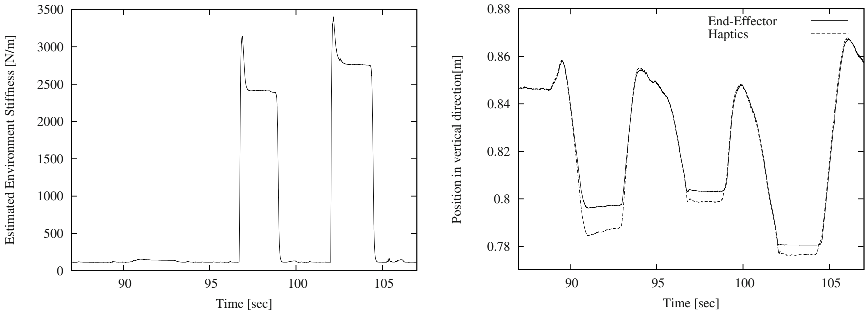
Two sets of experiments were conducted to show the decoupling of the end-effector control from the base control. Only the PUMA robot was controlled through teleoperation in the first set of experiments. The results are shown in Figure 6-8. The second set of experiments was conducted while the base was also controlled to move in the lateral direction (i.e. along the table in Figure 5) using the null space control. The desired trajectory was a sine function with an amplitude of 20 cm and a period of 12 seconds as shown in Figure 9. The results are shown in Figure 10-12. The fact that there is little difference in the performance of the two sets of experiments demonstrates the effectiveness of the decoupled control structure.

In both experiments, the operator began moving the slave manipulator in free space (i.e. no contact) by teleoperation. The different objects were then contacted sequentially by the end-effector of the slave manipulator: a sponge, a book, and a table. Their stiffnesses were identified off-line as 300, 3000, and 6000 [N/m] respectively, in order to be compared with the estimated values in Figure 7(a) and Figure 11(a). The robot was in free space in between the contacts (i.e. where $F_m \approx 0$).



(a) (b)

Fig. 6. Fixed base teleoperation: force response. The desired, estimated, and measured force at the end-effector of the slave manipulator are compared in (a) and (b). The desired force is generated by a virtual spring, $K_{vir}(x_m - x_s)$. The estimated force is from the AOB (a modified Kalman Estimator). The force is measured by the JR3 wrist force sensor.



(a) (b)

Fig. 7. Fixed base teleoperation: $\hat{K}_s$ and position tracking. (a) The estimated stiffness $\hat{K}_s$ for the environment that the end-effector of the slave manipulator is in contact with. (b) The end-effector of the slave manipulator tracking haptic position in the vertical direction.

Figure 6 and 10 show the contact force in the vertical direction at the end-effector of the slave manipulator. The manipulator was in free space in the region where the measured contact forces were near zero. These results demonstrate that the force controller with AOBs and stiffness adaptation performs well even in the presence of vastly different environmental changes. The estimated environment stiffness $\hat{K}_s$ was updated quickly and accurately enough to achieve the designed performance and stability criteria as shown in Figure 7(a) and 11(a). Position tracking performance is shown in Figure 7(b) and 11(b).

The desired contact force for the slave end-effector and the applied haptic force are compared in Figure 8(a) and 12(a). Without time-delay, they would be exactly the same. However, the effect of time-delay results in large differences especially

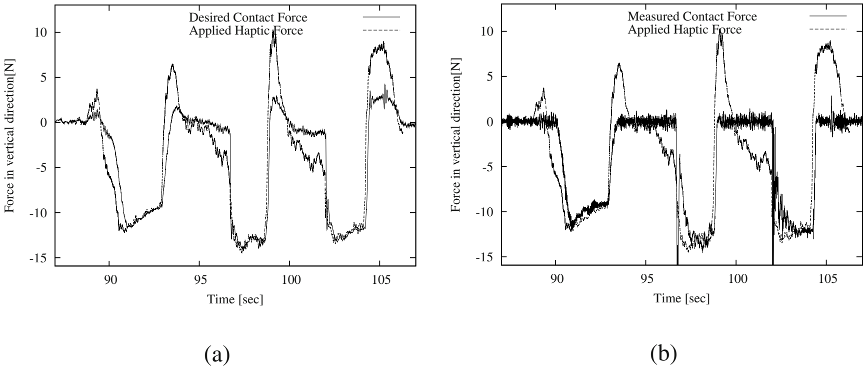


Fig. 8. Fixed base teleoperation: force comparison at the slave and master.(a) The haptic force and the desired contact force. The difference between them comes from the effect of time-delay. The haptic force is multiplied by the scaling factor 10.0 for comparison. (b) The haptic force and the measured contact force.

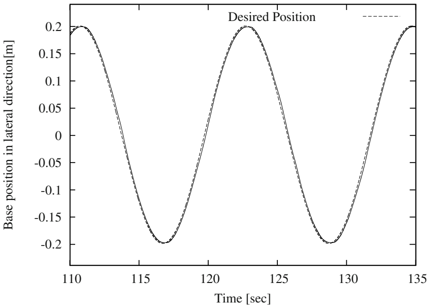
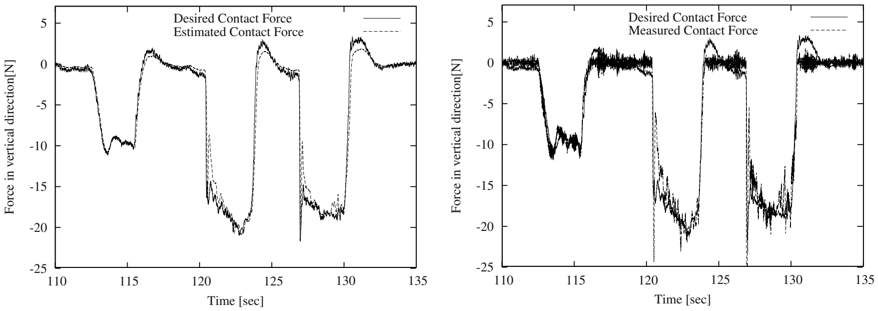


Fig. 9. Moving base teleoperation: base motion. The base moves in the lateral direction, (i.e. along the table in Figure 5) while the end-effector of the slave manipulator is controlled by teleoperation. The amplitude is 20 cm and the period is 12 seconds.

in free space motion. This contributes to the damping effect that the operator feels in free space operation. Figure 8(b) and 12(b) illustrates the measured contact force with the applied haptic force to the operator.

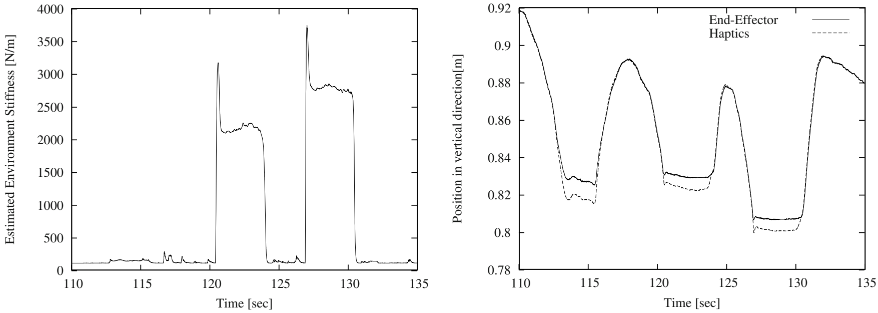
7 Conclusion

Robust local force control enables the overall teleoperation scheme to be simple and stable while providing realistic force feedback to a human operator. There is no switching in the control structure for different environments. The on-line stiffness adaptation performs effectively enough to match the changes in the environment. Thus, the entire tele-manipulation is always stable within the time delay margin. The teleoperation scheme developed for a single DOF system has been successfully



(a) (b)

Fig. 10. Moving base teleoperation: force response. The desired, estimated, and measured force at the end-effector of the slave manipulator are compared in (a) and (b). The desired force is generated by a virtual spring, $K_{vir}(x_m - x_s)$. The estimated force is from the AOB (a modified Kalman Estimator). The force is measured by the JR3 wrist force sensor.



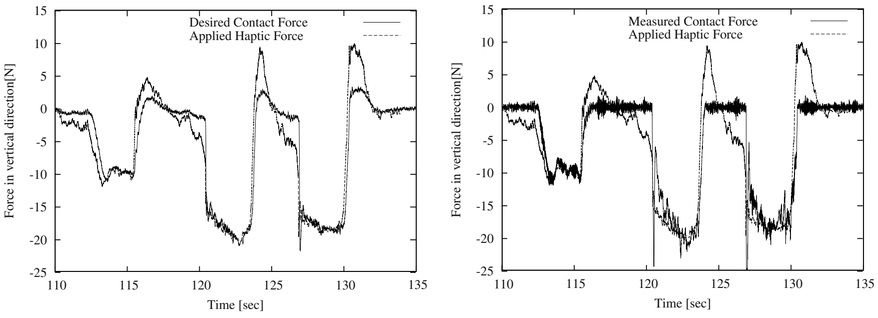
(a) (b)

Fig. 11. Moving base teleoperation: $\hat{K}_s$ and position tracking. (a) The estimated stiffness $\hat{K}_s$ for the environment that the end-effector of the slave manipulator is in contact with. (b) The end-effector of the slave manipulator tracking haptic position in the vertical direction.

applied to the mobile manipulator using the operational space framework. The user is able to control the end-effector and feel the environment realistically while the mobile base autonomously achieves the specified goals.

Acknowledgment

The authors would like to acknowledge Peter Thaulad for his contributions to the setup of hardware for the mobile base and the manipulator gripper.



(a) (b)

Fig. 12. Moving base teleoperation: force comparison at the slave and master. (a) The haptic force and the desired contact force. The difference between them comes from the effect of time-delay. The haptic force is multiplied by the scaling factor 10.0 for comparison. (b) The haptic force and the measured contact force.

References

1. R. Cortesão. *Kalman Techniques for Intelligent Control Systems: Theory and Robotic Experiments*. PhD thesis, University of Coimbra, 2002.
2. R. Cortesão, J. Park, and O. Khatib. Real-time adaptive control for haptic manipulation with active observers. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2938–2943, Las Vegas, 2003.
3. R. Daniel and P. McAree. Fundamental limits of performance for force reflecting teleoperation. *International Journal of Robotics Research*, 17(8):811–830, August 1998.
4. J.Park, R. Cortesão, and O. Khatib. Robust and adaptive teleoperation for compliant motion tasks. In *Proc. of the Int. Conf. on Advanced Robotics*, pages 513–519, Portugal, 2003.
5. O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *Int. J. on Robotics and Automation*, 3(1):43–53, February 1987.
6. W.S. Kim, B. Hannaford, and A.K. Bejczy. Force-reflecting and shared compliant control in operating telemanipulators with time delay. *Int. J. on Robotics and Automation*, 8:176–185, April 1992.
7. D. Lawrence. Stability and transparancy in bilateral teleoperation. *IEEE Trans. on Robotics and Automation*, 9(5):624–637, October 1993.
8. W. Zhu and S. Salcudean. Stability guaranteed teleoperation: An adaptive motion/force control approach. *IEEE Trans. on Automatic Control*, 45(11):1951–1969, November 2000.

XIV. Perception

Perception plays an important role in robotics. Various sensing modalities provide different levels of perception. In the first paper in this chapter, Rikoski *et al* investigates underwater tracking of a curved abject using sonar. The technique was able to tind regularities in the data and track various objects despite measurement and navigational uncertainties. The processed measurements are used as the input to an underwater SLAM algorithm.

A 3D model of the unknown environment is autonomously constructed by the identification of geometric parameters and contact configuration transitions during force-controlled task in the second paper by Slaets and co-workers. The algorithm also provides a force estimation to monitor contact forces or as a feedback to force controller. It also proposes a reduction in the modeling parameters to improve computational efficiency and more accurate geometric descriptions.

The Stiffness Imager by Kaneko, Kawahara and Tanaka in the third article is inspired by a medical application. The goal is to obtain an estimate ofthe stiffness of the area of interest during various surgical exercises, such as endoscopy. The article proposes a non-contact method of probing, using air-jet, onto the point of interest. The response of the environment of interest is recorded by a CCD camera, which is an existing part of the endoscopy procedure. The method was found to be effective and simple to implement that it was not deemed to introduce too much extra risk to the procedures.

The fourth article by Singh and Waldron deals with the unique dynamics oflegged locomotion. The problem lies in the estimation of the position and attitude of the robot. This paper proposes and shows that modified Extended Kalman Filter estimation techniques, combined with ground directed range sensors and characteristics unique to legged robots, such as the stride period and other periodic features, result in improved attitude tracking and reduced delay.

The fifth article by Kelly and co-workers presents the result of very rigorous evaluative tests on autonomous field mobile robots. The tests have helped to understand different natures of the problems and the shortcoming of the present technologies. In this paper, the result of the test by CMU PerceptOR team is presented and various observations were discussed. Immediate issues on perception and planning were highlighted and future work plans are suggested.

The last article, by Pradalier, Bessiere and Laugier, studied and tested an algorithm where a mobile robot re-traces its task by replaying sensori-motor trajectory - trajectory defined as a sequence of perceptions and actions, instead of the traditional way of following a predefined (geometric) trajectory - defined in position 1orientation with respect to time. The result was validated by simulation and real implementation on CyCab.

Trajectory Sonar Perception in the Ligurian Sea

Richard J. Rikoski[1], John J. Leonard[2], Paul M. Newman[3], and Henrik Schmidt[2]

[1] NSWC Panama City
Robotic Technologies Branch
richard.rikoski@navy.mil

[2] Massachusetts Institute of Technology
Department of Ocean Engineering
http://acoustics.mit.edu/GOATS

[3] University of Oxford
Robotics Research Group

Abstract. This paper describes a new technique for tracking locally curved unknown objects using sonar. The approach explicitly accounts for relevant robot dynamics. Objects are tracked by looking for temporal sequences of observations that fit a kinematic model. The method is illustrated using data from a synthetic aperture sonar gathered by an Odyssey III at NATO SACLANT's GOATS 2002 experiment in the Ligurian Sea.

1 Introduction and State-of-Art

To navigate relative to features, avoid obstacles, recognize objects, and interact with or manipulate the underwater environment, substantial improvements in sonar perception need to be made.

Numerous groups have investigated sonar perception. Leonard and Durrant-Whyte [7] developed regions of constant depth (RCDs) to determine correspondence in a sonar scan. Nagatani *et al.* [10] developed the Arc Transversal Median method to find edges. Mataric and Brooks [2] developed a robot that maintained a fixed distance from walls by tracking without an explicit model. Kleeman and Kuc [5] used the method of images to distinguish corners and walls. Barshan and Kuc [1] developed a sonar that differentiated between convex corners and walls based on amplitude measurements. Kuc [6] developed a sonar that could recognize objects based on waveforms; the sonar adaptively positioned itself to search for unique aspects of the target. Peremans *et al.* [13] built a tri-aural sonar for estimating the range and bearing to a target. Wijk and Christensen [18,19,17] developed Triangulation Based Fusion, a method for mapping vertical edges using a sliding window of recent sonar measurements. Kleeman [4] designed a sonar that used double pulse coding to reject interference and perform classification.

Rikoski and Leonard developed a method for tracking locally curved objects in a two-dimensional environment [15]. This paper will extend that result to three dimensions.

In Newman *et al.* [12], a constant time Simultaneous Localization and Mapping (SLAM) implementation was presented. The output of the sonar processing presented in this paper was used as an input to the mapping and navigation algorithm

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 557–570, 2006.

used in that paper. Newman's paper focused on SLAM; this paper will focus on sonar processing.

2 Trajectory Sonar Perception

Motivated by Marr's suggestion to "do what is possible, and proceed from there toward what is desirable" [9], the perception problem was broken down into a sequence of processing stages: signal processing (beamforming and matched filtering), detection, tracking, modeling, mapping, and object recognition. Each stage or competence had a well defined input, output, and transformation and was heavily grounded in physics [14]. The approach was designed to take advantage of delayed decision making [15,8]. Tracking, modeling, and object recognition were considered separate problems, as they happen on different time scales. This paper focuses on tracking.

We want to determine the minimum information necessary to determine correspondence between measurements of static features. We would like to do this without knowing what or where the feature is. This is appropriate because the modeling problem is greatly simplified if correspondence is known.

Trajectory Sonar Perception models geometric relationships between robots and features as potential fields. As a moving observer passes through a field, the observed rate of change of the field is known as the substantial derivative [11]. For a field Φ, the substantial derivative is defined as

$$\frac{D\Phi}{Dt} = \frac{\partial \Phi}{\partial t} + \dot{\mathbf{x}} \cdot \nabla \Phi. \tag{1}$$

Given a prior measurement of the field Φ_{i-1}, Φ_i can be predicted using a Taylor series expansion of substantial derivatives

$$\Phi_i = \Phi_{i-1} + \frac{D\Phi_{i-1}}{Dt}\Delta t + \frac{D^2\Phi_{i-1}}{Dt^2}\frac{\Delta t^2}{2!} + \text{h.o.t.} \tag{2}$$

Assume a sensor that measures a geometric relationship between a robot and a target. Assuming that the geometric measurements vary continuously, and assuming the robot motion to be continuous, it is reasonable to apply the substantial derivative. A typical marine robot displaces 50-500kg, transmits its sonar 3-10 times per second, and is perturbed by waves with periods on the order of 5-30 seconds, making these assumptions reasonable. The robot's dynamic state changes very little on the timescale of the measurements.

Assume an initial range, azimuth, and elevation measurement of a target $\begin{bmatrix} r\ \theta\ \phi \end{bmatrix}^T$. A subsequent measurement can be predicted through a Taylor series expansion of substantial derivatives.

$$\begin{bmatrix} r_1 \\ \theta_1 \\ \phi_1 \end{bmatrix} = \begin{bmatrix} r_0 + \frac{Dr}{Dt}\Delta t + \frac{D^2 r}{Dt^2}\frac{\Delta t^2}{2!} + \text{h.o.t.} \\ \theta_0 + \frac{D\theta}{Dt}\Delta t + \text{h.o.t.} \\ \phi_0 + \frac{D\phi}{Dt}\Delta t + \text{h.o.t.} \end{bmatrix} \tag{3}$$

The Taylor series for range includes a second order term. As will be shown later, the second substantial derivative of range is equivalent to the first substantial derivatives of azimuth and elevation. The second substantial derivative of range is

$$\frac{D^2r}{Dt^2} = \frac{\partial^2 r}{\partial t^2} + 2\dot{\mathbf{x}} \cdot \nabla \frac{\partial r}{\partial t} + \ddot{\mathbf{x}} \cdot \nabla r + \dot{\mathbf{x}}^T (\nabla \nabla^T r)\dot{\mathbf{x}}. \tag{4}$$

In the typical definition of the substantial derivative, $\dot{x}$ is the vector of vehicle velocities $\begin{bmatrix} u \; v \; w \end{bmatrix}^T$. For the purposes of this paper, $\dot{x}$ will be extended to include the roll, pitch, and yaw rates of the robot $\begin{bmatrix} P \; Q \; R \end{bmatrix}^T$. This is because the relative target bearing changes as the robot turns. Expanded, the substantial derivative of range becomes

$$\frac{Dr}{Dt} = \frac{\partial r}{\partial t} + \frac{\partial r}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial r}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial r}{\partial z}\frac{\partial z}{\partial t} + \frac{\partial r}{\partial \xi_x}\frac{\partial \xi_x}{\partial t} + \frac{\partial r}{\partial \xi_y}\frac{\partial \xi_y}{\partial t} + \frac{\partial r}{\partial \xi_z}\frac{\partial \xi_z}{\partial t} \tag{5}$$

$$= \frac{\partial r}{\partial t} + \frac{\partial r}{\partial x}u + \frac{\partial r}{\partial y}v + \frac{\partial r}{\partial z}w + \frac{\partial r}{\partial \xi_x}P + \frac{\partial r}{\partial \xi_y}Q + \frac{\partial r}{\partial \xi_z}R$$

where ξ_x, ξ_y, and ξ_z denote rotation around the x, y, and z axes of the robot respectively. The partial derivatives $\frac{\partial \xi_x}{\partial t}$, $\frac{\partial \xi_y}{\partial t}$, and $\frac{\partial \xi_z}{\partial t}$ are the same as the roll, pitch and yaw (P, Q, R).

By assuming the medium to be coherent [16], $\frac{\partial r}{\partial t}$, $\frac{\partial \theta}{\partial t}$,and $\frac{\partial \phi}{\partial t}$ can be neglected. This assumption implies that when the robot is stationary the measurements are stationary.

3 Universal Target Model

In order to calculate the substantial derivative of measurements, one must have a geometric model. In most prior work, point-like and planar objects have been treated as distinct. While maintaining discrete feature types is sound when there are relatively few, this approach will likely break down as more feature types are added and the differences between them is reduced. It would be desirable to have a single generic feature model that applies to all possible feature types.

By assuming targets to be locally curved, it is possible to accommodate both point-like and planar surfaces. If a locally curved surface has a center of curvature (x_c, y_c, z_c) and a radius of curvature ρ, then a point-like object has $\rho = 0$ and a planar surface has $\rho = \infty$. All other objects have finite non-zero radii of curvature. A convex object has $\rho > 0$, a concave object has $\rho < 0$. Putting the feature into robot coordinates (the robot is at $(0, 0, 0)$ facing in the x direction), but leaving the robot coordinates (x, y, z) in the equations to illustrate the derivatives, the predicted measurement is

$$\begin{bmatrix} r \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(x_c - x)^2 + (y_c - y)^2 + (z_c - z)^2} - \rho \\ \arctan(\frac{y_c - y}{x_c - x}) \\ \arccos(\frac{x_c - x}{\sqrt{(x_c-x)^2+(y_c-y)^2+(z_c-z)^2}}) \end{bmatrix}. \tag{6}$$

Unfortunately, a measurement $\left[r\ \theta\ \phi\right]^T$ cannot be directly inverted to yield the target model $\left[x_c\ y_c\ z_c\ \rho\right]^T$. The model has four degrees of freedom, but the measurement has only three degrees of freedom. However, it is possible to calculate the substantial derivatives.

Consider the first term in the substantial derivative of range, $\frac{\partial r}{\partial x}u$. The partial derivative $\frac{\partial r}{\partial x}$ is

$$\frac{\partial r}{\partial x} = -\frac{x_c - x}{r + \rho}, \tag{7}$$

where $r + \rho$ is the distance to the center of curvature.

Although the robot does not know the center of curvature (x_c, y_c, z_c) or the radius of curvature ρ, it can measure θ and ϕ. Substituting, it is evident that

$$\frac{\partial r}{\partial x} = -\cos(\theta)\sin(\phi). \tag{8}$$

By designing the sensor to measure θ and ϕ, it is possible to essentially measure the substantial derivative of range. The first substantial derivative of range of a locally curved object calculated using angular measurements is

$$\frac{Dr}{Dt} = -u\cos(\theta)\sin(\phi) - v\sin(\theta)\sin(\phi) - w\cos(\phi). \tag{9}$$

Notice that the first substantial derivative of range depends only on measurements and robot velocities (which can also be measured); it does not require globally referenced information or angular rates. Given velocity information, it can be calculated entirely from a previous measurement.

The derivatives of θ and ϕ, and the second substantial derivative of r are more difficult. The distance to the center curvature $r + \rho$ is squared in the denominator and cannot be removed entirely by substituting trigonometric identities. For instance:

$$\frac{D\theta}{Dt} = \frac{u\sin(\theta) - v\cos(\theta)}{\sin(\phi)(r + \rho)} - \frac{P\cos(\theta)\cos(\phi) + Q\sin(\theta)\cos(\phi)}{\sin(\phi)} - R. \tag{10}$$

The component with $r + \rho$ in the denominator is the angular rate of the target due to robot translation. At broadside ($\theta = \frac{\pi}{2}$, $\phi = \frac{\pi}{2}$) the angular rate of the target is $\frac{u}{r+\rho}$.Unfortunately, the radius of curvature cannot be observed from a single measurement so this term cannot be directly predicted.

The second substantial derivative of range suffers from a similar problem. When a robot passes a target at broadside, the target appears to accelerate away from the robot at $\frac{u^2}{r+\rho}$. While this is very similar to centripetal acceleration, since the target is "center avoiding" this is, strictly speaking, a centrifugal effect.

What becomes evident is that by substituting the first substantial derivatives of the angles into the second substantial derivative of range, unwanted terms can be forced to drop out. In addition to removing the undesired $r + \rho$, the angular rates can be forced to drop out. Looking at $\frac{D\theta}{Dt}$, it is evident that if the robot has a yaw rate R, the angular rate of the target will be $-R$. If the robot has velocity u and is passing the target at broadside on the right as it yaws, it will accelerate away from the target at uR. If the robot has a turning radius ρ_r, so that its yaw rate $R = \frac{u}{\rho_r}$, the robot accelerates away from the target at $\frac{u^2}{\rho_r}$, which is the robot's centripetal acceleration. Finishing the substitution yields the second substantial derivative in range

$$\frac{D^2r}{Dt^2} = -\dot{u}\cos(\theta)\sin(\phi) - \dot{v}\sin(\theta)\sin(\phi) - \dot{w}\cos(\phi) \tag{11}$$
$$+(u\sin(\theta)\sin(\phi) - v\cos(\theta)\sin(\phi))\frac{D\theta}{Dt}$$
$$-(u\cos(\theta)\cos(\phi) + v\sin(\theta)\cos(\phi) - w\cos(\phi))\frac{D\phi}{Dt}.$$

It is not possible to predict the second substantial derivative in range from one prior measurement. Given two prior measurements, however, it is possible to measure the angular substantial derivatives and then predict a third measurement.

To track, measurements are predicted using the universal feature model. A Mahalanobis test is used to associate new measurements with old measurements. If a measurement trajectory reaches a minimum length, it is kept. For this paper, measurement trajectories were not used if they contained fewer than five measurements.

4 Non-dimensional Analysis

It is necessary to determine when terms are observable for two reasons. First, it is important to determine how many terms should be kept in the Taylor expansion. Second, terms such as the angular rates must be observable if they are used to calculate other quantities such as the second substantial derivative of range.

The first question is whether the range changes enough ping to ping to justify using the first substantial derivative. It will be assumed that the sonar that has a ping rate f_s, a wavelength λ, and quarter wavelength resolution. In order for the first substantial derivative of range to be observable, the range must change more than a quarter wavelength between pings:

$$\frac{Dr}{Dt}\Delta t > \frac{\lambda}{4} \tag{12}$$

Since the substantial derivative of range is maximized when the target is in the robot's direction of travel, using V for the robot's speed, and substituting $\frac{1}{f_s}$ for Δt, this can be reduced to

$$\frac{V}{f_s} > \frac{\lambda}{4}. \tag{13}$$

This leads to the non-dimensional number

$$N_1 = \frac{V}{\lambda f_s}. \tag{14}$$

If N_1 exceeds $\frac{1}{4}$, then the range rate is observable. If the resolution were some other fraction of a wavelength, then N_1 would have to exceed that fraction. N_1 is also used to determine whether the heave and sway velocities need to be modeled. If, using the heave or sway velocity, the calculated N_1 value is less than $\frac{1}{4}$, those terms can be dropped from first substantial derivative of range. This is important if velocity information is not available, as it reduces the minimum number of features necessary for a velocity solution.

The second substantial derivative of range is observable if

$$\frac{\varDelta t^2}{2!}\frac{D^2 r}{Dt^2} > \frac{\lambda}{4}. \tag{15}$$

It can be thought of as containing three components. The first component, described by N_2, analyzes the tendency of the range rate to change as a robot drives past a target(the previously mentioned centrifugal effect). The second component considers the contribution of the robot's angular rate to the second substantial derivative of range, and is described by N_3. The third component, the contribution of acceleration, has the non-dimensional number N_4. The non-dimensional numbers are

$$N_2 = \frac{V^2}{2f_s^2 \lambda r_{min}} \tag{16}$$

$$N_3 = \frac{V\varUpsilon}{2f_s^2 \lambda} \tag{17}$$

$$N_4 = \frac{\dot{V}}{2f_s^2 \lambda}. \tag{18}$$

In N_3, $\varUpsilon$ represents the maximum angular rate of the robot. If either N_2, N_3, or N_4 exceed the fraction describing the wavelength resolution of the sonar, the second substantial derivative of range is observable.

The angular rate of the robot is due to two effects. The target bearing can change because the robot passes the target, or because the robot turns. N_5 describes the contribution of robot translation, N_6 describes the contribution of turning.

$$N_5 = \frac{VL}{r_{min}\lambda f_s} \tag{19}$$

$$N_6 = \frac{\varUpsilon L}{\lambda f_s} \tag{20}$$

In both N_5 and N_6, the aperture of the sonar L appears. It is assumed that the angular rate of the target is observable only if the change in the target bearing exceeds the resolution of the sonar. The broadside resolution of a line array, $\frac{\lambda}{4L}$ is used for simplicity. If a robot has a sonar with an aperture L, the angular rate is observable if N_5 or N_6 exceed $\frac{1}{4}$.

The first six numbers describe robot effects in calm water. However, the ocean does not stand still. The effect of a current can be modeled through a straightforward pplication of N_1 , but the effects of waves are slightly more complicated. Only linear waves will be considered in this paper. To avoid confusion with acoustic parameters, the parameters describing surface waves will use uppercase variables (because surface waves are "bigger" than acoustic waves). The relevant parameters are the angular frequency Ω, amplitude A, wave number K, and wavelength Λ.

In deep water, it is obvious that particles on the water surface move up and down as waves pass. Particles also have horizontal motions that are equal in in amplitude to the vertical motions, but $90°$ out of phase. Consequently, the particle motions are circular, and the paths are called particle orbits. The diameter of these orbits decay exponentially with depth. If a robot is neutrally buoyant and small with respect to the wavelength of the wave, it is reasonable to expect it to orbit like a particle. Inserting the particle velocity derived in Newman [11] into N_1 yields a non-dimensional number describing the effect of linear deep water waves:

$$N_7 = \frac{A\Omega e^{Kz}}{f_s \lambda}. \tag{21}$$

The z variable is referenced to the surface and underwater values are negative. At a depth of half a wavelength Λ, the amplitude of a particle orbit is reduced to $Ae^{-\pi}$. However, a moderate ocean wave can have a wavelength in excess of $100m$, so wave effects can extend considerably below the surface.

The shallow water case is more complicated. The vertical motion of water particles is restricted, as they cannot pass into the sea floor. Shallow water wave particles move in ellipses; at the bottom the particles only move horizontally. The non-dimensional number that describes the contribution of linear shallow water waves is

$$N_8 = \frac{gAK}{\Omega} \frac{\cosh(K(z+h))}{f_s \lambda \cosh(Kh)}. \tag{22}$$

For this number, h is the water depth and $g = 9.8\frac{m}{s^2}$.

5 Experimental Setup

In 2002, in conjunction with the NATO SACLANT research center in La Spezia, Italy, a set of missions were conducted as part of the GOATS 2002 experiment. The experiment was conducted using Caribou, an Odyssey III vehicle built by Bluefin Robotics. The Odyssey III is a streamlined vehicle with a vectored thruster. The vehicle featured a 16-element dual line array synthetic aperture sonar built by MIT.Since the drag on the cylindrical acoustic transmitter likely exceeded the vehicle drag by an order of magnitude, it was hidden inside the vehicle hull, which was designed to be acoustically transparent. The transmitter was aimed at broadside, and transmitted a $5-25$kHz linear chirp. The receiver, mounted on the nose, consisted of two parallel arrays of eight elements. The elements were spaced 10cm apart and the arrays were spaced 15cm apart.

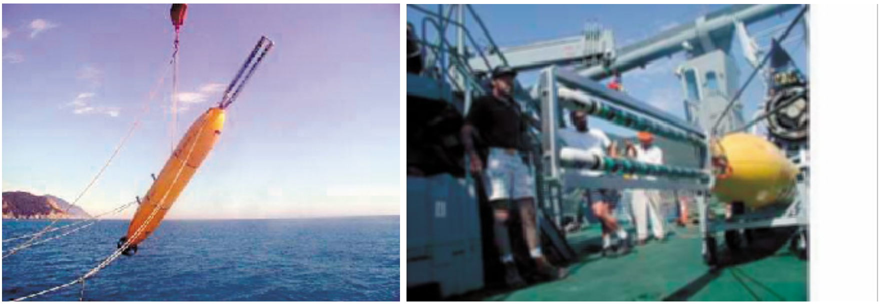


Fig. 1. On the left, the robot, *Caribou*, being lowered into the water off the coast of Elba. On the right is a close up of the 16-element dual line array used to receive the sonar signals. The green spheres are individual hydrophones.

A characteristic along-track robot velocity was $1.5\frac{m}{s}$, a representative wavelength was 0.1m, and the ping rate $f_s = 3$Hz. Since $N_1 = 5$, the range rate was observable. Although direct sea state measurements were unavailable, wave induced heave and sway velocities between $0.1\frac{m}{s}$ and $0.25\frac{m}{s}$ were routinely observed. Since any velocity in excess of $0.075\frac{m}{s}$ would cause noticeable contribute to the range rate, the contributions of the induced heave and sway velocities were observable.

The robot typically operated 10m off the bottom. At that minimum range, $N_2 = 0.125$. The maximum yaw rate and acceleration were approximately $\Upsilon = \frac{10^\circ}{s} = 0.1745\frac{rad}{s}$ and $0.1\frac{m}{s^2}$ respectively. The other two measures of the second substantial derivative of range were $N_3 = 0.22$ and $N_4 = 0.06$. Although N_3 put the robot close to the transition, the second substantial derivative was neglected. Most targets were at least 20m away, and targets directly under the robot were usually masked by the powerful bottom reflection.

Since $N_5 = 0.35$ and $N_6 = 0.41$, the angular rates were barely observable. The beamformer used to produce the results in [12] was noisy enough to justify ignoring the angular rate. The beamformer used in this paper allowed the angular rate to occasionally be observable. Nevertheless, the information was not used. Consequently, the measurement prediction used in this paper was

$$\begin{bmatrix} r_1 \\ \theta_1 \\ \phi_1 \end{bmatrix} = \begin{bmatrix} r_0 - (u\cos(\theta)\sin(\phi) + v\sin(\theta)\sin(\phi) + w\cos(\phi))\varDelta t \\ \theta_0 \\ \phi_0 \end{bmatrix} \tag{23}$$

The results presented here are from the *Caribou* mission initiated at 4:11pm on June 15, 2002, off the coast of Framura, Italy. In the GOATS data sets, this mission is referred to as SAS_15_6_2002____16_11. This is the same data set used in [12]. The mission lasted for roughly half an hour, and was conducted in approximately 20m of water about 500m from shore. The robot started on the surface, dove to roughly 10m, and then circled the targets. The vehicle path is shown in Fig 2.

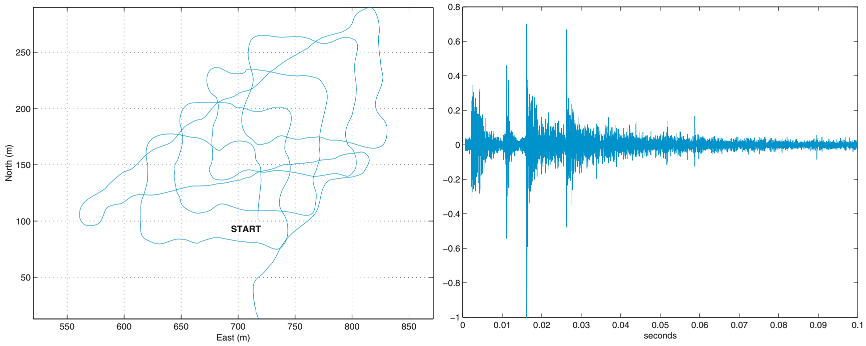


Fig. 2. The robot path is shown on the left. Since the sonar transmitted out the left side of the vehicle, the vehicle constantly turned left to keep the target field insonified. A typical matched filtered waveform is shown on the right. The pulse at the front of the waveform is a remnant of the outgoing signal. The next three pulses correspond to the reflection off the surface, the reflection off the bottom, and the multipath reflection off the surface and then the bottom. Potential targets are evident in the second half of the waveform.

6 Experimental Results

Starting with raw sonar signals, the robot systematically refined the data, building towards higher level constructs. The data was matched filtered, beamformed, and passed to a detector. The output of the detector was passed to feature tracker, which determined correspondence between pings of unknown objects using vehicle dynamics. The time series of raw detections and tracked features are shown in Fig 3 and Fig 4. Detections are shown in blue and measurements from tracked targets are shown in red.

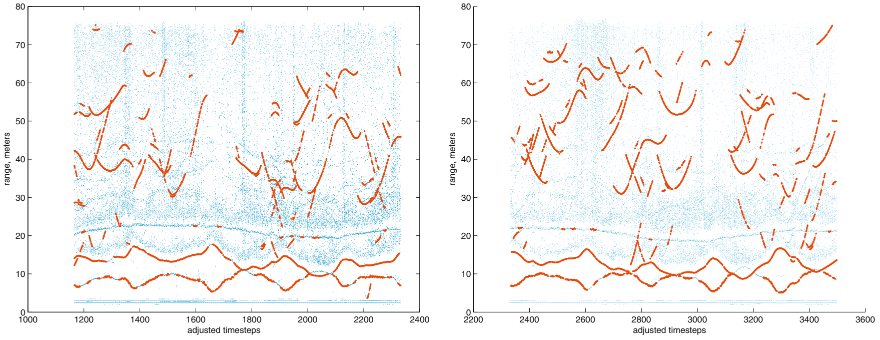


Fig. 3. This figure shows measurements plotted in measurement space. The x-axis is adjusted timestep, the y-axis is range in meters. Detections are plotted in blue, measurement trajectories are shown in red. The measurements on the right are for pings 1000 to 2000 (adjusted timesteps 1167 to 2334), the measurements on the right are for pings 2000 to 3000.

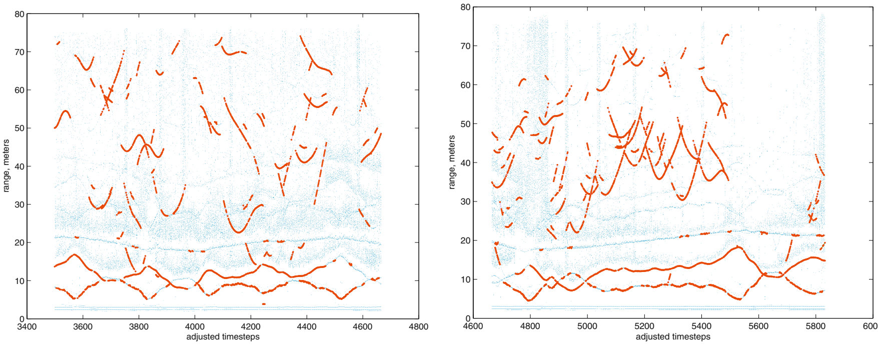


Fig. 4. On the left, the measurements and tracked features for pings 3000 to 4000. Pings 4000 to 5000 are shown on the right.

In the time series of measurements, the detections at 3m are artifacts of the outgoing chirp. The two curves that are centered at approximately 10m are the reflections off the ocean surface and bottom. The two curves are mirror images of one another because, for a flat bottom, if the robot depth increases the robot altitude decreases. The measurements of the surface reflections appear to be noisier than measurements of the bottom because the ocean surface is not static.

The slowly varying line at 20m corresponds to the water depth. The signal reflects off the surface, off the bottom, and then back to the robot (or off the bottom, then off the surface, then back to the robot, there are two paths). Centered around 30m are multi-path observations of the water surface and bottom. The surface multi-path reflection would bounce off the surface, off the bottom, off the surface, and then back to the robot. The reflections at approximately 40m are at twice the water depth, and correspond to two reflections off the surface and bottom. Each reflection off the surface introduces unmodeled noise into the sequence of measurements, since the water surface changes shape from ping to ping. Consequently, multi-path reflections are not tracked well.

The short vertical streaks are due to signals from the acoustic transponders used for navigation. The long wide streaks correspond to acoustic modem transmissions. Both signals tended to cause numerous false alarms at the level of the detector.

The detections and measurements from tracked features from pings 3000 to 4000 are projected into global coordinates in Fig 5. Since the altitude and depth reflections are from directly below and above the robot, they correspond to the vehicle's path. Notice that, in general, this is an awkward representation for feature extraction. Also in Fig 5, there is a sequence of measurements corresponding to a mid-water target. This target was not tracked. Because the target was close, the second substantial derivative of range was observable. Additionally, the target was inside the nearfield of the array. Bearing is a well defined construct for targets in the farfield of an array. However, when the farfield definition is applied to targets in the nearfield, the results range from imprecise to nonsensical.

Tracked targets start to appear after the bottom bounce, and extend out to 75m. Typical sequences of target measurements and predictions are shown in Figs 6 and 8. The robot is able to predict measurements to a fraction of wavelength. Differences between the predicted and actual measurements are primarily due to navigational errors. This is especially evident when comparing the predicted and actual range rates (Figs 7 and 9). The predicted range rate tends to lag the actual range rate. This is most likely because the state estimate from the navigation filter lags the true robot state.

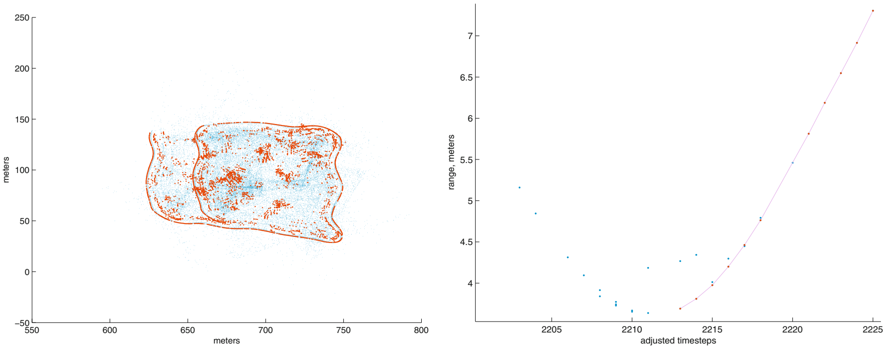


Fig. 5. On the right, the detections and tracked features from pings 3000 to 4000 are projected into cartesian coordinates. Clearly, this representation has less structure, making feature extraction more difficult. On the right is a feature that the algorithm failed to track. The curve on the right corresponds to a midwater object. Although the measurement are continuous, the measurement sequence does not fit two of the assumptions. First, the second substantial derivative of range is non-negligible because the target is close. Since the robot skipped a transmission at the closest point, the second order effect is four times greater than normal. Second, because the target is so close it's inside the nearfield of the array. This leads to more detections and reduced accuracy.

7 Conclusion

Using the constraint based on a universal feature model, a variety of features (point objects, concrete blocks, simple geometric shapes, the surface, the ocean floor) were tracked in a dynamic shallow water environment. The technique was able to find regularities in the data despite measurement and navigational uncertainty, and numerous interfering sources such as acoustic modems and long baseline navigation beacons. The processed measurements were used by Newman as the input to a constant time SLAM algorithm [12].

There are four obvious research paths leading from this work. First, the work should be extended to dynamic objects. Second, techniques for creating geometric models of features given correspondence need to be investigated. Third, the approach

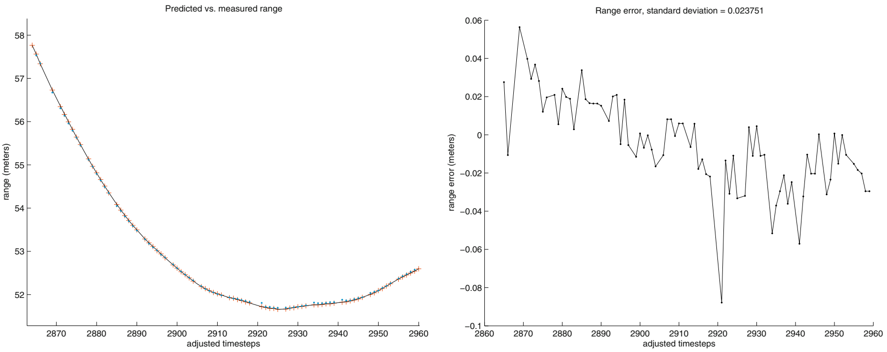


Fig. 6. On the right, a time series of measurements from a single target. The x-axis is adjusted timestep, the y axis is range. The measurements are plotted as red points connected by the black line, the predicted measurements are plotted in blue. The graph on the right shows the difference between the predicted and actual measurements. The errors are on the order of centimeters despite the robot moving at roughly $1.5\frac{m}{s}$ and the target being over 50m away.

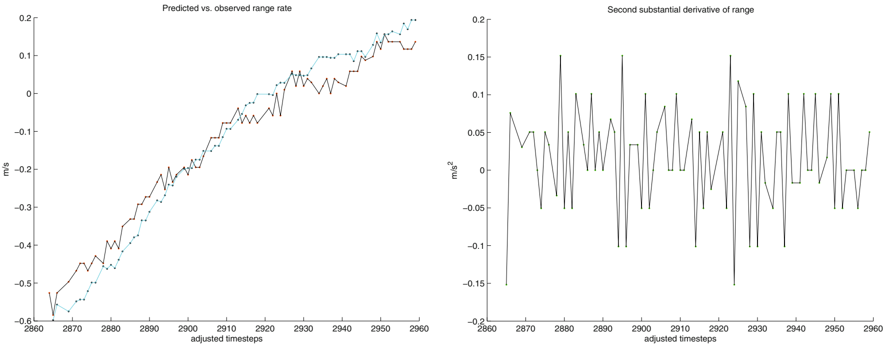


Fig. 7. The predicted and measured substantial derivatives of range are shown on the left. The range rate calculated from measurements is shown as red points connected by a black line, the range rate predicted using vehicle dynamics is shown as black points connected by a blue line. The second substantial derivative of range as calculated from sets of three measurements is shown on the right. Using non-dimensional analysis it was concluded that this derivative would not noticeably affect the time sequence of measurements from ping to ping. The data agrees with the non-dimensional analysis; the second substantial derivative is essentially zero mean noise.

needs to be extended beyond the Freedman model [3] to include rough surface scattering. Finally, this analysis needs to be used to create a holistic design guide for marine robots using sonar.

Acknowledgement

This work was supported by the Office of Naval Research under grant N00014-97-1-0202.

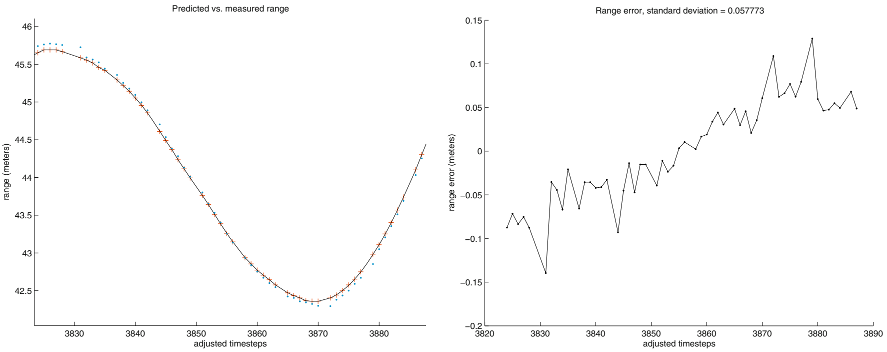


Fig. 8. A second time series of measurements from a target, and the difference between the predicted and observed range. In this case, the robot is turning while tracking the feature. This sequence of measurements has greater range errors, due to velocity errors as the robot turns.

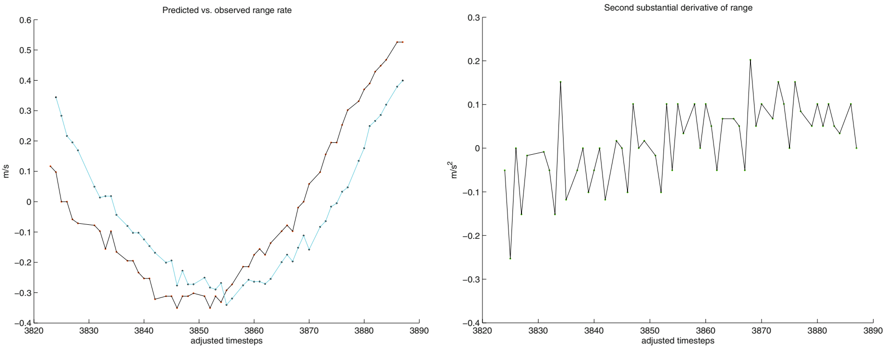


Fig. 9. Looking at the range rates, it is apparent why there were range errors in the previous figure. The navigation filter tends to lag the true state, that lag causes errors in the predicted range rate, which causes errors in the range prediction.

We would like to thank the entire GOATS team. On the sonar side, Don Eickstedt, T. C. Liu, Wen Xu, and Joe Edwards designed and built the sensor used in this experiment, without them this work could never have happened. Rob Damus, Sam Desset and Jim Morash had the tough but vital job of fielding and maintaining the robots and their software. Matt Grund integrated the WHOI modems into the MIT vehicles, without which our capabilities would have been significantly reduced. The crew of the R.V. Alliance did a tremendous job of making the entire experiment come off without a hitch, and keeping the scientific staff from hurting themselves.

References

1. B. Barshan and R. Kuc. Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(6):560–569, June 1990.

2. R. A. Brooks. *Cambrian Intelligence: The Early History of the New AI*. The MIT Press, 1999.
3. A. Freedman. A mechanism of acoustic echo formation. *Acustica*, 12(1):10–21, 1962.
4. L. Kleeman. Advanced sonar with velocity compensation. *Int. J. of Robot. Res.*, 23(2):111–126, February 2004.
5. L. Kleeman and R. Kuc. Mobile robot sonar for target localization and classification. Technical Report ISL-9301, Intelligent Sensors Laboratory, Yale University, 1993.
6. R. Kuc. Fusing binaural sonar information for object recognition. In *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 727–735, 1996.
7. J. J. Leonard. *Directed Sonar Sensing for Mobile Robot Navigation*. PhD thesis, University of Oxford, 1990.
8. J. J. Leonard, R. J. Rikoski, P. M. Newman, and M. Bosse. Mapping partially observable features from multiple uncertain vantage points. *Int. J. of Robot. Res.*, 21(10/11):943–976, October 2002.
9. D. Marr. *Vision*. New York: W. H. Freeman and Co., 1982.
10. K. Nagatani, H. Choset, and N. Lazar. The arc-transversal median algorithm: an approach to increasing ultrasonic sensor accuracy. In *IEEE ICRA*, pages 644–651, 1999.
11. J. N. Newman. *Marine Hydrodynamics*. The MIT Press, 1977.
12. P. M. Newman, J. J. Leonard, and R. J. Rikoski. Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. In *ISRR*, 2003.
13. H. Peremans, K. Audenaert, and C. J. M. Van. A high-resolution sensor based on triaural perception. *IEEE Transactions on Robotics And Automation*, 9(1):36–48, Feb. 1993. USA.
14. W. Richards, editor. *Natural Computation*. The MIT Press, 1988.
15. R. J. Rikoski. *Dynamic Sonar Perception*. PhD thesis, MIT, 2003.
16. S. J. Stanic, Ralph R. Goodman, Roger W. Meredith, Edgar Kennedy. Measurements of high-frequency shallow-water acoustic phase fluctuations. *IEEE J. of Oceanic Eng.*, 25(4):507–515, October 2000.
17. O. Wijk. *Triangulation Based Fusion of Sonar Data with Application in Mobile Robot Mapping and Localization*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2001.
18. O. Wijk, H. Christensen. Triangulation based fusion of sonar data with application in robot pose tracking. *IEEE Trans. Robotics and Automation*, 16(6):740–752, December 2000.
19. O. Wijk, H. I. Christensen. Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data. *Robotics and Autonomous Systems*, 31:31–42, 2000.

Construction of a geometric 3-D model from sensor measurements collected during compliant motion

Peter Slaets[1], Johan Rutgeerts, Klaas Gadeyne, Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter

Department of Mechanical Engineering , Katholieke Universiteit Leuven
Celestijnenlaan 300B, B3001 Leuven(Heverlee)Belgium.
Email: Peter.Slaets@mech.kuleuven.ac.be

Abstract. This paper describes the construction of a 3-D model from the identification of geometrical parameters of polyhedral objects during a force-controlled task. Following improvements with respect to the state of the art are made: (i) creation of a 3-D model from a previously unknown environment, (ii) force estimation for feedback to a force controller or for monitoring contact forces, (iii) reduction of the number of modelling parameters, leading to computational reduction and more accurate geometric descriptions.

1 Introduction

This paper describes the construction of a geometric 3-D model derived from position and force sensor measurements during an action executed in an unknown environment. Subsequent Bayesian processing leads to the gradual building of a geometric 3-D model of the environment, which is reduced to the smallest possible set of polyhedral geometric primitives. The method can be applied to other sensors than force and position, e.g. for all kinds of *programming by demonstration (PbD).*

Figure 1 describes the task used to demonstrate our method: the robot grasps a cube and a human steers the robot with a joystick through a sequence of contact formations (CFs). The geometry of the object and the environment are completely unknown, and are constructed as follows: (1) *modeling/estimation* of the geometric contact parameters with the *non-minimal state Kalman filter* of [1]; (2) transitions between contact formations (CFs) are detected as soon as measurements become inconsistent with the current measurement equations; (3) *model reduction* is done with the *Bayes factor* method [2], i.e.the ratio of the marginal likelihoods.

This paper's major contributions are: (1) while previous work [3] focused on estimation within a *given* polyhedral geometry, this work is able *to create* a geometric 3-D model; (2) the estimation also provides a *force decomposition* in the individual contacts; (3) *data reduction* is performed from a non-minimal set of geometric primitives (edge-face contacts, basically) to ge-

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 571–580, 2006.

ometric models with less, but more relevant, geometric parameters (multiple face-face contacts, for example).

2 Modeling

We consider polyhedral objects, so only three elementary contacts exist: vertex-face, face-vertex, edge-edge. All other contacts are a combination of these elementary contacts, e.g., a face-face contact can be modeled as a combination of three vertex-face contacts. (This section describes the vertex-face contact; other contacts are tackled analogously.) The geometrical parameters are: (i) the position of the *vertex*, $p = [p_x, p_y, p_z]^T$, and (ii) the parameters $[a, b, c, d]$ of the *face*. These parameters are collected in the Kalman Filter's state vector x_g. The measurements (z) are: (i) contact wrench $w = [f^T, m^T]^T$, measured by a wrist force sensor; (ii) end-effector pose t_d (position/orientation). The *pose measurement equation*, $h(x_g, t_d) = 0$, relates t_d to the state x_g, for every possible contact. For a vertex-face contact, this expresses that the vertex lies in the face: $ap_x + bp_y + cp_z - d = 0$. The nonlinearity of the Kalman Filter equations come from the transformations of the measurements and contact parameters to the same Cartesian frame. A similar nonlinear transformation applies to the *wrench measurement equation*, that transforms the wrench measured at the wrist to the (unknown) contact positions. In addition, the polyhedral contacts are *unilateral*, which introduces inequality constraints on the *wrench coordinate* ϕ, i.e., the (to be estimated) magnitude of the contact force. This paper extends the Kalman Filter state with this ϕ, in order to achieve *force decomposition*, i.e., the estimation of the contact forces at each of the k individual CFs. The selection of the stochastically optimally matching combination of k pose and wrench measurement equations is done by using SNIS and Bayesian hypothesis tests.

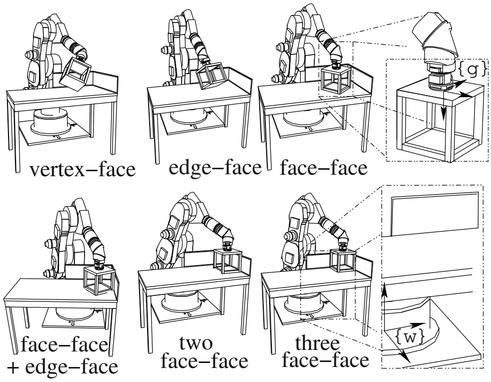


Fig. 1. A *cube-in-corner* assembly.

So, the geometrical parameters x_g and the force magnitudes ϕ are combined into the state vector ($x = [x_g^T \;\; \phi^T]^T$), and the pose ($t_d$) and wrench ($w$) measurements in the measurements vector ($z = [t_d^T \;\; w^T]^T$).

We use automatic generation of the measurement equations, which is simplest if using more than the minimal number of elementary CFs is allowed, e.g., it should be possible to represent an edge-face contact with two elementary vertex-face contacts, without bothering about whether this is the minimal number or not. Our non-minimal state Kalmen Filter estimator recognizes measurement equations being non-minimal, and assures that the information is not processed more than once.

The Process model updates the state estimates between measurements in order to follow the dynamics of the system. The part of the state corresponding to the geometrical parameters is static, i.e., the real value of the parameters do not change over time. These parameters can however become very accurate (small covariances) after a number of measurement updates resulting in numerical instability. A limitation of accuracy of the covariance matrix of the state is implemented to prevent this. The part of the state corresponding to the wrench coordinates can not be predicted without having a model of the contact dynamics (which is the case). The process model for this part of the state makes its estimate uncertain again (the next estimate is only dependent on the current measurement). This leads to:

$$x_{k+1|k} = Ax_{k|k} + w, \quad P_{k+1|k} = AP_{k|k}A^T + Q, \quad \text{with} \tag{1}$$

$$x = [x_g^T \;\; \phi^T]^T, \quad w = [0^T \;\; \phi_{0|0}]^T, \quad A = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} I & 0 \\ 0 & P_{\phi,0|0} \end{bmatrix}.$$

The Non-minimal state Kalman filter transforms the state x into a higher-dimensional state $x' = g(x)$ in which the measurement equations are linear. (The appropriate transformation must be found by the engineer.) This quasi-linear model is processed by a Kalman filter using the transformed state space x'. An IEKF is used to transform the non-normal pdf from the x' space to the original x space; it gives a good approximation of the expected value and covariance if the measurements fully observe the system, see [4].

At every time step, an IEKF is constructed with a prior pdf having a mean derived from the current estimate of x' and a large covariance. The measurements are equal to the mean of the non-minimal state estimate ($h(x, z) = g(x) - \mu'_k = 0$) and the uncertainty on the measurements equals the covariance of the non-minimal state P'_k. The intial estimate of the prior pdf of the state x is important because this second filter is an IEKF that converges only when the estimates stay close to the real value, which is the case if the initial values are well chosen. A good initial estimate is obtained by approximating the minimal state using the current estimate of the non-minimal filter. Due to the non-minimal representation of a plane, a *singularity* exists if the parameters a, b, c and d are zero, which gives a solution without physical meaning. To prevent the Kalman filter from converging to this solution an constraint equation $1 = (a^2 + b^2 + c^2)$ is introduced.

3 Model reduction

Every non-elementary contact is modeled by a non-minimal number of geometrical parameters, e.g., two vertex-face contacts models use fourteen parameters, while a minimal edge-face contact model uses only eight parameters. The minimal contact model is selected by comparing the probabilities of various model hypotheses, [5], based on the Bayes factor (B), which is the posterior to prior odds ratio:

$$B = \frac{p(M_1|z)/p(M_2|z)}{p(M_1)/p_(M_2)} = \frac{p(z|M_1)}{p(z|M_2)}; \tag{2}$$

The Bayes factor for one model against another model is the posterior odds for that model against the other, when neither model is favored a priori ($p(M_1) = p(M_2)$). It has the following advantages: (i) it incorporates prior knowledge about statistical uncertainties, (ii) it can guide evolutionary model-building (because it does not require the various hypothis models to be "nested"), (iii) it does not suffer from approximations errors. (iv) it evaluates different models without fixing to a particular choice of the parameter value, but rather averages over all possible parameters using the prior distribution.

A Kalman filter produces at every moment an estimation of the state parameters $p(x|M, z_{1,..,k})$, which can be used as prior distribution before a measurement has been added. This joint density can be written as:

$$p(x, z|M, z_{1,...,k}) = p(z|x, M)p(x|M, z_{1,..,k}); \tag{3}$$

Where $p(z|x, M) = l$ is called the likelihood of the model M given the state x and $p(x|z_{1,...,k}, M)$ is the prior knowledge concerning the states of model M. An integration of this joint density with respect to the state leads to an exact integrated likelihood. This integral is approximated around the current state estimate:

$$\begin{aligned}\log(p(x, z|M, z_{1,..,k})) &= \log(p(\hat{x}, z|M, z_{1,..k})) + \dots \\ &G(\hat{x}, z)(x - \hat{x}) - 0.5(x - \hat{x})^T H(\hat{x})(x - \hat{x}).\end{aligned} \tag{4}$$

$G(\hat{x}, z)$ is the gradient and $H(\hat{x})$ the negative Hessian of $log(p(x, z|M, z_{1,...,k}))$ evaluated at $x = \hat{x}$. This approximation is exact for a normal distributed posterior. In our case we assume a normal distributed posterior function because an IEKF is used to estimate this posterior. Our current estimate of $\hat{x}$ is the maximum posterior estimate of the joint density due to the use of Kalman filter estimations. Therefore the estimate satisfies $G(\hat{x}, z) = 0$. The integrated likelihood results in the integration of a scaled normal distribution with mean $\hat{x}$ and covariance matrix equal to the inverse of the hessian $H(\hat{x})$:

$$p(z|M, z_{1,\ldots,k}) = p(\hat{x}, z|M, z_{1,..k})(2\pi)^{0.5d} \int e^{-\frac{1}{2}(x-\hat{x})^T H(\hat{x})(x-\hat{x})} dx, \tag{5}$$

$$= p(\hat{x}, z|M, z_{1,..k})(2\pi)^{0.5d} \det H(\hat{x})^{-0.5}. \tag{6}$$

3.1 Implementation of Bayes factor into NMSKF

New hypothetical measurement models can be inserted into the first or the second step of the filtering process (NMSKF or IEKF step). Insertion into the first step implies that every hypothetical model requires the start of two independent NMSKF processes. Insertion into the second step implies that the estimated posterior is re-used in every independent IEKF process. This yields a computational advantage that is our motivation for chosing this type of hypothesis testing. The construction of a new hypothetical model results in reduction of the parameter space x by imposing relations between different state variables. For example, a hypothesis stating that the contacting face of the first contact point (a^1, b^1, c^1, d^1) coincides with that of the second contact point (a^2, b^2, c^2, d^2) imposes the following equalities in the state space x: $a^1 = a^2$, $b^1 = b^2$, $c^1 = c^2$ and $d^1 = d^2$. This results in reduction of the state space from n parameters to a state space x_r of $n-4$ parameters. This reduction leads to the construction of a new measurement equation: $h_r(x_r, \mu_k')$. This new measurement model is inserted into the IEKF corresponding to this hypothesis.

4 Experiments

The experimental data is collected from an assembly of a cube with side $l = 250\ mm$ in a corner executed by a KUKA 361 industrial robot arm with a 6D JR3 force/torque on its wrist, Fig. 1. This compliant motion is specified manually using the Task Frame Formalism [6] and is executed by a hybrid force/position controller. The geometrical parameters of both the manipulated object (cube) and that of the environment object (corner) are supposed to be unknown. The initial position of the cube is such that no edge-edge or edge-vertex CF contacts occur. The estimates are obtained by off-line processing of the measurements of wrenches and poses collected at 10 Hz. The estimators are able to process the measurements on-line. Processing all measurements by a C++ program [1], on a *Pentium 4* 2 GHz, 500 Mb RAM laptop goes much faster than the measurement frequency, however this is has not yet integrated in our current software infrastructure. After detection of a new CF the pose and wrench measurement equations and ϕ limiting equation are adapted. This detection is, for the time being, still performed manually by measuring inconsistencies with the measurement equations using a SNIS-test.

[1] http://people.mech.kuleuven.ac.be/~kgadeyne/bfl.html

Initial estimates. For the vertices, we take the origin of a gripper frame g ($p_x = 0, p_y = 0, p_z = 0$) with covariances: $\sigma^2_{p_x} = l^2$, $\sigma^2_{p_y} = l^2$ and $\sigma^2_{p_z} = l^2$. The initial estimate of (a, b, c) is derived from the current estimate of the non-minimal filter, and the fourth parameter d is set to zero; the covariances are $\sigma^2_a = \sigma^2_b = \sigma^2_c = 2^2$, and $\sigma^2_d = 5l^2$. In order to cope with linearisation errors, the experimentally determined pose and wrench measurement standard deviations are multiplied by 5. The unmodeled friction is taken into account by multiplying once again the wrench standard deviation by 50: the friction force is approximatly 6-7 Newton (N) for a contact force of 25 N (in the most constrained case of five CFs), and the noise on the force sensor measurements is approximatly 0.1 N, resulting in a multiplication by 50 to "cope" with the friction effects.

Results. The experiment consists in the following sequence of CFs: vertex-face, edge-face, face-face, edge-face and face-face, 2x face-face, 3x face-face, Fig. 1. The first five are recognized and modeled as shown in Fig. 3. Fig. 3 shows the evolution of the estimated geometrical parameters. Only the geometrical parameters that are relevant to the geometrical model building process are visualised. Meaning that the parameters estimates that have a large uncertainty are omitted. Fig. 4 visualizes the estimated vertices and faces, as obtained at the end of each CF, and quantifies the numerical accuracy of the estimations at the end of the filtering process.

The first contact is a vertex-face contact, starting at measurement 1 and ending at measurement 946. Sub-figures E_1 and E_2 show the evolution of the estimated parameters of the face and the vertex, respectively. Sub-figure R_1 visualizes the estimated vertex ($V1$) and face (A) at the end of the CF, i.e., just before transition to the edge-face contact. The vertex is shown by its estimated position and its 2σ uncertainty ellipsoid.

The second contact formation, between measurements 964 and 1830, consists of two vertex-face contacts. After hypothesis testing using a Bayes factor, an edge-face contact model becomes the most plausible. This allows a reduction of the number of parameters from fourteen (two times seven) to eight: seven parameters from the first contact (E_1 and E_2) and one new parameter describing the unaccuratly known position of the second contact ($V2$) on the edge. The model imposes no limitations to the position of the vertex $V2$ on the edge, leading to a large uncertainty on the estimate. An edge (D_1) containing $V1$ and $V2$ is added to the 3-D model (R_2).

The third contact formation, between measurements 1837 and 2518, consists of three vertex-face contacts. After hypothesis testing, a face-face contact becomes the most plausible CF. This results in a reduction of number of parameters from twenty-one (three times seven) to ten: eight from the edge-face contact and two new parameters that describe the unaccuratly known position of the contact ($V3$) in the face of the manipulated object (A').

The fourth contact formation, between measurements 2535 and 3720, adds

a new vertex-face to the previous face-face model. After hypothesis testing, an additional vertex-face that lies in a plane of the cube that goes through the edge D_1 and perpendicular to the xy-plane becomes the most plausible. This results in a reduction of the number of parameters from twenty-eight to fifteen: ten parameters describing the face-face contact and five new parameters that describe a new vertical environmental face (E_3,B) and the position of the fourth contact ($V4$). This position is very well determined in two directions ($E_4 = p_{x2}$ and $p_{y2} = p_{y1}$) and uncertain in the third direction (z).
The last contact formation, between measurements 3735 and 4116, adds an extra vertex-face contact to the previous model. After hypothesis testing, the fifth vertex ($V5$) lies in the second face (B) and in the previously described vertical face of the cube; therefore, the position of the vertex in this face is very uncertain in the other two directions.
An extra figure (2) describes the recognition of the edge-face contact model using a SNIS consistency test, that indicates if the measurements are consistent with the assumed model, and by comparing different hypothetical models.

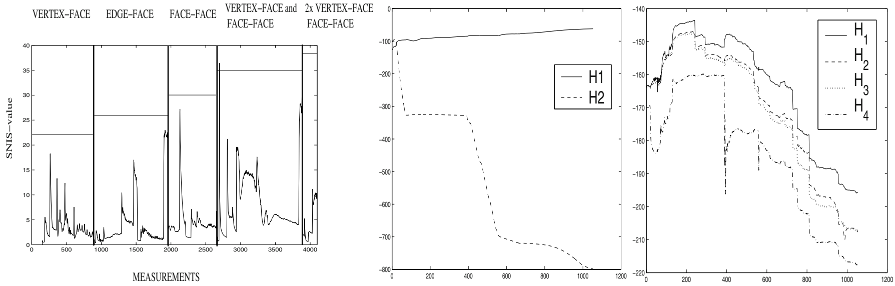


Fig. 2. The left subfigure visualises the evolution of the Snis-values during all the different contact situation, the horizontal line is the transition detection border. The middle subfigure compares the integrated likelihood of one vertex-face contact (H2) versus two vertex-face contacts (H1) when the second contact situation is reached. The right subfigure compares different hypothetical models when the two vertex-face contacts is reached : an edge-face contact situation (H1), a face-face contact situation (H2), two vertices contacting a ground face (H3) and no hypothesis (H4).

5 Conclusion

We have shown that geometric parameters and contact configuration transitions of moderately complex contact situations can be estimated autonomously, during force-controlled compliant motions. The presented Bayesian approach is general, and capable of working with very large initial estimation errors. In addition, it can perform automatic model reduction, i.e., it can detect whether

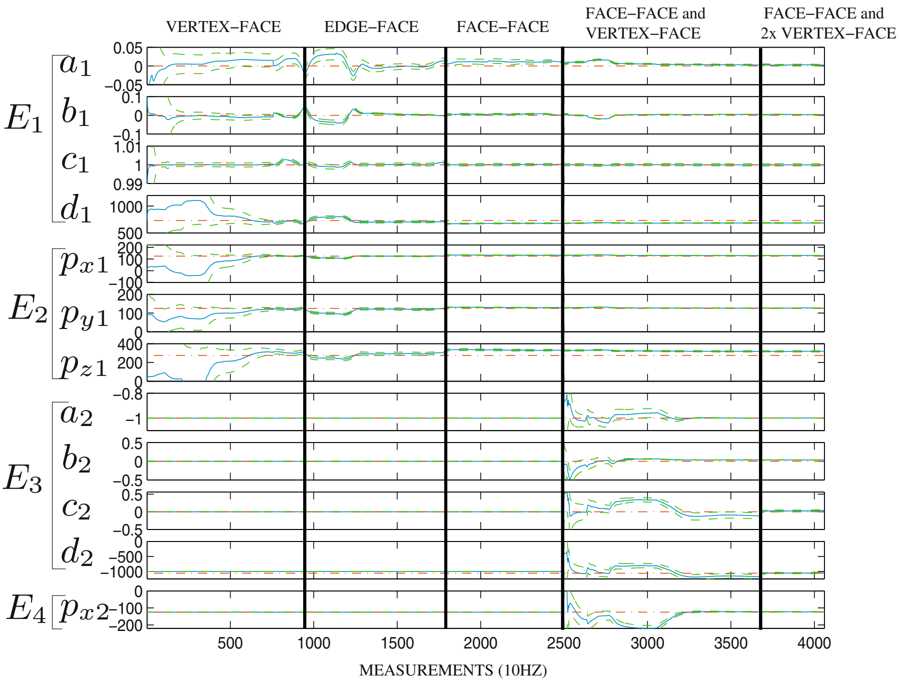


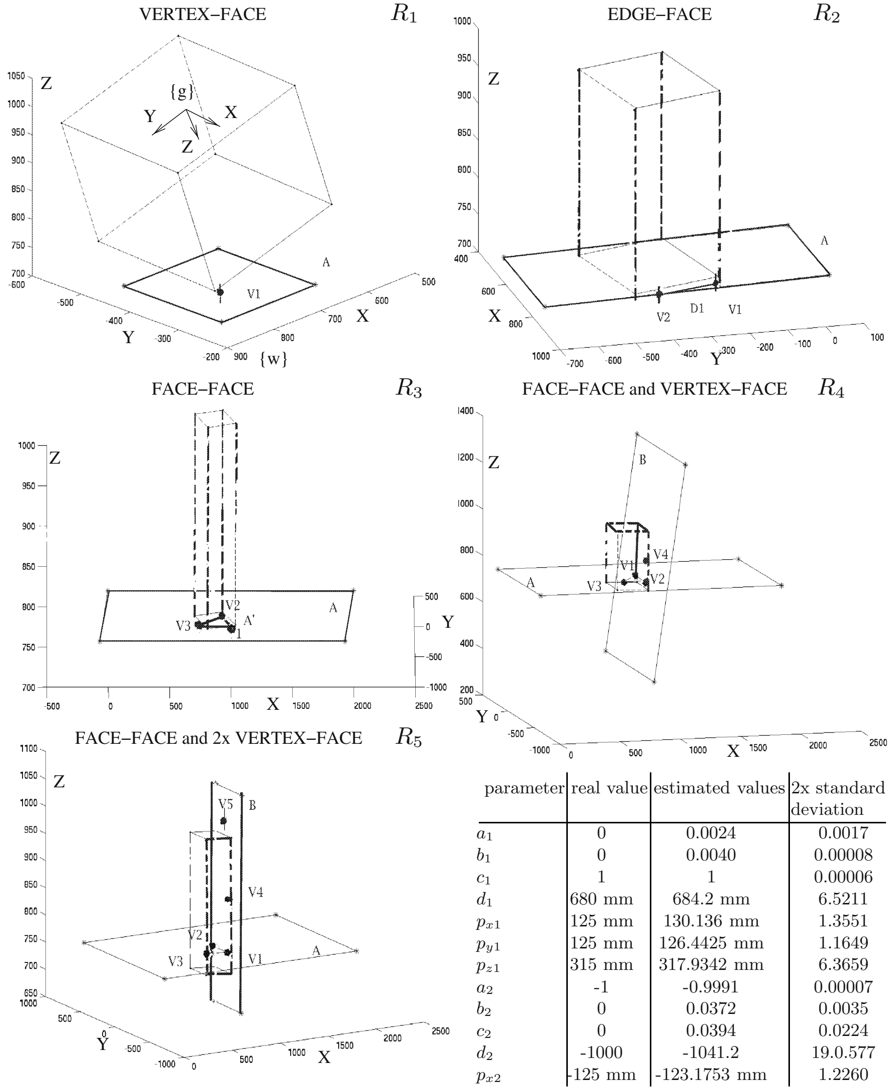
Fig. 3. This figure visualizes the state estimations (full lines) and twice their standard deviation (dashed lines) and their real values (dashed-dotted lines):E_1 the parameters of the ground plane of the corner (A), E_2 the coordinates of the vertex $V1$, E_3 the parameters of a vertcal plane B of the cube, E_4 the x coordinate of the fourth vertex-face contact

a set of *vertex-face* contacts belong together to form a "higher-level" contact situation such as *edge-face* or *face-face*. Our algorithms are not limted to the particular task we have shown, except for the fact that a "model library" of only polyhedral contact situations is used.

Our main short term research approach is to integrate this estimation process into a real-time controlled robot system in order to perform compliant motion without the need of a model of the environment. Next we will focus on the automatic generation of all the plausible hypothesis by taking into account all the collected knowledge about the environment.

Acknowledgment

All authors gratefully acknowledge the financial support by K.U.Leuven's Concerted Research Action GOA/99/04. T. Lefebvre is a Postdoctoral Fellow of the Fund for Scientific Research–Flanders (F.W.O.) in Belgium.



parameter	real value	estimated values	2x standard deviation
a_1	0	0.0024	0.0017
b_1	0	0.0040	0.00008
c_1	1	1	0.00006
d_1	680 mm	684.2 mm	6.5211
p_{x1}	125 mm	130.136 mm	1.3551
p_{y1}	125 mm	126.4425 mm	1.1649
p_{z1}	315 mm	317.9342 mm	6.3659
a_2	-1	-0.9991	0.00007
b_2	0	0.0372	0.0035
c_2	0	0.0394	0.0224
d_2	-1000	-1041.2	19.0.577
p_{x2}	-125 mm	-123.1753 mm	1.2260

Fig. 4. This figure makes a visualization of the constructed geometric 3-D model and of the final numerical results, R_1: a vertex-face contact with vertex ($V1$) and face (A), R_2: an edge-face contact with an extra vertex ($V2$) on the edge (D_1), R_3: a face-face contact with an extra vertex ($V3$) leading to an estimated face of the manipulated object (A'), R_4: a face-face and vertex-face contact with an extra vertex ($V4$) leading the estimation of a new face of the environment (B), R_5: a face-face contact combined with two vertex-face contacts ($V4$, $V5$) in the same face B, right bottom: the geometrical estimates and their corresponding accuracy (2x standard deviation) are compared with the real values. The vertices ($V1, V2, V3, V4, V5$) are expressed in a gripper frame g and the faces (A,B) in a world frame w shown in figure R_1.

References

1. T. Lefebvre, K. Gadeyne, H. Bruyninckx, and J. De Schutter, *Bayesian Statistics 7*, chapter Exact Bayesian inference for a class of Nonlinear systems with Application to robotic assembly, Oxford University Press, 2003, pp 587-596.
2. H. Jeffreys, *Theory of probability*, Oxford University Press, Oxford, UK, 3rd edn edition, 1961.
3. T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Polyhedral contact formation modeling and identification for autonomous compliant motion," *IEEE Transactions On Robotics And Automation*, vol. 19, no. no. 1, pp. pp 26–41, February 2003.
4. T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Kalman filters for nonlinear systems: a comparison of performance," *The International Journal of Control*, 2004, (in press).
5. R. Kass and A.E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, pp. 773–795, 1995.
6. H. Bruynickx and J. De Schutter, "Specification of force-controlled actions actions in the task frame formalism: A synthesis," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 581–589, August 1996.

Stiffness Imager

Makoto Kaneko[1], Tomohiro Kawahara[1] and Shinji Tanaka[2]

[1] Graduate School of Eng., Hiroshima University, Higashi-Hiroshima, Japan
{kaneko, kawahara}@hfl.hiroshima-u.ac.jp
http://www.hfl.hiroshima-u.ac.jp

[2] Graduate School of Biomed. Sciences, Hiroshima University, Hiroshima, Japan
colon@hiroshima-u.ac.jp
http://home.hiroshima-u.ac.jp/naika1

Abstract. This paper discusses the Stiffness Imager that can provide us with the pattern of stiffness (or pseudo-stiffness) distribution of environment. While there are four combinations between force applying method and the way for measuring the displacement, we pick up the combination in which the force is given to the environment at a local point and the displacement of environment is measured over the neighboring area where the force is given. We apply the idea to an endoscope camera, so that we can obtain not only visual information but also the pseudo-stiffness pattern. We also show a couple of experimental results exhibiting the pseudo-stiffness pattern.

1 Introduction

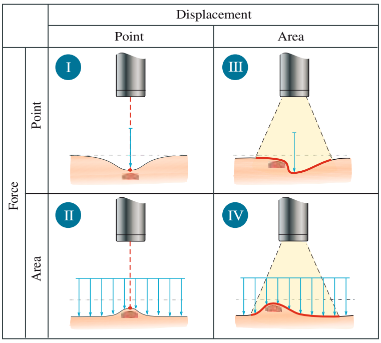


Fig. 1. Four types of Stiffness Imager

For detecting the existence of cancer in human body, the Positron Emission Tomography (PET) is used as a powerful tool[1]. It tells us visually whether there exist cancer tissues in our body or not and rough position where they are if this is the case. Toward a further treatment or an operation, however, it never tells us the quality of cancer, such as how advanced and spread they are. To cope with this, both endoscope camera and ultrasonic probe should take over a further medical examination in case of stomach cancer. An endoscope camera provides us with visual information of the surface of stomach, for example. Due to the direct observation, it is the most convenient tool for examining cancer, especially located at the surface. On the other hand, however, such an endoscope camera can never obtain any information of tissues locating behind the surface of stomach. An ultrasonic probe is often utilized as an alternative tool. The probe enables us to examine the tissue behind the surface of stomach through the visualization of density distribution of tissues. However, since stomach has to be filled with water for obtaining an appropriate signal from the probe, patients feel painful during examination. Knowing that the density distribution and stiffness distribution are close relationship each other, we can equivalently obtain the inner information through the examination of stiffness distribution. If we can combine the capability of obtaining the stiffness distribution with an endoscope camera, patients are released from filling with water in stomach. Except this example, there are various cases where stiffness information is really crucial in diagnosis of our body, such as inspection of gum, medical examination of tumor of throat, and medical

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 581–587, 2006.

examination of internal organs. One remark is that a medical examination strongly requests a non-contact method for avoiding any damage of tissues.

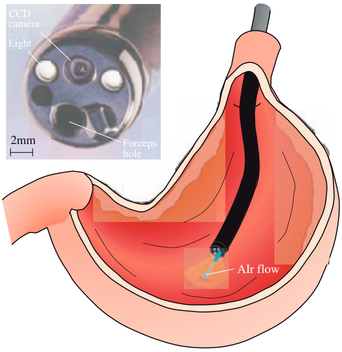


Fig. 2. Application to endoscope camera

Based on the above background, we discuss the Stiffness Imager capable of providing the stiffness distribution of environment to be inspected. Fig. 1 shows a conceptual image of Stiffness Imager, where a force is given to the environment uniformly in Fig. 1 II and IV while it is given at a point in Fig. 1 I an III, respectively. Suppose that the environment is flat along the horizontal direction, as shown in the dotted lines before imparting a force. Without any active force impartment, we can neither detect how the stiffness distribution is, nor even how pseudo-stiffness distribution is. However, when giving a uniform force to the environment, we can clearly see the stiffness distribution of environment as the displacement pattern, as shown in Fig. 1 IV. This pattern can be obtained through CCD camera, for example. Fig. 1 shows a family of Stiffness Imager, while the combination shown in Fig. 1 II does not make much sense.

This paper picks up the combination III in Fig. 1, where the force is given at the particular point and the displacement patter is obtained by a CCD camera. Even for the case of a combination shown in Fig. 1 III, we can observe the pattern of pseudo-stiffness distribution around the point where the force is imparted. Since it is hard to impart a uniform force as shown in Fig. 1 II or IV, the combination III is perhaps a practical one for discriminating a stiff point from neighbors. We would note that we can easily impart a force to an environment by using an appropriate air jet[8]. Such a non-contact approach releases us to impart a serious damage to the environment. From a viewpoint of medical application, this is a big advantage compared with a contact based approach. Fig. 2 shows an application example to an endoscope camera for examining our stomach.

This paper is organized as follows. After briefly reviewing related works, we explain the basic working principle of the Stiffness Imager in section 3. In section 4, we show a couple of preliminary experiments toward the implementation to an endoscope. In section 5, we give some issues to be cleared for applying an endoscope camera, before concluding remarks.

2 Related Works

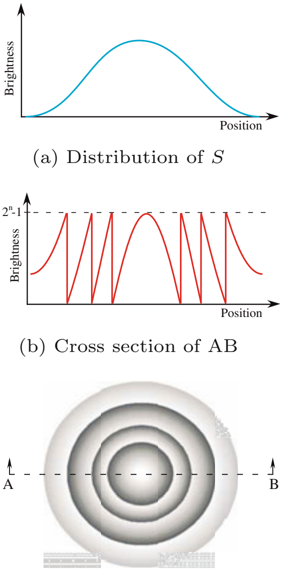
Medical: There have been many works [2]-[7] discussing medical robots, such as minimum invasive surgery, brain surgery, bone surgery, heart surgery, catheter based inspection, tissues model for medical application, design of catheter, training for manipulating a catheter and so forth. While a catheter is a good tool for diagnosing inner surface of internal organs, most conventional catheters have been focused on visual based inspection. Toward impedance based inspection in addition to visual one for tissues, we have proposed an approach for non-contact impedance measurement of tissues with combination of an air-jet force probe and a laser distance sensor [8]. Through the work, we learnt that such a pin-point based scanning takes time for examination and it is hard to implement a laser projection equipment at the tip face of endoscope due to insufficient space.
Active Sensing: The Stiffness Imager is a kind of active sensor with combination of vision and force probe. As for tactile based active sensing, various approaches

have been proposed for detecting the contact point between flexible beam(s) and environment through the measurement of either stiffness or impedance by using an appropriate force sensor and a local position sensor [9]-[13]. Also, there are a couple of works discussing the contact point or the contact stiffness with combination of vision(s) and a force probe [14][15], while they have addressed only for getting local information of environment. We believe that a big advantage for using a vision is that global information can be obtained simultaneously. We proposed the Non-Contact Stiffness Imager[16] where we showed the advantage for using the non-contact version compared with contact one. This paper is an extended version for our former papers.

3 Non-Contact Stiffness Imager for Endoscope

As well known, an endoscope camera is normally composed of a flexible tube controlled by wires, a stereo CCD camera, and fiber cables for guiding lightening signal. The flexible tube has a hole for inserting a forceps as well. Through this hole, an air jet is supplied such that we can impart a force to the surface of stomach. Instead of projecting a light pattern actively, we directly utilize the captured image without any active pattern and the captured data are fed into the image converter where marks are artificially added so that medical doctors can clearly see the stiffness distribution visually.

For a constant point force, we can regard the distribution of stiffness as that of displacement. Generally, however, it is hard to see the displacement pattern from the vision data captured by the CCD camera, while it can obtain a contour map pattern using the brightness distribution of the light source of endoscope. In order to emphasize the change of brightness due to the displacement of environment, we utilize the light saturation method so that we can make marks artificially. Suppose that the captured data have RGB-color information. For each pixel, we convert the brightness into black-and-white mono-color as follows:



(a) Distribution of S

(b) Cross section of AB

(c) Top view of D

Fig. 3. Brightness pattern

$$S(i,j) = \frac{R(i,j) + G(i,j) + B(i,j)}{3} \tag{1}$$

where $R(i,j)$, $G(i,j)$, and $B(i,j)$ are n-bits red, green, and blue level in the (i,j) pixel, respectively.

$$S'(i,j) = \alpha S(i,j) \tag{2}$$

where α is an appropriate positive value. Now, suppose a display with n-bit gray scale. Under a small displacement with $\alpha = 1$, from the gray image, it is hard to recognize for doctors how much the displacement is. In order to enhance the displacement image, we choose an appropriate large α so that $S'(i,j)$ may exceed 2^n in some pixels. By choosing such a large α, we can intentionally make $S'(i,j)$ overflow for pixels having a change of displacement, while pixels without displacement do not. We now rewrite $S'(i,j)$ as follows:

$$S'(i,j) = 2^n k_1 + D(i,j) \tag{3}$$

where $k_1 = 0, 1, \ldots,$ and $0 < D(i,j) < 2^n$. Suppose $0 < S'(i,j) < 2^n$ for every pixel under $\alpha = 1$. This means $k_1 = 0$, under which the brightness continuously changes between pixels when $D(i,j)$ is displayed on screen. As increasing α, however, $S'(i,j)$ will be more than 2^n, which implies $k_1 \geqq 1$. Under $k_1 \geqq 1$, the brightness abruptly changes between pixels when $D(i,j)$ is displayed. Such a boundary can be utilized as a mark in visualization of displacement. For example, when the environment deforms due to local compliance, circular marks will appear on screen especially around the point where the displacement happens. The local compliance of environment can be estimated how circular marks change. For example, suppose the brightness distribution of $S(i,j)$, as shown in Fig. 3(a). Fig. 3(b) and (c) show the distribution of $D(i,j)$ in the cross section and the top view, respectively. As seen from these figures, through the monitor display, we can estimate how the distribution of displacement is, and, as a result, where the stiff area is. We believe that this information may be useful for doctors to judge the quality of disease, such as width and the depth of cancer which can not see visually. We would note that since both light sources and CCD camera are ready for use in a conventional endoscope camera, the Stiffness Imager can be easily implemented to it by simply adding an air supply system, without any sensors improvement. The flexible tube of endoscope camera equipped with a hole for inserting a forceps further supports an easy implementation, because the tube can be utilized as an air supply one.

4 Experiments

4.1 Experimental Setup

Fig. 4 shows an experimental setup mainly composed of a computer, an endoscope camera and an air-supply system. The air-supply system is composed of a compressor for producing pressurized air, an accumulator for keeping pressurized air, a high speed solenoid valve for changing the applied force quickly, a video processor for processing visual data, and a PC for sending a control command to the valve as well as for controlling the video processor. Fig. 5 shows the three components of the specially designed high speed solenoid valve. The pressurized air is guided into the end face through the small hole for inserting a forceps and supplied to the environment. The CCD camera captures the image of environment. This information is fed into the computer through the video processor and displayed in a monitor with 320 × 240 pixels after processing data in each pixel.

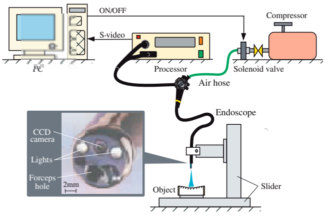


Fig. 4. Experimental system

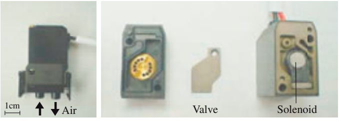


Fig.5. Components of high speed solenoid valve

4.2 Experimental Results

Fig. 6 shows the test piece where the object includes partial stiff area with the number between 4 through 6. Fig. 7 shows an experimental result where (a) and (b) are direct image through CCD camera and the $D(i,j)$ map after processing, respectively, and each number corresponds to that shown in Fig. 6. Force is not given for figures with the serial #1, while a constant force is given for others. From these figures, we can clearly observe the point where the air force is given. Also, we can see the change of the $D(i,j)$ map when the probe comes to the area of the stiff piece. Especially, see between #4 and #6. Through this experiment, we can clearly observe the change of circular pattern coming from the existence of piece with high stiffness. As shown in these experimental results, the Stiffness Imager can provide us with a distorted image when there exists a stiffer point compared with neighbors.

5 Discussions

Fig. 8 shows both the direct CCD image and $D(i,j)$ map when the tip of the endoscope is manually moved in human stomach, where an air jet is not in operation. These data under the shutter of air jet can be easily obtained without any medical permission, since we can directly utilize the image data coming from an endoscope camera. Through these preliminary experiments, we found a couple of issues to be cleared for actual implementation. One is the halation where an illuminated light reflects on the surface of stomach, since our stomach is partially wet and shinny just like a mirror. Such a halation makes it difficult to separate the brightness change of the detected light caused by the distance change due to the pushing force from the one cased by the light refection due to halation. This is actually a serious issue to be cleared. The second issue is that the head of endoscope moves dynamically by a reaction force when an air jet is given to an environment, as shown in Fig. 9. Once such a head swinging motion happens, it is hard to estimate how much force is given and where the force is given, since we lose the origin of base coordinate system of the endoscope.

6 Conclusions

We discussed the Stiffness Imager capable of showing the change of local stiffness as patterns. As an application example, we showed the endoscope camera with the function of Stiffness Imager and showed an approach how to visually enhance the displacement of tissue for a given point force. We confirmed the effectiveness of the idea through experimental works. Due to the simplicity of the Stiffness Imager, we can easily apply the endoscope to human without any danger. We also showed a couple issues to be cleared toward an actual implementation of the hybrid type endoscope camera capable of detecting both visual imformation and stiffness pattern.

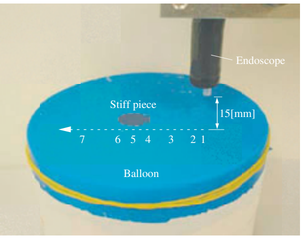
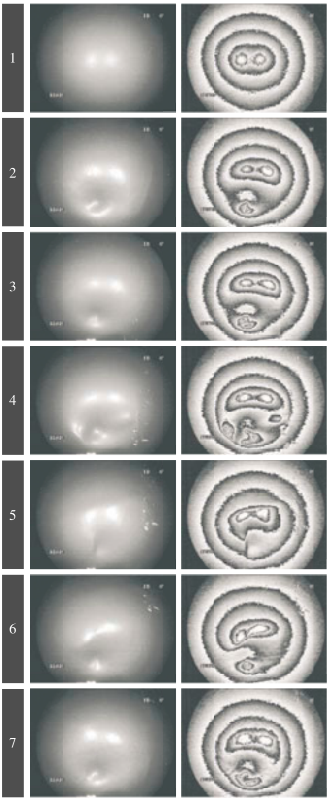


Fig. 6. Test piece

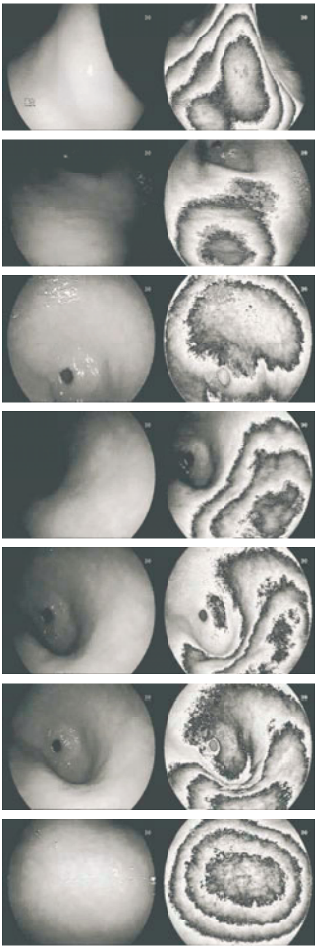


(a) Real image (b) $D(i,j)$ map

Fig. 7. Experimental results

Acknowledgement

We would like to express our sincere thanks to Mr. Satoshi Matsunaga, Mr. Kan'ichi Tokuda, Mr. Shinji Takeuchi and Dr. Yukio Hosaka for their helps for this work.



(a) Real image (b) $D(i,j)$ map

Fig. 8. Image of human stomach

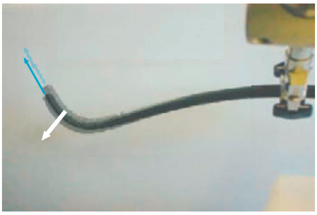


Fig. 9. Reacting motion of endoscope

References

1. Biomedical Research Foundation of Northwest Louisiana: *http://www.biomed.org/pet/*.
2. K. Ikuta, T. Hasegawa, and S. Daifu: Hyper redundant miniature manipulator "Hyper Finger " for remote minimally invasive surgery in deep area, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1098–1102, 2003.
3. R. Moser, R. Gassert, E. Burdet, L. Sache, H. R. Woodtli, J. Erni, W. Maeder, and H. Bleuler: An MR compatible robot technology, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.670–675, 2003.
4. R. H. W. Lam, and W. J. Li, and N. Xi: A wireless temperature measurement guide rod for internal bone fixation surgery, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1768–1773, 2003.
5. D. Liu and T. Wang: Study on robot-assisted minimally invasive neurosugery and its clinical application, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.2008–2013, 2001.
6. K. Ikuta, H. Ichikawa, K. Suzuki, and T. Yamamoto: Micro hydrodynamic actuated multiple segments catheter for safety minimally invasive therapy, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.2640–2645, 2003.
7. C. Mendoza and C. Laugier: Simulating soft tissue cutting using finite element models, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1109–1114, 2003.
8. M. Kaneko, T. Kawahara, S. Matsunaga, T. Tsuji, and S. Tanaka: Tauching stomach by air, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.664–669, 2003.
9. M. Kaneko: Active antenna, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.2665–2672, 1994.

10. N. Ueno and M. Kaneko: Dynamic active antenna, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1784–1790, 1994.
11. M. Kaneko, N. Ueno, and T. Tsuji: Active antenna: basic consideration on the working principle, *In Proc. of the IEEE Int. Conf. on Intelligent Robotics and Systems*, pp.1774–1750, 1994.
12. N. Ueno, M. Kaneko, and M. Svinin: Theoretical and experimental investigation on dynamic active antenna, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.3557–3563, 1996.
13. N. Ueno, M. Kaneko, and M. Svinin: Dynamic active antenna considering with multi oscillation modes, *Journal of Advanced Robotics*, vol.15, no.1, pp.61–67, 1997.
14. M. Kaneko, N. Kanayama, and T. Tsuji: Vision based active antenna, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.2555–2560, 1996.
15. N. Kanayama, M. Kaneko, and T. Tsuji: On 3D vision based active antenna, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.143–148, 1997.
16. M. Kaneko, T. Kawahara, and S. Tanaka:Non-Contact Stiffness Imger, *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1562–1567, 2004.

Towards High-Fidelity On-Board Attitude Estimation for Legged Locomotion via a Hybrid Range and Inertial Approach

Surya P. N. Singh and Kenneth J. Waldron

Robotics Locomotion Laboratory (LOCOLab)
Department of Mechanical Engineering
Stanford University
Stanford, CA 94305, USA
(spns,waldron)@cdr.stanford.edu

Abstract. Legged robots display a characteristically periodic motion. Measuring and tracking this motion has traditionally been performed using general inertial measurement techniques. While widely applied in robotics, this approach is limited in dynamic legged locomotion due to the excessive accumulation of drift from severe impact shocks (nearly $9g$ in single leg experiments). This paper introduces the attitude estimation problem for legged locomotion and shows preliminary results from a more powerful combined range and inertial sensing approach. Based on a modified Extended Kalman Filter the method uses ground-directed range sensors, the stride period, and other periodic features of legged locomotion in order to address inertial drift. Together this provides rapid, robust estimates of flight phases and attitude necessary for extended dynamic legged operations.

1 Introduction

Most natural and architectural environments are well suited for legged locomotion. Humans and other animals easily traverse terrains without requiring a cleared path, as is necessary for wheeled vehicles. Recent advances in the design of legged robotic platforms have led to a variety of new roles for legged robots, ranging from dancing humanoids [1] to exploring hexapods [2,3]. Because these robots locomote at higher speeds and with greater agility [2], they need to reliably assess the environment and rapidly estimate their position within this environment, accurately and robustly.

Localization of position and attitude is critical for both control and utility of the robot. This can be performed using a variety of methods ranging from inertial techniques to Simultaneous Localization and Mapping (SLAM) [4]. Inertial navigation is an inherently difficult problem due to the inevitable drift in the estimation. The long-term stability of the sensed position is diminished by integration errors, noisy readings, and offset errors. Gravitational and inertial accelerations are inseparable (for a given measurement), thus small alignment or calibration errors may appear as robot accelerations which, when integrated, results in position errors that grow quadratically in time. The typical solutions to this problem (such as external updates via the Global Positioning System (GPS) or passive-marker optoelectronic systems) are limited by sensor eccentricities, bandwidth, and occlusions to update signals [5].

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 589–598, 2006.

Inertial drift is especially pronounced in dynamic legged locomotion. Legged machines tend to experience repeated impacts (due to foot fall) and have large magnitude disturbances (due to terrain irregularities). The impacts tend to excite secondary motions which can not be readily filtered as they are often at frequencies of interest for navigation. However, this vexing behavior is not without its applications. Most legged gaits tend to be periodic and, further, many tend to be symmetric. Assuming the full kinematics of the robot are known, a partial state estimate update during ground contact is possible since the leg serves as a kinematic chain between the ground and the robot origin. This method is complicated, however, due to the compliance of the leg members and foot placement uncertainty.

This paper describes how these general approaches can be used with estimation methods such as the Extended Kalman Filter (EKF). Interestingly, the EKF provides a convenient framework for integrating measurement updates from redundant sources, and thus yields a high-fidelity attitude (and position) estimator [6]. The paper describes initial experimental results using this framework as tested on a single leg connected to an instrumented boom-arm platform are detailed. The approach is currently being adapted to a galloping quadrupedal robot, KOLT.

2 KOLT Platform Description

The Kinetically Ordered Locomotion Test (KOLT) quadrupedal robot Figure 1) is being developed to test high-speed (up to 7 m/s [25 km/h]) dynamic robot locomotion by galloping [7]. Galloping is fast, nimble gait that is preferred for quadrupeds moving at high speeds [8].

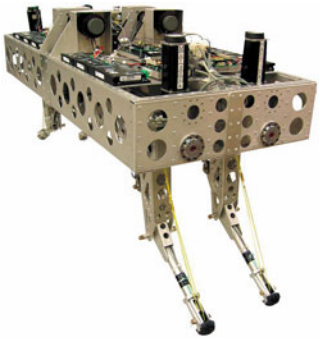


Fig. 1. The KOLT robot ($2m$ long, $90kg$ weight) has been constructed specifically to study dynamic maneuvers of legged systems. The four identical legs each have three degrees of freedom and thrust primarily along the leg axis.

3 Method

Attitude estimation on KOLT follows the standard "sense then update" paradigm [6] whereby the inertial motions (at high rates) are corrected with redundant measurements obtained from imprecise, low-bandwidth range sensors via an EKF framework. We have modified this approach to include a more detailed sensor model with features that are unique to dynamic legged locomotion, such as the regular stride periods, parabolic motion, and (depending on gait) symmetry.

3.1 Sensing Elements

The KOLT uses a Microstrain 3DM-G-ADI IMU based on micromachined inertial sensing elements. Each leg is instrumented with a MEMSIC MXA2050AL Dual-axis $\pm 10g$ accelerometer; and a Sharp GP2Y0A02YK long distance (150 cm), infrared (IR) range sensor. Proprioception is performed using 1000-count encoders at all of the actuated joints.

3.2 Attitude estimation from distributed range measurements

Updates from external aids, such as external cameras, cannot be relied on due to potential for occlusion and the extended ranges of operation needed for gallop. Thus, a self-contained method is desired.

The use of range data to provide a measure of orientation has been shown and is derived for an ideal planar case in [9]. The KOLT expands on the ground range approach illustrated in [9,10] by using these sensors to estimate attitude and to use this data in a batch process with the inertial data. The ground range measurement is obtained using low-cost Sharp infrared range sensors. These miniature and mechanically robust sensors are low-weight and easily exchanged. They are not-ideal as they have delays (5 ms), low-bandwidth (20 Hz), significant noise, variation to reflected surface color/texture, and a highly non-linear response characteristic [11]; thus, resulting in a non-trivial data filtering process.

The attitude is calculated in a spatial context by revisiting the derivation of the range equations using frame transformations (Figure 2) instead of a purely geometric approach as used in [9]. The orientations are solved using a pitch-roll-yaw or θ_y-then-θ_x-then-θ_z rotation sequence.

The sensors are placed near the ends of the robot side so as to increase the baseline and sensitivity height difference measurement. Analysis makes the following assumptions: sensor positions (S_i) with respect the body frame (B) are known and given by the vector $\mathbf{r}_{ib} = \left[\, x_i \; y_i \; z_i \,\right]$, the ground is planar (i.e., P$_i$ is in the *xy* plane of the origin frame), and a lambertian surface (i.e., reflection response independent of inclination). By further assuming that the beam direction is perpendicular to the *xy*-plane of the body (i.e., l_i aligned to the $\mathbf{z}_B$ axis), the projection (P) can be in described in body and fixed frames. Thus, the position of any i^{th} sensor relative to the fixed frame can be expressed as:

$$\mathbf{p}_i = \mathbf{M}\mathbf{r}_{ib} + \mathbf{p}_B \tag{1}$$

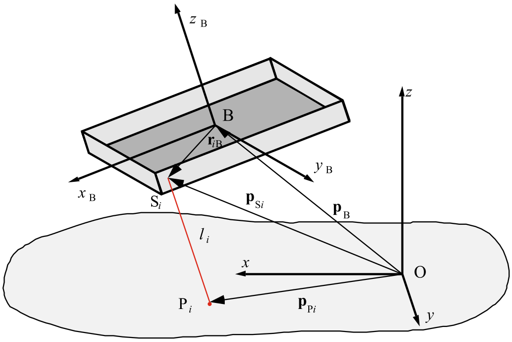


Fig. 2. Spatial estimation of attitude determination using range sensors (S_i located by vector $\mathbf{r}_{iB}$) that measure the distance to point P_i

where:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta_x & -S\theta_x \\ 0 & S\theta_x & C\theta_x \end{bmatrix} \begin{bmatrix} C\theta_y & 0 & -S\theta_y \\ 0 & 1 & 0 \\ S\theta_y & 0 & C\theta_y \end{bmatrix} \begin{bmatrix} C\theta_z & -S\theta_z & 0 \\ S\theta_z & C\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} C\theta_y C\theta_z & -C\theta_y C\theta_z & -S\theta_y \\ C\theta_x S\theta_z - S\theta_x S\theta_y C\theta_z & C\theta_x C\theta_z + S\theta_x S\theta_y S\theta_z & -S\theta_x C\theta_y \\ S\theta_x S\theta_z + C\theta_x S\theta_y C\theta_z & S\theta_x C\theta_z - C\theta_x S\theta_y S\theta_z & C\theta_x C\theta_y \end{bmatrix} \tag{2}$$

and

$$\mathbf{p}_B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \tag{3}$$

The perpendicular beam projection assumption gives that orientation of the beam (l_i) can be represented by a unit vector parallel to the z_B axis ($\mathbf{k}_b$). Thus, the direction of the beam relative to the fixed frame is:

$$\mathbf{k}_r = \mathbf{M}\mathbf{k}_b = \begin{bmatrix} -S\theta_y \\ -S\theta_x C\theta_y \\ C\theta_x C\theta_y \end{bmatrix} \tag{4}$$

Thus, the position of the point P becomes:

$$\mathbf{p}_i - l_i \mathbf{k}_r = \mathbf{M}\mathbf{r}_{ib} + \mathbf{p}_G - l_i \mathbf{k}_r \tag{5}$$

The planar ground assumption gives that the z component of the $\mathbf{p}_i$ is zero. so:

$$\mathbf{k} \cdot (\mathbf{M}\mathbf{r}_{ib} + \mathbf{p}_G - l_i \mathbf{k}_r) = 0 \tag{6}$$

which gives:

$$\mathbf{k} \cdot \mathbf{M}\mathbf{r}_{ib} = C\theta_x S\theta_y x_i + S\theta_x y_i + C\theta_x C\theta_y z_i \tag{7}$$

thus, the governing equation for each sensor (S_i) is:

$$C\theta_x S\theta_y x_i + S\theta_x y_i + C\theta_x C\theta_y (z_i - l_i) + z_B = 0 \tag{8}$$

For the three range sensors this will result in a simultaneous set of equations of the form in (8). The solution is obtained by eliminating the height ($\mathbf{z}_G$) and then the roll (θ_x) which gives the solution for the pitch (θ_y) as:

$$tan(\theta_y) = \frac{(y_2 - y_1)\left((z_3 - z_1) - (l_3 - l_1)\right) - (y_3 - y_1)\left((z_2 - z_1) - (l_2 - l_1)\right)}{(y_1 - y_2)(x_3 - x_1) + (y_3 - y_1)(x_2 - x_1)} \tag{9}$$

Note that as with the planar case, a redundant configuration (i.e., more than three sensors) can be used in conjunction with the estimation process to improve robustness.

3.3 Features of Legged Locomotion

Legged locomotion has a series of characteristics that are not present in general robot or vehicle estimation. For bounding, the stride period (and hence frequency) is coupled to the rate and distance traveled [12]. Using this characteristic places a bound on the estimated motion of the body center (Point B in Figure 2) that is a function of the stride frequency.

Further, many running gaits can be considered to be symmetric [12]. This arises from the requirement that in order for the robot to maintain running at a fixed forward speed the horizontal forces and torques should integrate to zero and the vertical forces should integrate to the product of the weight times the stride duration [12]. Thus, for the cases where the unmodeled losses during motion are low, the motion is symmetric and an additional estimation criterion is available.

A third characteristic of running gaits is the existence of flight phases. During these phases the robot exhibits ballistic motion under the influence of gravitational acceleration. The flight phase, which begins after lift-off and ends on ground contact, can be characterized by its top-of-flight (TOF) times. The TOF is useful in delineating strides and aids control and coordination as when TOF is sensed, a cascade of actions can be initiated to prepare the leg for the impending ground contact and subsequent bound. This has been determined from inertial data using bandpass filters to remove noise and to find the time corresponding to the inflection of the vertical position [7]. For the KOLT robot, the flight phases range from 400 to 600 milliseconds.

4 Estimation Process

The process of integrating the sensed data and state information to estimate attitude is calculated using low pass filters (implemented in hardware) followed by a Kalman filter routine.

4.1 Sensor Models

Micromachined inertial sensors, like those used on the KOLT robot, display nonlinear characteristics unique to this scale of sensor [13,14]. The nonlinear aspects of the sensor include a predominance of $1/f$ (or flicker) noise, bias, hysteresis, scale factor, and internal misalignment [13]. Reference [14] presents an empirically derived model for the hysteresis, scale factor, and misalignment present in a ADXL-203 accelerometer, which is similar to the ADXL-210 accelerometer used in the KOLT IMU. The KOLT extends the models to include a polynomial model based on a full-range dynamic calibration of the inertial sensors.

The basic linear approximation of the device is shown in Equation 10. This model is extended in Equation 11 to correct for the quadratic effects presnet. The misalignment of the axis (γ) was determined by checking the response at rest state when the only acceleration present is gravity. The IR range sensors are is similarly modeled using an empirical fourth-order polynomial fit to calibration data.

$$Acceleration = \frac{(V_o - Bias)}{Sensitivity} \tag{10}$$

$$A_i\left(V_i\right) = H_{1i}V_i^2 + H_{0i}V_i + Bias_i \quad [i \in (x, y)] \tag{11}$$

where: H_{1i} and H_{0i} are obtained empirically through calibration

The acceleration values (A_i), now in units of m/sec^2, are corrected for misalignment to give the reported acceleration measurement.

4.2 Extensions to Kalman Filter Framework

The Kalman filter is a linear optimal estimation technique that has been widely applied to this class of robotics problems. The EKF allows for nonlinear state transitions by providing a means (such as Taylor series approximations) to provide linearization [15]. The work adds to this technique by adding the characteristic effects associated with legged locomotion.

Assuming a normally distrusted error with zero mean and known system and measurement covariances (Q_t and R_t), the standard Extended Kalman filter equations are [6]:

$$x_t = g\left(u_t, x_{t-1}\right) + N\left(0, R_t\right) \tag{12}$$

$$z_t = h\left(x_t\right) + N\left(0, Q_t\right) \tag{13}$$

The stride period can be determined as the period between the TOF events, which can be approximated from the inertial data. Based on observation of fairly periodic locomotion, the estimator can bias data from previous strides through the addition of a deterministic input function (s) that is a function of the previous period.

$$x_t = g\left(u_t, x_{t-1}\right) + s\left(x_{t-1-\Delta TOF}\right) + N\left(0, R_t\right) \tag{14}$$

where: s represents the weighting of data from the previous period.

In addition to a a symmetric operation, the ballistic running model also suggest that the vertical velocity $(\dot{y})$ at TOF must be zero.

The measurement covariance for the range sensor measurement update is modified to include the variation in sensor performance as a function of distance. That is, the covariance (or measurement error) throughout the range of distances measured. Thus, $Q_t = f(x_{t-1})$. For the range sensor, this was based on the empirically model and implemented as a lookup table.

Thus, the modified EKF estimation algorithm for range measurement updates (z) can be expressed as:

$$\begin{aligned}
&G_t = g'(u_t, x_{t-1}),\ H_t = h'(x_t)\\
&M_t = G_t \hat{x}_t G_t^T + Q_t \rightarrow K_t = M_t H_t^T \left(H M_t H^T + R_t\right)^{-1} \rightarrow P_t = (I - K_t H_t)\\
&\hat{x}_t = G_t \hat{x}_{t-1} + K_t \left(z_t - H_t G_t \hat{x}_{t-1}\right)\\
&\hat{x}_{t=TOF}(\dot{y}) = 0
\end{aligned}$$

5 Experiments and Results

This procedure was evaluated via experimental operations on a single leg connected to an instrumented, six-foot long boom arm with two degrees of freedom (roll and yaw). For these experiments, data from the precision encoders on the boom arm, although not without error, are considered to be the actual or control values. The attitude estimates computed on the robot leg are transformed to the boom arm origin for comparison.

Due to the constraints imposed by the boom arm, the linear dynamic system used by the Kalman filter was modified to include the relations present between the boom arm origin and the leg position (at the end of the arm). The state vector is $\vec{x} = \left[x\ \dot{x}\ x_{bias}\ y\ \dot{y}\ y_{bias}\right]$, where x,y are the position estimates, $\dot{x}$, $\dot{y}$ are the velocity estimates, and x_{bias}, y_{bias} are the accelerometer biases. The control vector is $\vec{u} = [a_x, a_y]$, where a_x, a_y are the measured accelerations.

In order to gauge the effectiveness of incorporating specific variations made to the EKF for legged locomotion, the data was also processed using a standard EKF formulation was identical, except it did not include data from the stride period.

5.1 Boom Attitude Estimation Experiments

Experiments were performed at speeds ranging from 0 to $3.5\ m/s$ with an average of 100 hops per experimental run. The amount of roll and yaw was estimated using the modified EKF estimation method. As there were no measurement updates along the yaw axis, the estimator had no direct means of performing drift correction. As such, the bias estimate determined for the roll (y-axis) was used instead.

The results for this method are promising and show improvement over Kalman filtered estimates, but remain colored by significant errors. As shown in Figures 3 and 4, the extended method described has less error compared to the control ($1.16°$ rms error) than the standard EKF routine ($1.66°$ rms error). Qualitatively the proposed method is better able to estimate the full range of motions

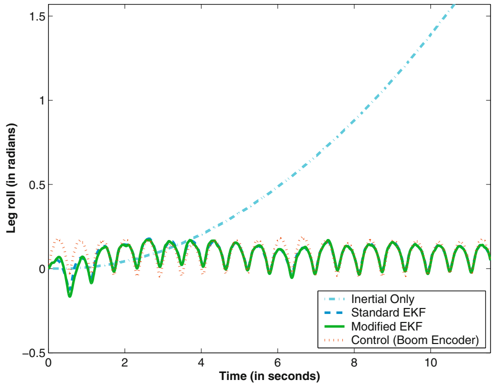


Fig. 3. Roll angle estimates by various approaches. The inertial error, even when filtered, drifts rapidly and is greater than 90° within 10 seconds.

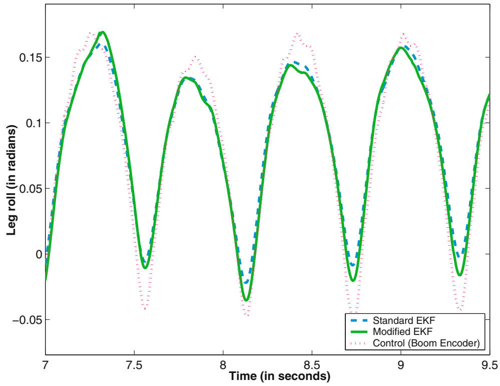


Fig. 4. Roll angle estimates for a section of the experimental run showing the slight differences between modified and standard EKF estimators. The plot shows a section consisting of four bounding periods and the attitude estimates of the two methods compared to the control value calculated from high-resolution encoders attached to the boom arm.

5.2 Top Of Flight Estimation

A second parameter that characterizes legged locomotion is the TOF. For proper control of the robot's motions, it is important to accurately determine its occurrence so as to correctly set the robot for the upcoming bound. The TOF is calculated by finding the point when velocity goes from positive to negative, suggesting an inflection of the position. For inertial data, this is found determined using an infinite impulse response (IIR) Chebyshev bandpass filter, whose order (fourth) was selected to be sufficiently high so as to remove the noise and artifacts, yet low enough so as not to add excessive amounts of delay.

Figure 5 shows the TOF estimate based on the velocity state computed by the modified EKF technique. In particular, the RMS errors for TOF (with respect to the control case) for the standard EKF, modified EKF, and IIR bandpass filter routines are 57, 53, and 119 milliseconds respectively. With respect to the short (approximately 500 milliseconds) stride periods for running on KOLT, the delay of modified EKF represents 10% of the stride as compared to 24% of the stride using purly inertial technique.

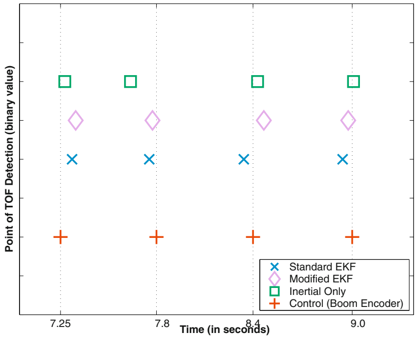


Fig. 5. Top-of-Flight estimates for a section of the experimental run. The grid lines are aligned with the TOF times determined from the control data. The markers show the TOF times calculated from the standard EKF, modified EKF, inertial only datasets

6 Conclusions

Dynamic legged locomotion is a unique domain separate from aerial or wheeled locomotion. The classic solution to robot position and attitude estimation in this domain has been to use a general measurement solution. We have proposed and shown that modifying EKF estimation techniques to include characteristics unique to legged robotics results in improved attitude tracking and reduced delay.

These estimation techniques will aid in the rapid and accurate on-board estimation of attitude necessary for dynamic motion, such as galloping. The experiments performed show the use of updates from optical range measurements to be a promising means of addressing the classic quadratic drift present in inertial sensors. The infrared ranger sensors used in these preliminary experiments are very limited in applicability beyond the laboratory due to their slow performance, variance with ground reflective properties. Even with an ideal sensor, the ground based range techniques described assume a level ground plane, which thereby precludes field operations.

While it is possible to replace the infrared range sensors with laser-based range estimators, this would still be subject to geometric uncertainties and interference. Further, such an approach is still limited to planar terrains and does not provide a means to correct yaw. Ongoing experimental efforts are looking at visual odometry (and related vision routines) to determine this motion and perform the necessary updates to enable field-robust high-fidelity attitude estimations.

7 Acknowledgments

This research and paper are supported, in part, through a National Science Foundation Grant (No. IIS-0208664) and of a National Defense Science and Engineering Graduate (NDSEG) program fellowship. The authors thank Jonathan Clark for comments on modeling legged locomotion. The authors acknowledge the numerous contributions of the KOLT team members including J. Gordon Nichol and Louis Burgos at Stanford University and Prof. David Orin and Luther Palmer at The Ohio State University.

References

1. Sony. *QRIO Technical Specification Brochure*. Sony Corporation, 2003.
2. J. G. Cham, S. A. Bailey, J. E. Clark, R. J. Full, and M. R. Cutkosky. Fast and robust: Hexapedal robots via shape deposition manufacturing. *International Journal of Robotics Research*, 21(10), 2002.
3. Bernhard Klaassen, Ralf Linnemann, Dirk Spenneberg, and Frank Kirchner. Biomimetic walking robot scorpion: Control and modeling. In *Proceedings of the ASME Design Engineering Technical Conference*, volume 5, pages 1105–1112, 2002.
4. Sanjiv Singh, George Kantor, and Dennis Strelow. Recent results to simultaneous localization and mapping. In *International Symposium on Experimental Robotics*, 2002.
5. Jay Farrell and Matthew Barth. *The Global Positioning System and Inertial Navigation*. New York, 1999.
6. Technical Staff of the Analytic Sciences Corporation. *Applied Optimal Estimation*. MIT Press, Cambridge, 1974.
7. J. Gordon Nichol, Luther R. Palmer III, Surya P.N. Singh, David E. Orin, and Kenneth J. Waldron. System design of a quadrupedal galloping machine. *International Journal of Robotics Research (to appear)*, 2004.
8. R. McNeil Alexander. *Animal Mechanics*. University of Washington Press,, Seattle, 1969.
9. Dave McMordie. *Towards Pronking with a Hexapod Robot*. Masters, McGill University, 2002.
10. G. Kantor and A. Rizzi. Feedback control of underactuated systems via sequential composition: Visually guided control of unicycle. In *International Symposium on Robotics Research (ISRR)*, 2003.
11. Sharp Corporation. Gp2y0s02yk data sheet, 2001.
12. Marc H. Raibert. *Legged robots that balance*. MIT Press series in artificial intelligence. MIT Press, Cambridge, 1986.
13. Marc J. Madou. *Fundamentals of microfabrication : the science of miniaturization*. CRC Press, Boca Raton, FL, 2nd edition, 2002.
14. W.T. Ang, S.Y. Khoo, P. K. Khosla, and C. N. Riviere. Phyiscal model of a mems accelerometer for low-g motion tracking applications. In *Proceedings of the Interenational Conference on Robotics and Automation 2004*, pages 1345–1351., 2004.
15. H. Rehbinder and Xiaoming Hu. Drift-free attitude estimation for accelerated rigid bodies. In *Proceedings of the International Conference on Robotics and Automation 2001*, volume 4, pages 4244–4249, 2001.

Toward Reliable Off Road Autonomous Vehicles Operating in Challenging Environments

Alonzo Kelly, Omead Amidi, Mike Bode, Mike Happold, Herman Herman, Tom Pilarski, Pete Rander, Anthony Stentz, Nick Vallidis, Randy Warner

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
alonzo@rec.ri.cmu.edu, etc.

Abstract. The DARPA PerceptOR program implements a rigorous evaluative test program which fosters the development of field relevant outdoor mobile robots. Autonomous ground vehicles are deployed on diverse test courses throughout the USA and quantitatively evaluated on such factors as autonomy level, waypoint acquisition, failure rate, speed, and communications bandwidth. Our efforts over the three year program have produced new approaches in planning, perception, localization, and control which have been driven by the quest for reliable operation in challenging environments. This paper focuses on some of the most unique aspects of the systems developed by the CMU PerceptOR team and the most immediate challenges that remain to be addressed.

1 Introduction

The potential applications of robotic ground vehicles have long been recognized. Off road mobile robots must operate effectively under forest canopy which occludes positioning satellite signals while the trees themselves present natural mazes to challenge motion planning. They must function in alpine areas where terrain slopes require safe operation in constant proximity to tipover, while precipitous ledges persistently threaten to end the mission in an instant. They must function in fields and forests where ground covering vegetation obscures both the shape of the ground and any occluded hazards. In short, off road autonomy is among the most ambitious of our aspirations for mobile robot technology.

Much of the work to date has been motivated by military [2][8] and space [1], [10], applications although agriculture [5], mining [4], and forestry have received more recent attention. The system described in this paper has evolved from local [6] and global [9], planning systems that we developed for the Demo II program.

Unlike most or perhaps all historical work on outdoor autonomous mobility, the PerceptOR program emphasizes independently administered evaluative testing as the primary mechanism to drive progress. The overall goal has been to simultaneously maximize autonomy, reliability, and speed.

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 599–608, 2006.

Tests are conducted on an unrehearsed basis, meaning the development team has no detailed knowledge of specific terrain before the test. While the development team may see the test courses during the conduct of the tests, individuals who operate the system are prevented from seeing the test courses until all tests are complete. The overall intent is to simulate the conditions of actual deployment of UGVs. This paper outlines the final design of the system produced by the CMU PerceptOR team, results achieved, and some of the most immediate challenges that remain to be addressed

2 System Design

The UGV hardware design is based on the Honda Rubicon All Terrain Vehicle (ATV). Elements for man-aboard driving were removed entirely and replaced with an autonomy retrofit. SICK ladars were custom modified to scan in a second axis to convert them to 2D imaging devices. Up to four of these devices have been used per vehicle at various times to generate forward and rearward perception for obstacle avoidance and omnidirectional views for operator interface purposes. Two stereo pairs developed by Sarnoff Corporation provide passive forward range imaging. Several monochrome teleop cameras are provided. Monocular digital color and analog FLIR cameras are used for estimating terrain composition. A Smiths land inertial navigation system and Novatel GPS receiver provide global position estimates.

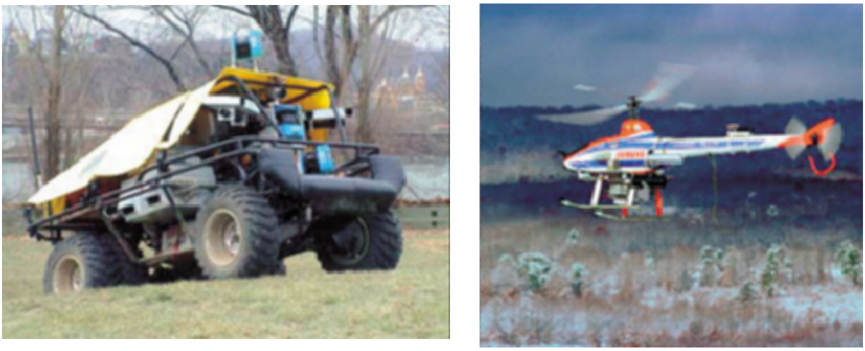


Fig 1: Autonomous Ground and Air Vehicles. A Honda ATV and Yamaha Rmax Helicopter were retrofitted for autonomy.

Over a three year period, the objectives and our experiences in field tests have driven us to implement new approaches at all levels of the traditional autonomy software hierarchy – from gross motion planning to reactive low level control. Some of the new elements are discussed below.

3 Position Estimation and Mapping

Various disappointments in field experiments have driven us to redesign the position estimation and mapping approach in order to respond to the challenges of generating high fidelity perception on a moving platform. While some of the following techniques have been used in earlier work, we take them to new extremes and organize the principles in this work.

3.1 Accumulation – Distortion Tradeoff

Nonideal pose estimates cause distortions in environmental models that are created with them. Pose error accumulation rate often increases with motion, motion is more difficult to measure on rough terrain, and sensitivity to these errors increases as the desired fidelity of perception increases.

There may at times be a fundamental requirement to accumulate data in a region; for example, in order to compute a region property the size of a wheel or of a vehicle. Nevertheless, the value of excess data accumulation (beyond the fundamental requirements) is at odds with the cost of the cumulative effects of pose error. Ironically, despite good intentions, too much oversampling eventually incorporates enough distortion to make it impractical to reliably resolve the features of interest at the scale of interest. Several design principles have emerged to manage this tradeoff. The impact of distortion can be minimized by …

- **Minimum Accumulation** … accumulating no more data than necessary. Hence, when obstacle signatures can be computed from one scan line of ladar data they should be. When a few lines are necessary, then only a few lines should be used.
- **Exploit Signal Properties** … exploiting the best properties of pose estimate. It is better to compute region properties within ladar scan lines than across them. If two sets of consecutive lines must be accumulated, then the computation is organized to compute high fidelity local results in each set first, and then merge them in a manner consistent with the larger error accumulated between the sets.
- **Engineer the Distortion Signal**. … providing the best possible pose signal(s). Ladar(s) can be oriented to extract preferred information from the faster of the two rotation axes. Multiple custom-designed pose estimates can easily be generated for multiple purposes.

These principles drive our approach to position estimation and mapping. A hierarchical arrangement of data structures provides decreasing periods of data accumulation and increasing levels of detail as the hierarchy is descended. An associated hierarchy of position estimates trade absolute accuracy for relative as the hierarchy is descended. Data is accumulated with one pose estimate but localized with the next highest estimate and, if necessary, re-localized to track the growing mis-registration between the two.

3.2 Multiple Pose Estimates and Maps

At the global level, a global pose estimate is computed from all available sensing several times a second. This estimate is used to control motion with respect to globally specified waypoints, and to fuse onboard and externally provided geo-referenced data.

At the local level, a local pose estimate is computed based only on sensing which does not project onto position states. It does not process GPS readings, nor would it process terrain aids like absolute landmarks if they were available. This estimate drifts with respect to the global estimate but it is completely immune to the characteristic jumps in GPS position which occur before loss of lock on satellite(s) and at their reacquisition. This estimate is used to provide feedback to motion control and to estimate motion relative to obstacles for obstacle avoidance purposes. Neither of these uses requires an absolute sense of position on the earth.

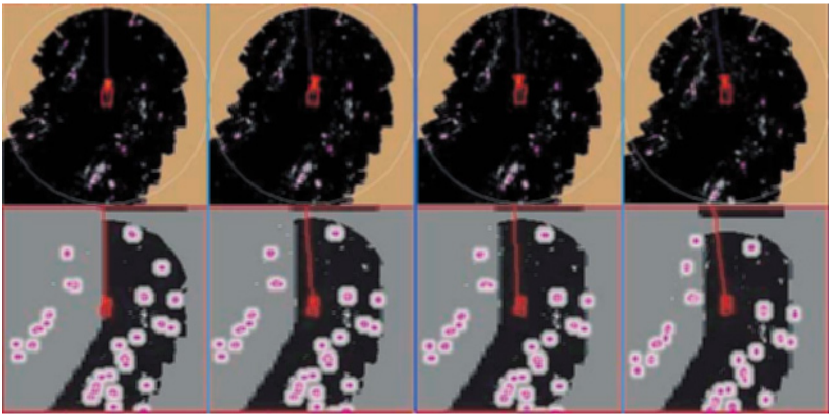


Fig 2: Dual Maps and Position Estimates. Overhead renderings of the local map (top) and corresponding global map (bottom) are shown at four closely spaced periods of time. Over this time, the global estimate moves the vehicle to the right by one vehicle width due to GPS drift. In the global map, the dark roadway is geo-registered and does not move. The obstacles, however, were located by perception and placed in the local map so they shift right with the vehicle. In this way, obstacle avoidance becomes immune to GPS jumps.

At the perceptual level, we have not yet invested in a different estimate. However, the assessment of local shape does not require a sense of absolute attitude or heading and errors in these quantities need not be passed to shape-level processing. An estimate based only on integrating gyros might be valuable.

Fig 2 illustrates the operation of the two highest levels of mapping during a GPS jump. The local map is re-registered to the global map at high rates so that while the relative position (local-level shape) of perception-derived data remains stable, its global position is adjusted continuously.

4 Perception

In addition to the aspects of perception which are implied above, a few other design approaches have emerged in order to address the challenges of vegetated and forested terrain.

4.1 3D Volumetric Density Mapping

While, in many earlier works, systems have been able to function using an overhead planar projection of the robot and the environment, and while we continue to do so at the global level, the overhead projection of the local map is derived from a more detailed 3D representation (**Fig 3**). Like the planar local map, this data structure is implemented as a circular queue which wraps around in 3 dimensions in order to avoid the need to move the data in memory as the vehicle moves over distances much larger than the size of the map. Both the 2D and 3D local maps have coordinates aligned with and orthogonal to gravity to facilitate attitude predictions.

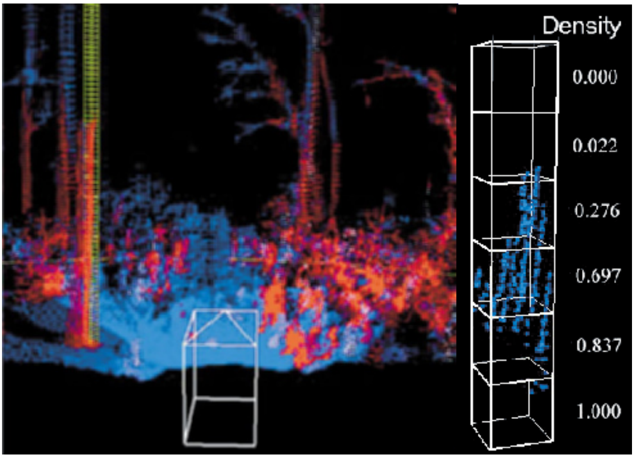


Fig 3: 3D Perception. Left: The colorized point cloud to the left indicates the typical 3D structure of a forested scene on a trail. Red regions are nontraversable; blue ones are traversable. Right: Density scores in a vertical column through the point cloud.

In this data structure, the entire volume swept by the vehicle during candidate motions can be tested for collisions. While the height of the column of data intersected by the vehicle does not vary, its vertical position in the 3D map does vary. Nonetheless, given the vertical position of the column, it is possible to identify the relevant data and collapse its net assessment into a 2D cell in the planar representation.

The basic attribute extracted in each 3D cell is the “density” computed as the ratio of the number of hits to hits plus misses. A cell gets a hit if the ladar beam terminates inside it and a miss if it was passed through. This calculation requires that each cell in the line from the sensor to the returning cell be identified in a ray tracing process.

4.2 Learning to Estimate the Ground Plane

Of course, the problem of identifying the ground plane, or more precisely the shape of the load bearing surface, is not straightforward in environments with vegetative ground cover. In meadows and under forest canopy, we have found it necessary to estimate the supporting surface based on the density scores.

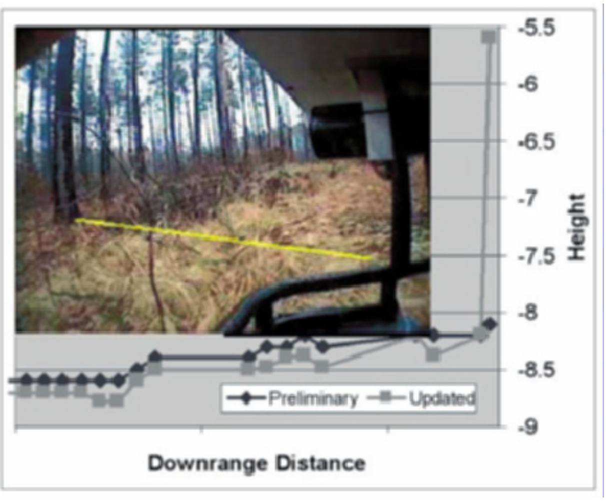


Fig 4: Computing the Load Bearing Surface Under Ground Cover. The yellow line in the inset image symbolically represents the slice through the world model density scores which is being processed. The neural net lowers the grass and raises the tree which is the correct adjustment of the load bearing surface.

Based on the learning principle originally developed for agricultural robots in [11] we developed a neural network which learned how to adjust a preliminary estimate of the load bearing surface elevation based on observing the ground truth data generated when the vehicle drives over terrain in the learning phase. **Fig 4** shows the operation of the neural net in a typical scene.

5 Motion Planning

In order to manage computational complexity, the planning system is also arranged hierarchically. It is composed of the Dstar global planner [9] and the Ranger local planner [6]. These elements were integrated for the first time on

PerceptOR. The basic principle used is the assumption that high fidelity models of vehicle dynamics and maneuverability are less important farther from the vehicle than near it (**Fig 5**.)

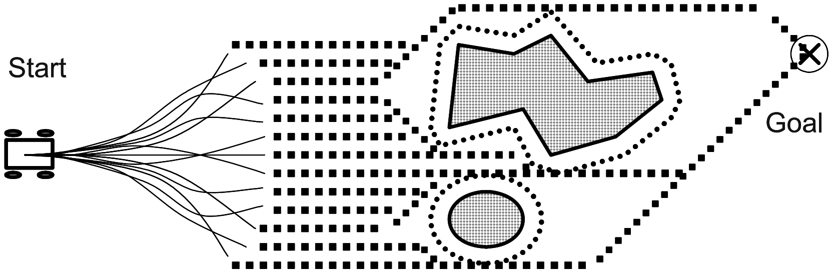


Fig 5: Hierarchical Motion Planner. Candidate trajectories are produced by forward simulation of the vehicle dynamics out to some distance after which the optimal remaining path in an 8-connected grid with expanded obstacles is used. An elevation grid is used in the high fidelity portion of the motion simulation and the total line integral over a cost field of both portions of each optimal path is used to select the option to execute.

Paths between waypoints of hundreds of meters separation are generated autonomously. Routinely, the system comes to a stop while following the best available plan because there are apparently no safe options for proceeding. A set of behaviors are then used to verify the assumption of no safe forward path and implement various recovery mechanisms.

First, a slow speed perception scan to regenerate the area in view is performed while the vehicle is stationary. If this mechanism does not generate a way forward, the system next executes a very effective reactive backup maneuver attempting to align the vehicle with the direction preferred by global planning. Dense surrounding obstacles will rarely cause this step to fail. A nonholonomic motion planner was developed to address these cases. If it fails, the system asks the operator for help.

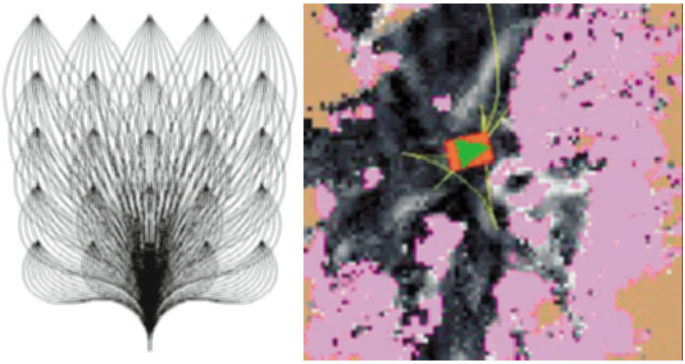


Fig 6: Nonholonomic Motion Planner. Left: In this zero heading slice (restricted to forward motions for display), steering functions to a regular array of neighboring poses are encoded. Right: A custom 5 point turn is generated to turn after detecting a natural cul-de-sac

This planner is based on a generalization of a grid which we call a pose lattice. The space of robot poses (x,y,heading) is discretized into a dense 3D grid and the trajectory generator described in [7] is used to connect every node to every reachable neighboring node out to some practical radius. The pose lattice structure can then be interpreted as a graph for purposes of implementing heuristic search (A* is implemented at the moment). The structure itself can be represented implicitly in terms of a repeated template. The result is a regular sampling of state space which encodes, to finite resolution, all possible motions between neighboring states using only feasible motions. Complex motions respecting the constraints of local obstacles are then generated by optimal network search.

6 Project Results

Test results on the PerceptOR program are tallied at the system level in order to focus effort on overall performance. Among the data collected is the number of times field safety personnel intervened in order to prevent vehicle damage, the communications bandwidth used, system ability to reach desired waypoints, speed, distance, and time. The number of operators was reduced to one near the middle of the program and at times, tests have been performed to evaluate the viability of restricting operator intervention to those cases when the robot asks for help.

Test courses were chosen by DARPA at each of four army bases in the north east, south central, south west and north west of the United States. Systems were thereby exposed to forested, desert, and mountainous terrains under varying weather conditions in different seasons. Exercises, consisting of several partially repetitive runs per day, were normally conducted over a one week period at each site. Two sites were visited twice in order to assess improvement over time. Approximate cumulative data for all six exercises are summarized below in Table 1 A rough sense of improvement over time can be obtained by comparing the cumulative scores for all 6 exercises with those of the final one. One of the tests of the fifth exercise is illustrated in detail in **Fig 7**.

Table 1: Cumulative and Final Data for PerceptOR Program Exercises (CMU Team Only)

Item	Unit	Cumulative	Last Test
Number of Runs		183	36
Distance Traveled	m	81,094	29,376
Test Duration	s	230,606	49,172
Average Speed	m/s	0.35	0.60
Uplink Distance Density	Kb/m	34.5	4.8
Downlink Distance Density	Kb/m	705	327

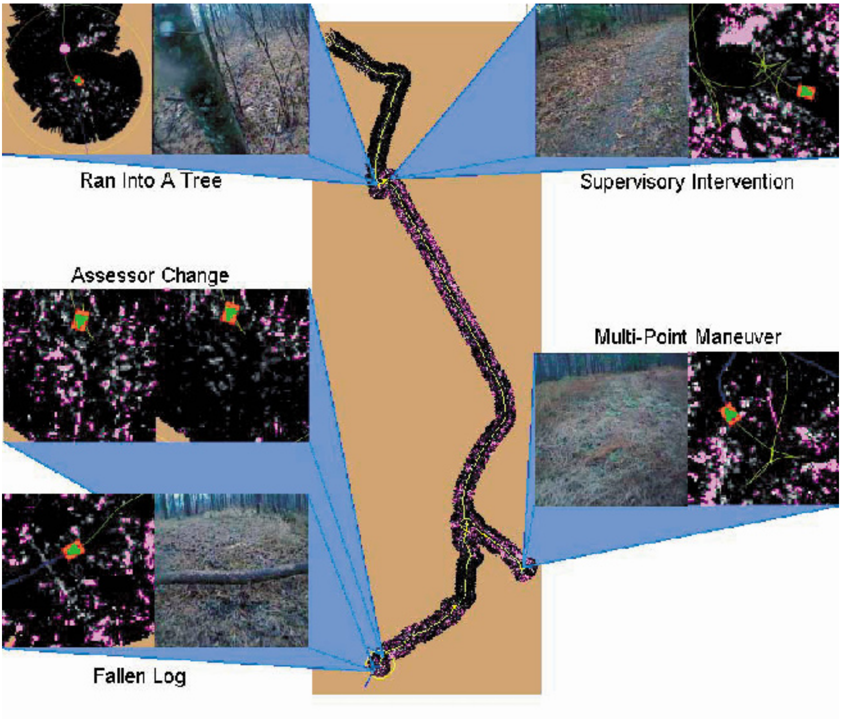


Fig 7: Test Run with Forest, Meadow, and Trail. After driving perhaps 50 meters through a forest, the system collides harmlessly with a small tree. The remote operator takes control to clear the error and move the vehicle away from the hazard. The system acquires a trail and follows it autonomously for a long distance before determining that the trail is no longer heading toward the goal. It then executes a multi point turn maneuver, retraces its path, and exists the trail at an appropriate place. The operator reconfigures the software to tolerate tall grass (assessor change) near the end of the run and it terminates at a fallen log visible as the bright streak in the lower left cost image.

7 Conclusions and Outlook

The PerceptOR program has been unprecedented in its rigorous approach to evaluative testing of unmanned ground vehicles. System performance has clearly improved over time in response to test results, and many opportunities for more performance improvements obviously remain.

Among those improvements are immediate issues in perception and planning. We remain dissatisfied with the level of adaptability of perception algorithms to varying terrain. Less manual tuning of algorithms could potentially be achieved by creating a higher level understanding of overall context (forest, trail) in the system. Issues relating to the interplay between perception and planning are becoming more significant. The advantages of real time replanning come at the cost of a system which can change the global plan too radically and too often. A lack of aggregate understanding of the local situation can lead to futile struggles to forge a way through a stand of trees immediately beside a perfectly good trail.

Some of the grand challenges of outdoor autonomy are well understood. A few are detecting small obstacles at high speeds, detecting negative ones at any speed, passive operations and night, and perceiving the groundplane and hazards hidden beneath obscuring vegetation. In addition to some progress on some of these fronts, the main contribution of the PerceptOR program has been the most rigorous attempt to date to quantify the present state of the art.

C/N 2497 DARPA Approved. Unlimited Distribution

8 References

[1] G. Andrade, F. Amara, P. Bidaud and R. Chatila, “Modelling of Robot-Soil Interaction for Planetary Rover Motion Control”, Proc. IEEE/RSJ ICIRS, Victoria, Canada, 1998.

[2] S Balarski and A. Lacaze, “World Modelling and Behavior Generation for Autonomous Robot Ground Vehicle”, Proc IEEE ICRA, San Francisco, USA, April 2000.

[3] P. Belluta, R. Manduchi, L. Matthies, K. Owens, and A. Rankin, “Terrain Perception for Demo III”. Proceedings of IEEE Intelligent Vehicles Symposium, pages 326-331, October 2000.

[4] J. Roberts, P. Corke, and G. Winstanley, “Development of a 3,500 tonne field robot”, *The International Journal of Robotics Research*, 18(7):739-752, July 1999

[5] H. Hagras, M. Colley, V. Callaghan, and M. Carr-West, “Online Learning and Adaptation of Autonomous Mobile Robots for Sustainable Agriculture”, Auton. Robots 13(1): 37-52 (2002).

[6] A. Kelly, and A. Stentz, “Rough Terrain Autonomous Mobility – Part 2: An Active Vision, Predictive Control Approach”, Autonomous Robots, 5, 163-198, 1998.

[7] A. Kelly, and B. Nagy, “Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control”, International Journal of Robotics Research, July 2003.

[8] A. Lacaze, Karl Murphy, M. Delgiorno, “Autonomous Mobility for the Demo III Experimental Unmanned Vehicles”, AUVSI 2002.

[9] A. Stentz, “Optimal and Efficient Path Planning For Partially-Known Environments”, Proc IEEE ICRA, May 1994.

[10] P. Schenker, T. Hunstberger, P. Pirjanian, S. Dubowski, K. Iagnemma, and V. Sujan, “Rovers for Agile Intelligent Traverse of Challenging Terrain *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS*, Nara, Japan, May 2003.

[11] C. Wellington and A. Stentz, “Learning Predictions of the Load-Bearing Surface for Autonomous Rough-Terrain Navigation in Vegetation.” International Conference on Field and Service Robots, July 2003, pp 49-54.

Driving on a Known Sensori-Motor Trajectory with a Car-like Robot

Cédric Pradalier, Pierre Bessière, and Christian Laugier

INP Grenoble – GRAVIR Laboratory – INRIA Rhône-Alpes
Unité de Recherche INRIA Rhône-Alpes :
Zirst - 655 avenue de l'Europe - Montbonnot
38334 Saint Ismier Cedex - France
Tel : +33 (0)4 76 61 53 45 - Fax : +33 (0)4 76 61 52 10

Abstract. Autonomous navigation of a mobile robot along a predefined trajectory is a widely studied problem in the robotics community. We propose a Bayesian architecture that aims at being able to replay any sensori-motor trajectory – trajectory defined as a sequence of perceptions and actions – as long as the robot starts in its neighbourhood. In order to increase robustness, we also use this Bayesian framework to build our controller and an obstacle avoidance module. This work has been validated both on a simulated robot and on a real robot: the CyCab.

1 Introduction

Autonomous navigation of a mobile robot has been a widely studied problem in the robotic community. Most robot designed for this task are equipped with on-board sensor(s) to perceive external world (sonars, laser telemeters, camera). Then, two main kinds of approach to autonomous navigation have been proposed: reactive navigation where the robot uses only current perception to move and explore without colliding ([1,3]) and servo-ed navigation where the robot is given a pre-planned reference trajectory and use some closed-loop control law to follow it [13,12]. Among servo-ed navigations, two classes of approaches can again be separated: state space tracking[8,7] and perception space tracking[15,4].

State space tracking implies two specificities: first to be given a reference trajectory in the state space, and second, to be able to localise the robot, also in the state space. Conversely, perception space tracking implies that trajectory is defined with respect to perception only, hence avoiding the need for global localisation. A specific application of perception tracking is visual servoing, classically implemented as the convergence of observed image to a fixed reference image.

In this paper, we are specifically interested in the case of perceptual tracking of a perceptual trajectory with a mobile robot. We assume that: i) reference trajectory is defined as a sequence of observations perceived by an on-board sensor along robot movement; ii) no localisation system (neither GPS nor landmark based) is available to perform tracking. This situation is interesting for at least three reasons: first, since trajectory is not defined with respect to a Cartesian frame we don't need to deal with the complex task of global localisation, second this kind of trajectory

M.H. Ang and O. Khatib (Eds.): Experimental Robotics IX, STAR 21, pp. 609–618, 2006.

can be naturally and easily learned from examples, and third, it can be seen has an hypothesis on how biologic entities memorise and represent paths.

This paper will be organised as follows: after some preliminary definitions and problem specification in section 2 and 3, we deal separately with initialisation, state estimation and trajectory tracking in section 4 to 6. Finally section 7 presents implementation details and results.

2 Definitions and hypothesis

Let us start by introducing some useful notations: our robot is described by a configuration C and commanded with a command U, our sensor gives observations O and a difference of configuration is noted ξ. $\mathcal{O}(t)$ is the set of past observations at time t: $\{O(u) \mid u \leq t\}$. Finally, previous instant $t - \Delta t$ with respect to instant t will be noted t'.

We assume to be given a sensor model H and a kinematic model K. H predicts observation $\tilde{O}$ given a reference observation O_{ref} and a difference of view point ξ: $\tilde{O} = H(O_{ref}, \xi)$. K is defined with $C(t) = K(C(t'), U(t'), \Delta t)$.

Finally, we write $\mathcal{G}(\mu, \sigma)$ a Gaussian distribution centred on μ with covariance σ.

3 Problem specification

3.1 Sensory-motor trajectories

Definition 1. We define a *sensori-motor trajectory* as a function of time with values in the Cartesian product of the robot command space $\mathbb{U}$ and its observation space $\mathbb{O}$. Formally:

$$\mathcal{T}_{sm} : [0, t_1] \longrightarrow \mathbb{O} \times \mathbb{U}$$

In the following, we will note $\mathcal{T}_{sm}(t) = [\mathcal{T}_{sm}(t).O, \mathcal{T}_{sm}(t).U)]$.

3.2 Objectives

In this paper, we want to be able to replay any sensori-motor trajectory, as long as the robot starts in its neighbourhood. We divided our objective in three tasks:

- **Initialisation:** Given a sensori-motor trajectory $\mathcal{T}_{sm}$, and given an observation $O \in \mathbb{O}$, what is the robot temporal position $\tau(0)$? In other words, we want to find $\tau(0) \in [0, t_1]$ for which O is closest from $\mathcal{T}_{sm}(\tau(0))$. Note that this does not guarantee that O is an observation on $\mathcal{T}_{sm}$.
- **State tracking:** Given $\mathcal{T}_{sm}$, O, we have to keep an estimate of,first, the robot temporal position $\tau(t)$ at time t, and, second, its configuration error with respect to the sensori-motor trajectory.
- **Trajectory tracking and control:** Given $\mathcal{T}_{sm}$, O and $\tau(t)$, we want the system to compute commands to apply in order to follow the trajectory as closely as possible, while being able to introduce some deformations to avoid obstacles.

3.3 Bayesian Software Architecture

Since we believe ([14,2]) that probabilities and Bayesian inference are well adapted tools to cope with incompleteness and uncertainties inherent to the navigation of a mobile robot in a real environment, we want to achieve sensori-motor trajectory replay using a Bayesian software architecture.

Following sections will show how a set of distinct Bayesian modules can be integrated in a modular Bayesian architecture in order to fulfil robustly objectives presented in this section.

4 Initialisation

In this section, we present how system initialisation is performed. We assume that a sensori-motor trajectory $\mathcal{T}_{sm}$ is known and that an observation O_0 has been done. We will use the Bayesian programming framework to compute a probability distribution over possible temporal position in the trajectory. Fig. 1 formalises the Bayesian program for initial localisation (see [14,2] for details about this formalism).

$$
\text{Program}\left\{\begin{array}{l}
\text{Description}\left\{\begin{array}{l}
\text{Specification}\left\{\begin{array}{l}
\text{Relevant Variables:}\\
\mathcal{T}_{sm} : \text{ Known sensori-motor trajectory}\\
O_0 : \text{ Initial observation}\\
\tau_0 : \text{ Initial temporal localisation}\\
\xi_0 : \text{ Initial difference of view point}\\
\text{Decomposition: } P(O_0\ \tau_0\ \xi_0 \mid \mathcal{T}_{sm}) =\\
\quad P(\tau_0)\ P(\xi_0)\ P(O_0 \mid \tau_0\ \xi_0\ \mathcal{T}_{sm})\\
\text{Parametric Forms:}\\
P(\tau_0) : \text{Uniform at initialisation}\\
P(\xi_0) : \text{Zero centred Gaussian}\\
P(O_0 \mid \tau_0\ \xi_0\ \mathcal{T}_{sm}) : \text{See eq. 1}
\end{array}\right.\\
\text{Identification: a priori}
\end{array}\right.\\
\text{Question: } P(\tau_0 \mid O_0\ \mathcal{T}_{sm}) \propto P(O_0 \mid \tau_0\ \mathcal{T}_{sm})
\end{array}\right.
$$

Fig. 1. Bayesian program for initial localisation

4.1 Expression of $P(O_0 \mid \tau_0\ \mathcal{T}_{sm})$

$P(O_0 \mid [\tau_0 = t]\ \mathcal{T}_{sm})$ should express which observations are expected around $\mathcal{T}_{sm}(t)$. To this end, let us first define a probabilistic sensor model. Given a reference observation and a difference of viewpoint ξ, a sensor model expresses the probability of expected observation:

$$P(O_0 \mid \tau_0\ \xi_0\ \mathcal{T}_{sm}) = P_m(O_0 \mid \mathcal{T}_{sm}(\tau_0).O\ \xi_0) = \mathcal{G}(H(\mathcal{T}_{sm}(\tau_0).O, \xi), \Sigma_m) \tag{1}$$

Using this model, we can define:

$$P(O_0 \mid \tau_0\ \mathcal{T}_{sm}) = \int_{\xi_0} P(O_0 \mid \tau_0\ \xi_0\ \mathcal{T}_{sm}) P(\xi_0) d\xi_0 \tag{2}$$

To express the fact that we are interested by what is expected around the nominal trajectory, $P(\xi_0)$ is defined as zero-centred Gaussian, with a covariance formalising the "around $\mathcal{T}_{sm}$" notion.

4.2 Initialisation with $P(\tau_0 \mid O_0\ \mathcal{T}_{sm})$

Using Bayesian program shown in fig. 1, we can compute a numerical approximation of $P(\tau_0 \mid O_0\ \mathcal{T}_{sm})$, denoted as $\hat{P}$ in the following. Fig. 2 gives a typical result of this distribution.

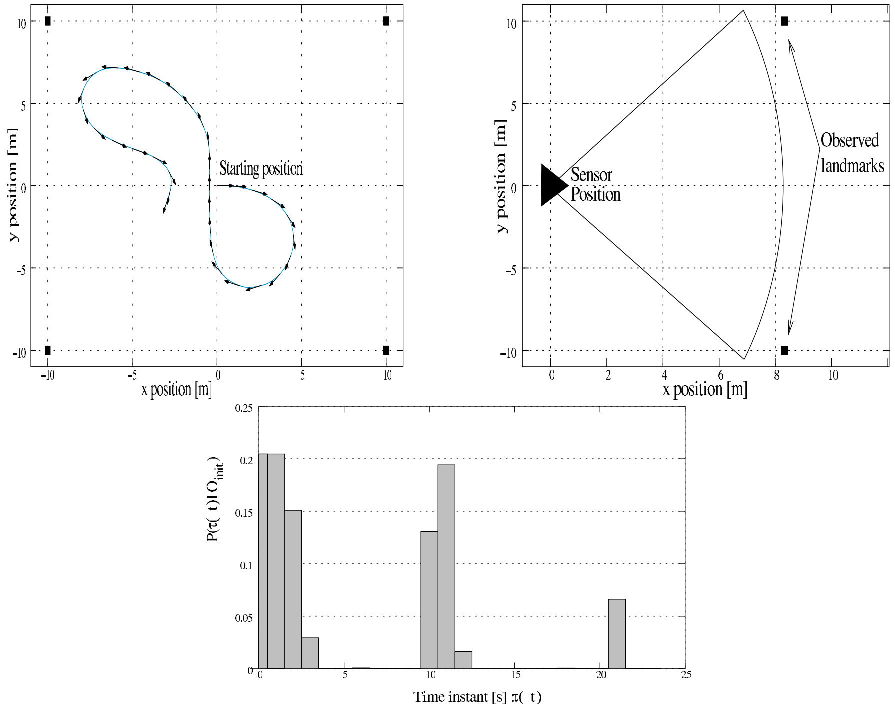


Fig. 2. Example of initial $P(T \mid O\ \mathcal{T}_{sm})$: Case of a planar robot observing landmarks (black squares) with a laser telemeter.

From this distribution, we have to extract a single value which will be our initial estimate $\tau(0)$. Depending on the shape of the distribution, the difficulty of this extraction can range from easy to hard or even impossible.

One point should be noted here. Even, when $\hat{P}$ is strongly peaked, there is no guarantee that we are indeed observing some part of $\mathcal{T}_{sm}$. Nevertheless, without other information, we believe that we have to start moving as if we were confident with respect to our first estimation, while keeping in mind that future observations may infirm this estimation. Further details on this point can be found in [16].

5 State estimation

Once we know where the system start in the *sensori-motor* trajectory, we can start replaying it. Two variables have to be tracked: temporal position and the error with respect to the nominal trajectory. From these variables, robot controls can be computed to track the trajectory.

5.1 Error tracking

Using current estimate τ_t of the temporal position, the sensor model defined in 4.1 ($P_m(O \mid O_{ref}\xi)$) and Bayes rule, we can infer a distribution over difference of viewpoint ξ_t given current observation. To track this variable, we want to use a Bayesian filter:

$$P(\xi_t \mid \mathcal{O}_t\ \mathcal{T}_{sm}(\tau_t).O) \propto P_m(O_t \mid \xi_t\ \mathcal{T}_{sm}(\tau_t).O) \times \int_{\xi_{t'}} P(\xi_t \mid \xi_{t'}\ \mathcal{T}_{sm}(t').U) P(\xi_{t'} \mid \mathcal{O}_{t'}) \tag{3}$$

To this end, we define $P(\xi_t \mid \xi_{t'}\ \mathcal{T}_{sm})$ using the kinematic model K of our robot. Since we are only interested in the relative change of ξ, we can express its variation as follows:

$$\hat{\xi}_t = K(\xi_{t'}, U(t'), \Delta t) - K(0, U_{ref}(t'), \Delta t) \tag{4}$$

And finally, $P(\xi_t \mid \xi_{t'}\ \mathcal{T}_{sm})$ will be defined as a Gaussian, centred on $\hat{\xi}_t$, with a covariance matrix expressing our confidence on this model.

5.2 Temporal position tracking

Estimation of temporal position can be seen as another localisation process. Again we are tempted to use a Bayesian filter for this estimation:

$$P(\tau_t \mid \mathcal{O}_t\ \mathcal{T}_{sm}) \propto P(O_t \mid \tau_t\ \mathcal{T}_{sm}) \times \int_{\tau_{t'}} P(\tau_t \mid \tau_{t'}) P(\tau_{t'} \mid \mathcal{O}_{t'}) \tag{5}$$

where eq. 2 defines $P(O_t \mid \tau_t\ \mathcal{T}_{sm})$, and transition model $P(\tau_t \mid \tau_{t'})$ is defined as $P(\tau_t \mid \tau_{t'}\ \Delta t) = \mathcal{G}(\tau_{t'} + \Delta t, \sigma_\tau)$. This method works, but its computational complexity is prohibitive: we have to integrate P_m over the configuration space for every value of τ.

Heuristic: Due to the computational cost of direct application of the Bayesian filter, we chose to use a heuristic to keep track of temporal position: we assume that executed trajectory keep pace with respect to reference trajectory (i.e. $\tau_t = \tau_{t'} + \Delta t$) while $|\xi_t| \leq \xi_{\text{lim}}$. When error ξ_t goes over a threshold, we keep τ constant (i.e. $\tau_t = \tau_{t'}$).

6 Trajectory tracking and obstacle avoidance

6.1 Trajectory tracking

Since we are replaying a *sensori-motor trajectory*, we can use temporal position estimate τ_t to extract reference observation $O_{ref}(t) = \mathcal{T}_{sm}(\tau_t).O$ and reference controls $U_{ref}(t) = \mathcal{T}_{sm}(\tau_t).U$. Then, from reference observation, we can compute error from current robot configuration to reference one. Finally, with reference controls and configuration error, we could apply a well-tuned control law given by control theory; and no doubt that robot would replay accurately its sensori-motor trajectory.

As we wanted to design a fully Bayesian application, we used inspiration from fuzzy logic control([10], [6]) to build a probabilistic control law, expressed as a Bayesian data fusion problem: $P(U \mid \xi\ U_{ref})$ being expressed with $P(U \mid U_{ref})$ and $P(U \mid \xi)$. Due to space limitation, this will not be developed in this paper. We refer interested reader to [17] for details about this implementation.

6.2 Obstacle avoidance

Since our trajectory replay was designed to work in a moderately dynamic environment, we use controls computed for trajectory replay as inputs in our obstacle avoidance module.

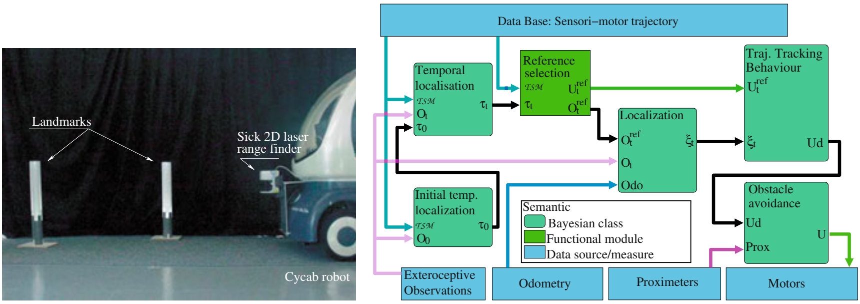
This module was presented in previous articles ([11], [17]). Its principle is similar to such methods as *Dynamic Window*[5] or *Ego-Kinematic Space*[9]. Its specificity is mainly its expression as a Bayesian inference problem, making it particularly well suited for integration in this paper's framework.

Basically, it takes as inputs data from proximity sensors and desired commands decided by an upper-level module (path planner, trajectory replay...) and uses them to compute commands really applied to the robot. These commands are built so as to follow desired commands as much as possible while granting security. We refer readers to our previous works for more details on this module.

7 Implementation

7.1 Experimental platform

Our experimental platform is a robotic golf-cab called CyCab, prototyped at INRIA and commercialised by french company Robosoft. Its main specificity resides in its



(a) The CyCab (b) Software architecture

Fig. 3. The CyCab robot and its software architecture

ability to steer both axles with different angles. This ability will not be fully used in this paper: we will steer both axles, but with equal angles.

For exteroceptive perception, our mini-car is equipped with a Sick laser range finder with an efficient range of 30 meters and an uncertainty of about 5 centimetres. Cylinders covered with reflector sheets act as landmarks in our environment(see fig. 3.a). We developed a preprocessing algorithm which detects out landmarks in the laser scans. By this way, we can consider that our robot is equipped with a landmark detector, and forget about the complete laser scan for the remaining of this paper. It should be noted that there is no distinctive marks on our landmarks, so they all return same sensor output. In the context of this robot, a configuration is the position of the middle of its rear axle (x, y) and its orientation θ. An observation is a set of $n \in \mathbb{N}$ positions $(O_i.x, O_i.y)$ of observed landmark in sensor frame (note that n is not constant).

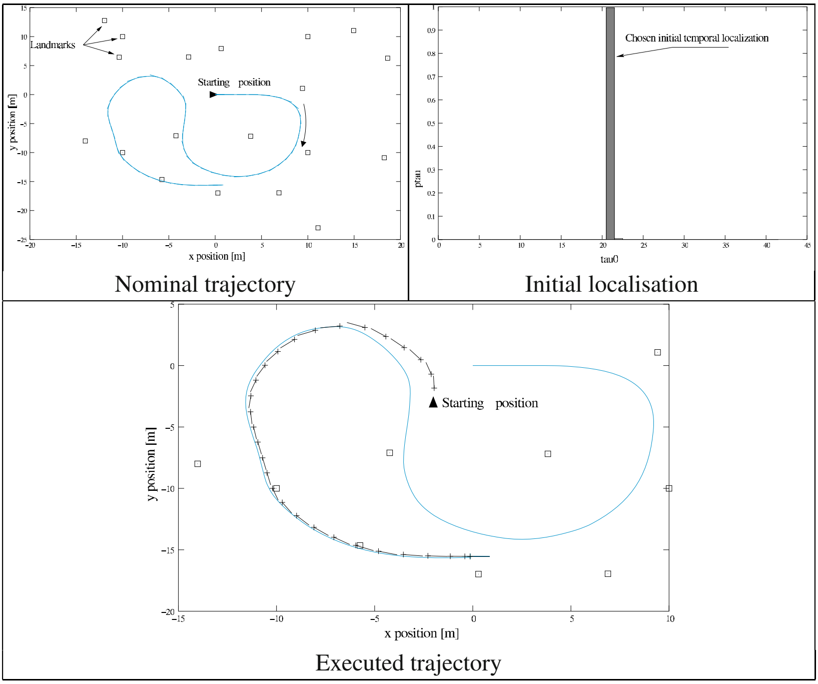
7.2 Software architecture

Our complete application architecture is presented in fig. 3.b. Each Bayesian class, represented by a big rounded box, expresses a distribution $P(\text{outputs} \mid \text{inputs})$, from which a unique value is synthesised, in order to reduce complexity and to achieve real-time. Using a multi-threaded implementation, and optimised Bayesian inference software, state tracking and controls generation can be done at about $50Hz$.

8 Results

8.1 Results on simulated platform

Results presented on figure 4 are computed on a simulated CyCab, equipped with a simulated sensor. The full sensori-motor trajectory was recorded on the trajectory depicted by figure 4 upper left. Then the system is initialised as shown by the black triangle in the lower part of the figure. Initial localisation is performed and give the distribution $P(\tau(0))$ presented in upper right part of the figure. Since this distribution is close to a Dirac distribution, the localisation system is completely certain of its results. Trajectory tracking is then performed and the system is rapidly on the tracks of the reference trajectory.



Nominal trajectory

Initial localisation

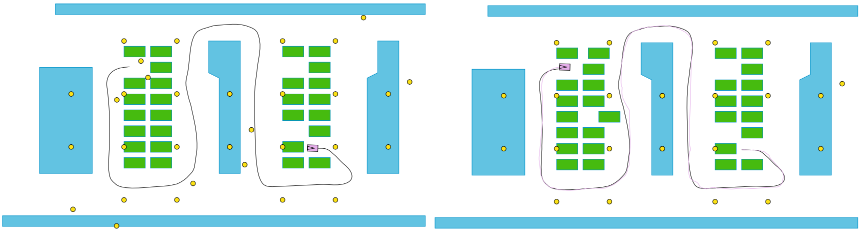
Executed trajectory

Fig. 4. Initialisation and replay, in a simulated trivial environment

Figure 5 gives an overview of the robustness of the approach in a more realistic environment: a simulated parking area (buildings in blue, cars in green, landmarks as yellow circles). On the left part, we show the reference trajectory with robot starting position (in magenta) and initial environment. Before starting to replay the trajectory we remove 10 out of 34 landmarks and we move one of the cars in order to make it interfere with the reference trajectory. Executed trajectory is shown in magenta on the right part: tracking is performed accurately while there is no risk of collision, even with strong curvatures and missing landmarks. Deformations of the trajectory occur around the moved car and when the reference trajectory was too close to the buildings.

8.2 Results on real platform

Figure 6 illustrates results of replayed trajectory on our real car-like robot. Our landmark detector is used as sensory input, and obstacle avoidance is used to check

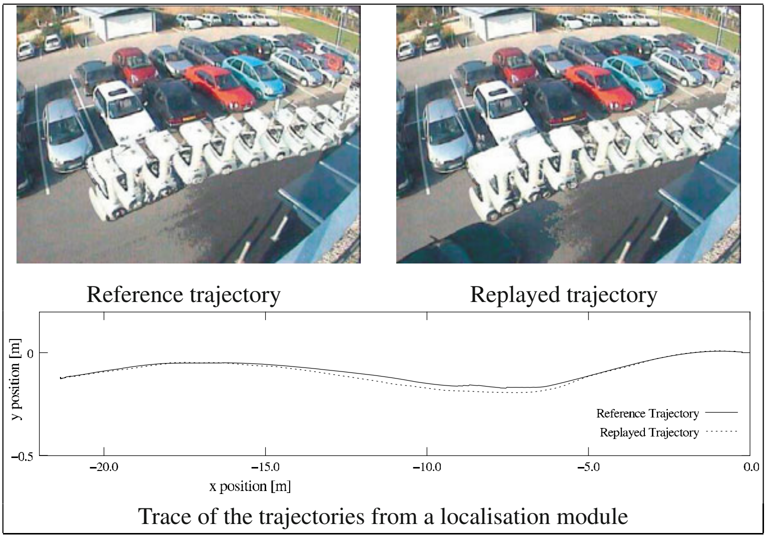


Reference trajectory

Replayed trajectory

Fig. 5. Sensori-motor trajectory tracking in a simulated parking area

whether proposed controls are safe. Trajectory replay is executed quite accurately at 2 m/s, moving at a few tenth of centimetres from parked cars.



Reference trajectory

Replayed trajectory

Trace of the trajectories from a localisation module

Fig. 6. Sensori-motor replay on a car-like vehicle

9 Conclusions

This paper presented how behavioural replay of a sensori-motor trajectory can be expressed as a fully Bayesian application: temporal and spatial localisation, trajectory tracking and obstacle avoidance were successfully implemented and integrated on a simulated robot and on a car-like autonomous vehicle on the car park area of our institute. Using this Bayesian modular framework, we developed a set of tools that have been used not only in this work, but also in a more classical robotic application: trajectory planning and execution in a moderately dynamic environment[17].

Acknowledgement

This work was partially supported by the European project IST-2001-32115 *“Bio-Inspired Brain & Artefacts”* (`http://www-biba.inrialpes.fr`).

References

1. R.C. Arkin. Reactive robotic systems, 1991.
2. P. Bessière and BIBA-INRIA Research Group. Survey: Probabilistic methodology and techniques for artefact conception and development. Technical Report RR-4730, INRIA, Grenoble, France, February 2003. http://www.inria.fr/rrrt/rr-4730.html.
3. R. P. Bonasso, D. Kortenkamp, D.P. Miller, and M.G Slack. Experiences with an architecture for intelligent reactive agents. In *Proc. of the Int. Joint Conf. on Artificial Intelligence*, 1995.
4. F. Chaumette. Visual servoing using image features defined upon geometrical primitives. In *Int. Conf. on Decision and Control*, volume 4, pages 3782–3787, 1994.
5. D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33, March 1997.
6. Th. Fraichard and Ph. Garnier. Fuzzy control to drive car-like vehicles. *Robotics and Autonomous Systems*, 34(1):1–22, December 2000.
7. J. Hermosillo, C. Pradalier, S. Sekhavat, and C. Laugier. Experimental issues from map building to trajectory execution for a bi-steerable car. In *Proc. of the IEEE Int. Conf. on Advanced Robotics*, Coimbra (PT), July 2003.
8. J. Hermosillo, C. Pradalier, S. Sekhavat, Ch. Laugier, and G. Baille. Towards motion autonomy of a bi-steerable car: Experimental issues from map-building to trajectory execution. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Taipei (TW), May 2003.
9. J. Santos-Victor J. Minguez, L. Montano. Reactive navigation for non-holonomic robots using the ego kinematic space. In *Int. Conf. On Robotic and Automation*, 2002.
10. Lawrence A. Klein. *Sensor Data Fusion Concepts and Applications*. SPIE, 1993.
11. C. Koike, C. Pradalier, P. Bessière, and E. Mazer. Proscriptive bayesian programming application for collision avoidance. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Las vegas, NV (US), October 2003.
12. F. Lamiraux, S. Sekhavat, and J.-P. Laumond. Motion planning and control for hilare pulling a trailer. *IEEE Trans. on Robotics and Automation*, 15(4):640–652, August 1999.
13. J.-P. Laumond, T. Siméon, R. Chatila, and G. Giralt. Trajectory planning and motion control for mobile robts. In J.-D. Boissonnat and J.-P. Laumond, editors, *Geometry and Robotics*, volume 391 of *Lecture Notes in Computer Science*, pages 133–149. Springer, 1989.
14. O. Lebeltel, P. Bessière, J. Diard, and E. Mazer. Bayesian robots progamming. *Autonomous Robots*, 2003. In Press.
15. E. Malis, G. Morel, and F. Chaumette. Robot control using disparate multiple sensors. *Int. Journal of Robotic Research*, 20(5):364–377, 2001.
16. C. Pradalier and P. Bessière. Perceptual navigation around a sensori-motor trajectory. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, April, 2004.
17. C. Pradalier, J. Hermosillo, C. Koike, C. Braillon, P. P.Bessière, and C. Laugier. Safe and autonomous navigation for a car-like robot among pedestrian. In *IARP Int. Workshop on Service, Assistive and Personal Robots*, Madrid (ES), October 2003.

List of Authors