Final Program and Book of Abstracts Third Swedish Workshop on Autonomous Robotics SWAR '05

Petter Ögren (Editor)

Kista, Stockholm, September 1-2, 2005

FOI-R-- 1756 --SE
ISSN 1650-1942

Systems Technology
Scientific report

October 2005

Final Program and Book of Abstracts

Third Swedish Workshop on Autonomous Robotics

SWAR '05

FOI-R--1756--SE Systems Technology October 2005
ISSN 1650-1942 Scientific report

Issuing organization	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency Systems Technology SE-164 90 Stockholm	FOI-R—1756--SE	Scientific report
	Research area code 7. Mobility and space technology, incl materials	
	Month year October 2005	**Project no.** E6941
	Sub area code 71 Unmanned Vehicles	
	Sub area code 2	
Author/s (editor/s) Petter Ögren (Editor)	**Project manager** Petter Ögren	
	Approved by Monica Dahlén	
	Sponsoring agency FMV	
	Scientifically and technically responsible Karl Henrik Johansson	

Report title

Final Program and Book of Abstracts, Third Swedish Workshop on Autonomous Robotics, SWAR 05

Abstract

This document contains the final program and 36 two page extended abstracts from the Third Swedish Workshop on Autonomous Robotics, SWAR'05, organized by FOI in September 1-2, 2005. SWAR is an opportunity for researchers and engineers working in the field of robotics and autonomous systems to learn about activities of neighboring institutions, discuss common interests and initiate new cooperations. The previous workshops, SWAR'00 and SWAR'02, were hosted by Örebro University and KTH, respectively. SWAR'05 gathered more than 80 participants from 25 different organizations, all over Sweden.

Keywords

Autonomy, Robotics, SWAR, Workshop

Further bibliographic information	Language English
ISSN 1650-1942	**Pages** 98 p.
	Price acc. to pricelist

Utgivare FOI - Totalförsvarets forskningsinstitut Systemteknik 164 90 Stockholm	**Rapportnummer, ISRN** FOI-R--1756--SE	**Klassificering** Vetenskaplig rapport
	Forskningsområde 7. Farkost- och rymdteknik, inkl material	
	Månad, år Oktober 2005	**Projektnummer** E6941
	Delområde 71 Obemannade farkoster	
	Delområde 2	
Författare/redaktör Petter Ögren (redaktör)	**Projektledare** Petter Ögren	
	Godkänd av Monica Dahlén	
	Uppdragsgivare/kundbeteckning FMV	
	Tekniskt och/eller vetenskapligt ansvarig Karl Henrik Johansson	

Rapportens titel

Slutgiltigt program och samtliga bidrag i sammanfattning från konferensen SWAR '05

Sammanfattning

Denna rapport innehåller program och 36 stycken tvåsidiga "extended abstracts" från konferensen SWAR'05, *Third Swedish Workshop on Autonomous Robotics,* som organiserades av FOI i Kista under 1-2 September 2005. SWAR erbjuder en möjlighet för forskare och ingenjörer verksamma inom robotik och autonomiområdet att träffas, utbyta forskningsresultat och erfarenheter samt diskutera möjliga samarbeten. De tidigare konferenserna, SWAR'00 och SWAR'02 organiserades av Örebro universitet respektive KTH. SWAR'05 samlade mer än 80 deltagare från 25 olika organisationer over hela Sverige.

Nyckelord

Autonomi, robotik, swar, konferens

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN 1650-1942	**Antal sidor:** 98 s.
Distribution enligt missiv	**Pris: Enligt prislista**

SWAR'05
Final Program and Book of Abstracts

Kista, Stockholm, September 1-2, 2005

SWAR '05 Aims and Objectives

Over the last few years, robotics has continued to expand from its factory birthplace into new unstructured and populated environments such as households and outdoor areas. Today, an increasing number of researchers and developers all over Sweden are actively participating in this expansion.

The Third Swedish Workshop on Autonomous Robotics, SWAR'05, is organized by FOI, the Swedish Defence Research Agency. It is an excellent opportunity for researchers and engineers working in robotics and autonomous systems in a wide sense to learn about activities of neighboring institutions, discuss common interests and initiate new cooperations. The previous workshops SWAR'00 and SWAR'02, hosted by Örebro University and KTH, respectively, were met with great enthusiasm by the participants.

Organizer: Petter Ögren, FOI. Program committee: Karl Henrik Johansson, KTH (Chair); Patrick Doherty, Linköping University; Tom Duckett, Örebro University; Anders Robertsson, LTH; Anton Shiriaev, Umeå University.

Sponsoring Organization: FMV, Swedish Defence Materiel Administration

Table of Contents

SWAR'05 Program at a Glance

Day 1 (9:30-16:55)

09:30	Coffee and Registration	
09:45	**Plenary 1:** Autonomous GIS-supported Positioning and Navigation	Fredrik Gustafsson, Linköping University
10:35	**Session UGV**	
10:35	Follow the Past – A Path Tracking method Using Recorded Orientation and Steering Commands,	Ringdahl, Hellström, Umeå U
10:50	Adaptive and Predictive Control for Off-road Mobile Robots Path Tracking	Lenain, Thuilot, Berducat, Martinet, CEMAGREF, LASMEA, FR
11:05	Autonomous Robot Navigation in a Public Nature Park	Andersen, Andersen, Ravn and Blas, DTU
11:20	Laser Guided Vehicles, LGV's: Industrial Experiences and Needs for the Future	Wernersson, Hyyppä, Åström, Hedström, Rahm, Luleå Tekniska U., Lund U., Danaher Motion AB
11:35	Spherical Mobile Robots for Extreme Conditions	Kaznov, Rotundus AB
11:50	Evaluations of a UGV in MOUT missions from a Man-System Interaction perspective	Lif, Jander, Borgvall, FOI
12:05	Military use of Unmanned Ground Vehicles in Urban Terrain	Lundberg, Christensen, Hedström, FHS, KTH
12:20	**Lunch**	
13:20	**Session Navigation and Sensors**	
13:20	Topological Map Building for Mobile Robots using Omnidirectional Vision	Wahlgren and Duckett, Örebro U.
13:35	Tracking for Human Augmented Mapping	Topp and Christensen, KTH
13:50	Automatic Building Detection for Mobile Robot Mapping	Persson and Duckett, Örebro U.
14:05	Closed Contour Reconstruction using Iterated Smoothing Splines	Karasalo, Hu and Martin, KTH, Texas Tech U.
14:20	Real-time Hough using Reconfigurable Hardware	Asplund, Bergkvist and Savo, Mälardalen U.
14:35	A Comparative Study for WiFi Localization in a Dynamic Environment	Duran, Halmstad U.
14:50	Fault Detection for Increased Robustness in Navigation	Sundvall and Jensfelt, KTH
15:05	Fast Laser Based Feature Recognition	Larsson, Broxvall, Örebro U.
15:20	**Coffee**	
15:40	**Session Architectures and User Interaction**	
15:40	Selection of Virtual Fixtures Based on Recognition of Motion Intention for Teleoperation Tasks	Aarno Ekvall and Kragic, KTH
15:55	A Software Infrastructure for Sensors, Actuators and Communication	Johansson and Hellström, Umeå U.
16:10	A Knowledge Processing Middleware Framework and its Relation to the JDL Data Fusion Model	Heintz and Doherty, Linköping U.
16:25	The Worlds Simplest Mechanism Simulator or Making Engineering Students Suddenly Discover that They Want to Devour Mathematics and Physics	Martin Nilsson, SICS, Mälardalen U., Lund U.
16:40	Binary Agents for the Control of Autonomous Vehicles	Wallace, FRONTIOS
19:00	Dinner	

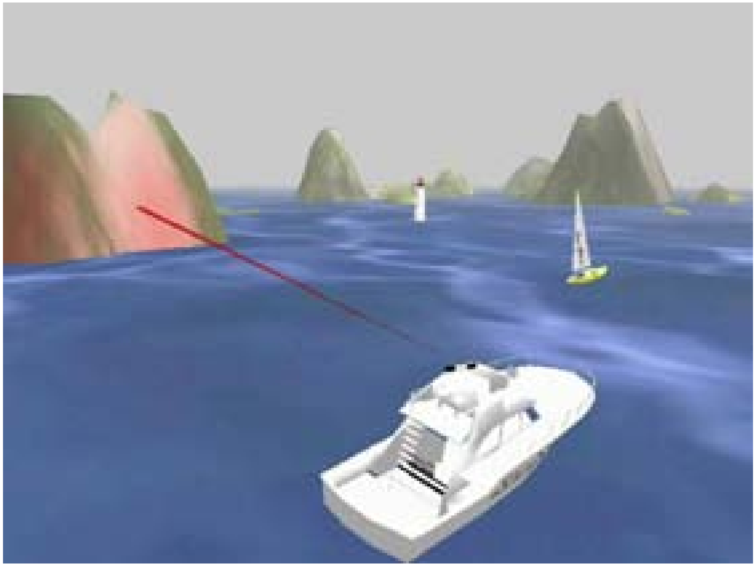
Day 2 (8:30-15:00)

08:30	**Session Manipulation**	
08:30	Telerobotics, for use in Contaminated Workspaces	Forsberg, Wernersson, Luleå Tekniska U.
08:45	Constrained Path Planning for Mobile Manipulators	Lingelbach, Aarno and Kragic, KTH
09:00	Simulating tactile Sensors in robotic grasping	Tegin, KTH
09:15	Productive Robots and the SMErobot project	Nilsson, Johansson, Robertsson, Bischoff, Brogårdh, Hägele, Lund U., Kuka Roboter GmbH, ABB Robotics.
09:30	**Coffee**	
09:50	**Session Multi Vehicles and Control**	
09:50	Symmetries in the Coordinated Rendezvous Problem	Speranzon, Carli, Fagnani and Zampieri, KTH, Padova U., Politecnico di Torino
10:05	Modification via Averaging of Passivity-based Control for Orbital Stabilization	Shiriaev, Freidovich, Umeå U.
10:20	Stabilizations of Vehicle Formations- A Case Study	Gattami, Berglund, Lund U.
10:35	Motion Planning and Dynamical Positioning for a Fleet of Underactuated Ships	Siriaev, Robertsson, Friedovich, Umeå U., Lund U.
10:50	Flocking with Obstacle avoidance, A New Distributed Coordination Algorithm Based on Voronoi Partitions	Lindhe, Johansson, Ögren, KTH, FOI
11:05	**Lunch**	
12:05	**Session UAV**	
12:05	A Low-cost UAV-system for Forestry and Agricultural Applications	Hagner and Friström, SLU, Smartplanes AB
12:20	Using Equivalence Classes of Multi-target Paths for Sensor Plan Evaluation	Mårtenson och Svensson, FOI
12:35	Combining Path Planning and Target Assignment to Minimize Risk in SEAD Missions	Ögren, Winstrand, FOI
12:50	On-line Trajectory Planning using Adaptive Temporal Discretization	Anisi, KTH, FOI
13:05	Creating Quality Imagery and Video from Inexpensive UAV	Klomp and Sandström, Uppsala U.
13:20	Information-Theoretic Approach for Concurrent Path and Sensor Planning for a UAV with EO/IR Sensors	Skoglar, Nygårds, Ulvklo, Ögren, FOI
13:35	Reconfigurable Path Planning for an Autonomous Unmanned Aerial Vehicle	Wzorek and Doherty, Linköping U.
13:50	**Coffee**	
14:10	**Plenary 2:** Autonomy on the SMART-1 Lunar Probe	Per Bodin,Swedish Space Corporation
14:50	Concluding Remarks	

Plenary 1: Autonomous GIS-supported Positioning and Navigation

Fredrik Gustafsson
Linköping University

The recent developments of computer-intensive methods for non-linear filtering, as the particle filter, enable new approaches to incorporate geographical information systems (GIS) in positioning and navigation applications. Here, information from standard sensors as inertial sensors, radar and vision systems are mixed with implicit constraints from GIS utilizing a motion model, which can be quite simple in many cases. The presentation includes positioning applications from road-bound vehicles, aircraft and sea vessels.



The particle filter computes the ship's position and heading by comparing the radar response from shore to a sea chart. This robust position estimate can be used as a support or backup to GPS to mitigate the navigation system's vulnerability to un-intentional or intentional disturbances.

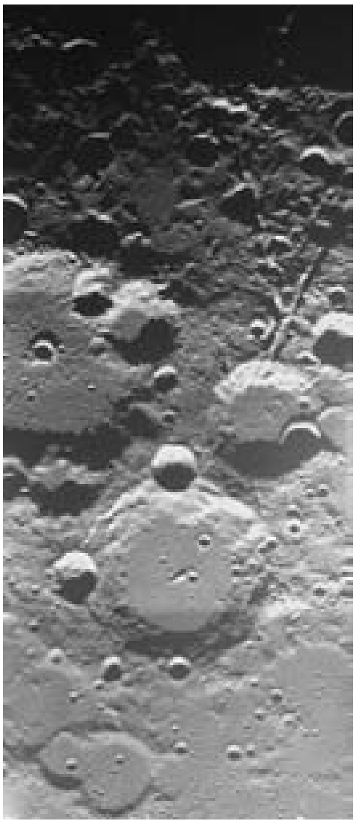
Plenary 2: Autonomy on the SMART-1 Lunar Probe

Per Bodin
Swedish Space Corporation
Solna, Sweden

SMART-1 was captured into lunar orbit on November 16, 2004 after having travelled 332 orbits around the Earth since its launch on September 27, 2003. The final lunar observation orbit was reached on February 27, 2005 marking the beginning of the scientific observations phase.

SMART-1 is the European Space Agency's first lunar mission. The Swedish Space Corporation (SSC) has been the prime contractor for the spacecraft and has also developed several of the on-board subsystems. The Attitude and Orbit Control system is one of these subsystems. The arrival to the Moon marks the completion of the primary mission objective which has been to demonstrate the use of Solar-Electric Propulsion for low-thrust transfer between the Earth and the Moon.

The presentation will give an overview of the SMART-1 mission in general with focus on the autonomy within its Attitude and Orbit Control System.



One of the first close-up images of the lunar surface taken by the AMIE instrument on SMART-1. The image was taken from a distance of 4000 km. The second largest crater in the image is called Pascal (middle bottom) and is situated at 74° north lunar latitude and 70° west lunar longitude. (Credits: ESA/Space-X)

Session UGV

Follow the Past - A Path Tracking Method Using Recorded Orientation and Steering Commands

Extended Abstract

Ola Ringdahl, Thomas Hellström
Department of Computer Science
Umeå University
901 87 Umeå, Sweden
{ringdahl, thomash}@cs.umu.se

1. Background and Introduction

The work presented in this paper is part of the project Autonomous Navigation for Forest Machines (Georgsson et al., 2005) at Umeå University, Sweden. The goal of this project is to develop a path-tracking forest vehicle. A human driver drives the path once, while a computer is continuously recording position, velocity, orientation, and steering angle. This information is then used to control the vehicle each time it autonomously travels along the path. If the vehicle gets off course, for example as a result of avoiding an obstacle or because of noise in the positioning sensors, the developed path-tracking algorithm steers like the driver, plus an additional angle, based on the distance to the path and the error in orientation. Traditional algorithms, like Follow the Carrot (Barton, 2001) and Pure Pursuit (Coulter, 1992), use position information only, and sometimes run into problems that can be avoided, by taking into account additional information from the human driver. There are many variations on the basic Pure-Pursuit algorithm. The Adaptive Pure-Pursuit algorithm (Hebert et al., 1997) addresses computational issues and stability at high speeds. The Feedforward Pure-Pursuit algorithm (Hebert et al., 1997) simulates the outcome of various control commands, and selects the most appropriate one. In (Coulter, 1992), the average curvature of the path is used instead of the recorded steering angle in a proportional path-following algorithm. Other researchers, e.g. (Ollero et al., 2001), have approached the problem with a fuzzy-logic controller that uses the same additional information as our suggested algorithm, but has a more complex design. A brief survey of more path-tracking control algorithms can be found in (Mäkelä, 2001).

2. Path Tracking and Obstacle Avoidance

To navigate safely through the forest, a new path-tracking algorithm named *Follow-the-Past* has been developed. Traditional algorithms like Follow-the-Carrot (Barton, 2001) and Pure-Pursuit (Coulter, 1992) use position information only, and sometimes run into problems that can be avoided by taking into account additional recorded information from a human driver. If the vehicle deviates from the recorded path, for example as a result of avoiding an obstacle, or because of noise in the positioning sensors, the Follow-the-Past algorithm steers like the driver, plus an additional angle, based on the distance to the path. The algorithm is described in the following section. More details and test results are found in (Hellström and Ringdahl, 2004).

2.1 Follow-the-Past Algorithm

While manually driving along the path, the orientation and steering angles are recorded together with the position at every moment. The recorded position (x', y'), the recorded orientation θ' and the recorded steering angle ϕ' are used by three independent behaviors:

- ϕ_β: Turn towards the recorded orientation θ'
- ϕ_γ: Mimic the recorded steering angle ϕ'
- ϕ_α: Move towards the path

Each behavior suggests a steering angle and is reactive, i.e. operates on the current input values; orientation, steering angle, and shortest distance to the path. ϕ_α uses recorded positions (x', y') and actual position (x, y) as inputs. ϕ_β uses recorded orientation θ' and actual orientation θ as inputs. ϕ_γ uses the recorded steering angle ϕ' as input. The three behaviors are fused into one action, the commanded steering angle ϕ_t, as shown in Fig. 1.

To function in the forest machine application, the path-tracking behavior has been combined with VFH+ (Ulrich and Borenstein, 1998) for obstacle avoidance. The HIMM grid used in the original VFH+ algorithm is replaced by an occupancy grid with Bayesian updating.

3. Results

The system has been successfully implemented on the simulator and on the Pioneer robot. The developed al-

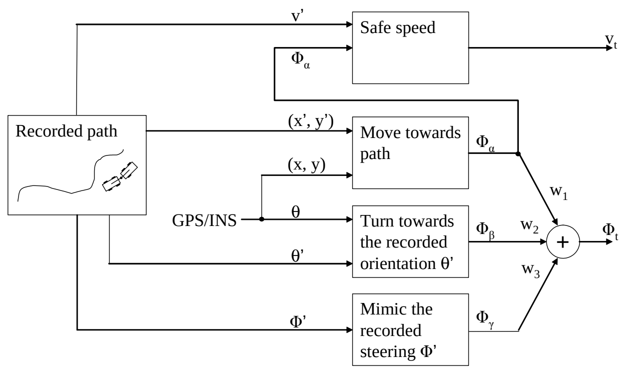


Figure 1: Path tracking with reactive control of steering angle ϕ_t.

gorithm for path tracking performs very well, see Figure 2. Initial tests with a full scale forest machine shows promising results as well, but more tests has to be done. Sensor tests and system adjustments for the forest machine are in progress and planned to be completed during 2005.

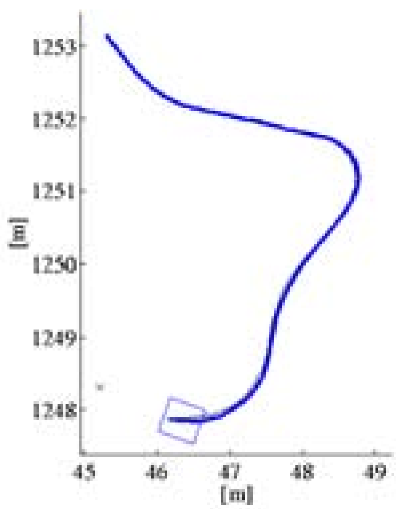


Figure 2: The Pioneer robot, represented by a box, is able to follow a predefined path almost perfectly with Follow the Past method. The thicker line shows how the vehicle has moved, while the thinner line show the recorded path.

Acknowledgements

This work was financed by The Kempe Foundations, VINNOVA, Land Systems Hägglunds, Carl Tryggers stiftelse, LKAB and Komatsu Forest AB. We acknowledge their support gratefully.

References

Barton, M. J. (2001). *Controller Development and Implementation for Path Planning and Following in an Autonomous Urban Vehicle*. Undergraduate thesis, University of Sydney.

Coulter, R. C. (1992). Implementation of the pure pursuit path tracking algorithm. Technical Report CMU-RI-TR-92-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Georgsson, F., Hellström, T., Johansson, T., Prorok, K., Ringdahl, O., and Sandström, U. (2005). Development of an autonomous path tracking forest machine — a status report —. Technical Report UMINF 05.08, Department of Computing Science, University of Umeå.

Hebert, M., Thorpe, C., and Stentz, A. T. (1997). *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*. KluwerAcademic Publishers.

Hellström, T. and Ringdahl, O. (2004). Follow the past — a path tracking algorithm for autonomous forest vehicles. Technical Report UMINF 04.11, Department of Computing Science, University of Umeå.

Mäkelä, H. (2001). *Outdoor Navigation of Mobile Robots*. PhD thesis, Helsinki University of Technology.

Ollero, A., Ferruz, J., Snchez, O., and Heredia, G. (2001). Mobile robot path tracking and visual target tracking using fuzzy logic. In Driankov, D. and Saffiotti, A., (Eds.), *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*, pages 51–72. Physica-Verlag Heidelberg.

Ulrich, I. and Borenstein, J. (1998). VFH+: Reliable obstacle avoidance for fast mobile robots. *IEEE Int. Conf. on Robotics and Automation*, pages 1572–1577.

Adaptive and predictive control for off-road mobile robots path tracking.

†Roland Lenain ‡Benoit Thuilot †Michel Berducat ‡Philippe Martinet

†Cemagref
24, avenue des Landais BP 50085
63172 Aubiere CEDEX
France
Tel : +33 473440690
Mail : roland.lenain@cemagref.fr

‡LASMEA
24, avenue des Landais BP 50085
63177 Aubiere CEDEX
France
Tel : 33473405211
Mail : benoit.thuilot@lasmea.univ-bpclermont.fr

Abstract:

If the development of algorithms dedicated for path tracking control of mobile robots is a well-known problem under the rolling without sliding condition (for vehicles moving on road for example), transfer of such control techniques are not always relevant for the case of mobile robots moving off road. Indeed, the application of control techniques developed under this assumption for off road vehicles demonstrates some unexpected behaviour, which depreciated considerably precision of a trajectory tracking (especially when vehicle executes a curve on a low adherent surface or when moving on a slope). Moreover, actuators used in natural environment are often more powerful and consequently less reactive, what introduces delays in vehicle dynamics. Such delays are emphasized by inertial effects, as the stability of off road vehicles requires the use of more important mass (and then inertia). An algorithm, taking into account for such phenomena and preserves the precision of path tracking for mobile robots, is presented. Capabilities of this approach are investigated through full-scale experiments, on farm tractor. Assistance of agricultural works indeed constitutes the application field of theoretical developments presented.

If localization problem is often an important topic for those robots, it is not considered here. Indeed, the approach here proposed is focused on single exteroceptive sensor (a Real Time Kinematic GPS), which supplies an accuracy of 2cm on coordinates signal at a sampling frequency of 10Hz. In addition, an angular sensor, based on steering wheel is present to ensure servoing of low-level actuator. Steering angle is here the unique control variable to be calculated, and velocity of robots is tuned manually and viewed, in control law, as a measured parameter, which can be variable.

Considering this measurement system and the numerous parameters required by a wheel-soil interaction model in a dynamical approach, the use of complex dynamical models (such as Pacejka or LuGree) is rejected. Instead, an extended kinematical model, adapted to describe sliding effects is designed. It takes account for vehicle behaviour (including sliding, which inevitably occurs on low adherent terrain) by the integration of two sliding parameters homogeneous to sideslip angles. However, it is necessary to estimate these parameters in order to feed the model and access to an accurate description of vehicle dynamics, with respect to the unique sensor approach. As considered parameters cannot be measured directly, an observer-based approach is then proposed, turned into a suitable way, in accordance with the duality between control and observation problems.

As an accurate model is then available, and considering this non-linear model can be turned into a linear one using a chained system form, a control law can so be designed. This law is able, on one part to compensate sliding phenomena in slow varying conditions, with respect to observer performances, but, on the other part, delays presents on actual vehicle (due to actuator properties and inertial effects) are neglected. As a result, accuracy of path tracking is preserved when sliding phenomenon can be considered as slow varying, but overshoots are present during transient phases. It is mainly the case during curve transition, appearing at begin or end of a curve.

Considering, that in path tracking case the entire trajectory to be followed is known and assuming on that it is possible to extract a model of the actuator response to a step of steering consign, a Model Predictive Control is introduced in the algorithm. Predictive action is here applied only on a part of control law, dedicated to the curvature servoing. An horizon of prediction is defined, allowing extraction of a future consign relevant with respect to reference path future configuration. Using the low level model, the actual steering angle can be predicted and error in the future between desired response and actual one can be computed. Then, the prediction algorithm consists in finding the set of control on the horizon of prediction, which minimizes this error. The first value of this set is then applied on the part of control dedicated to curvature servoing. As a result, overshoots are considerably limited. Moreover, even if only a low level model is considered, this predictive action is able to compensate implicitly delays induced by inertial effects. The accuracy is then also preserved during transient phases of curvature.

Finally, the global algorithm allows to achieve a high accurate tracking for off road mobile robots, whatever the path to be followed is, whatever the conditions of adherence are, and whatever configuration of terrain is (flat or sloppy ground). Accuracy reached by this control techniques (including adaptive and predictive actions), is close to sensor precision (considering the noise introduced by the roll effects of vehicle) during actual path following. Full scale experiments show that the vehicle is able to stay inside an acceptation interval of ±15cm. With respect to the application field, such accuracy meets the farmer expectation for agricultural tasks.

AUTONOMOUS ROBOT NAVIGATION IN PUBLIC NATURE PARK

Extended abstract

Jens Christian Andersen * Nils A. Andersen *
Ole Ravn * Morten Rufus Blas *

* *Automation Ørsted•DTU, Technical University of Denmark, building 326, DK-2800 Kgs. Lyngby, Denmark.*
(jca,naa,or)@oersted.dtu.dk

Abstract: This extended abstract describes a project to make a robot travel autonomously across a public nature park. The challenge is to detect and follow the right path across junctions and open squares avoiding people and obstacles. The robot is equipped with a laser scanner, a (low accuracy) GPS, wheel odometry and a rate gyro. The initial results show that this configuration is sufficient to traverse the park consistently, by following simple preplanned guidelines.

1. INTRODUCTION

This project is intended to produce a stable platform, that, with time, can be expanded from following a predetermined route, to be able to explore alternative routes, and draw a consistent map of the area. This is all part of an attempt to find a combination of tools and methods that will allow robots to cope autonomously in a real environments.

The environment for the project is selected to be a near by nature park (the Eremitagen near Copenhagen), where a high number of suitable paths, junctions and surface types are available, and not much traffic other than pedestrians, horses and bicycles.

The work is inspired by (Urmson *et al.*, 2004), (Klöör *et al.*, 1993) among others, and builds on results from (Blas *et al.*, 2005) and (Andersen and Ravn, 2004)

Fig. 1. The robot and some kids in the park last winter.

2. OBJECTIVES AND OVERVIEW

The objectives, so far, is to select, develop and implement the solutions, that is needed to make a robot autonomously and consistently follow a safe route to a destination, with as little preplanned guidance as possible. A route of about 4 km—across the park—is set as the initial navigation challenge.

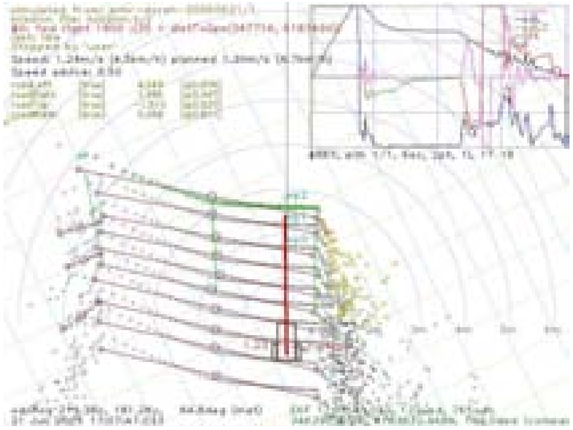


Fig. 2. Road detection on asphalt road. About 3 laser scans each second is used, these (the last 10) are shown overlaid with detected passable interval lines. Circles are drawn at ends and at the top of detected road profile. The image is from the robot monitoring application, and shows other support information.

The robot used—see figure 1—has two driving rear wheels and a caster front wheel. Each of the rear wheels are mounted with encoders for the odometry. A rate gyro is used to assist in determining the robot heading. The robot has a GPS receiver with an accuracy of about 5 meter (in open areas). The main sensor for detection of the environment is a laser scanner.

The selected solution has no map of the area, but is guided by a script, that in a few lines describe the route to follow.

The surface of the traversable tracks and open areas is either asphalt or gravel, the gravel parts may be corrupted by horse tracks and to some extend eroded by rainfall.

The selected guidance of the robot is comparable to the guidance you would give to a person, in this case like: Follow this asphalt road about for about 2 km, then, at the castle, turn left to the narrow road downhill (second left), follow that road, ignoring all side roads and junctions, until you reach the red gate destination point. The semantics should be the same, but in a syntax that matches the robot sensor capabilities.

The guidance script allow decisions based on many of the detected features from the laser scanner, from odometry and GPS position, and include simple calculation abilities, so that e.g. waypoints can be calculated based on obtained measurements.

3. RESULTS

The robot follows an asphalt road the first about 1.8 km using one of the road edges. This part runs smoothly. The road ends at a castle square. The GPS is here used to determine when to look for the castle square opening. Figure 2 shows data from this road.

The manoeuvres across the open gravel area in front of the castle, are planned based on the path driven on the last about 20 meter before the square were detected, from this reference, an odometry route is planned to the exit. The robot follows this route, avoiding any pedestrians, and gets to the start of the exit road. The exit road is rather rough and the grass edges is rather worn down, and the re-acquisition of the road edges are therefore a bit problematic. This is one of the situations, where consistency could be better.

The narrow gravel road with irregular edges behind the castle is traversed following the top of the road profile. At times the robot gets tilted on the uneven surface, but manages to keep track of the road profile, and avoid the rough edges.

When crossing a wider road (again with a highly curved profile) the ability to plan a waypoint based on the last 30 meters of position history is very helpfull, as the actual heading, when detecting the crossing road, is inconsistent. After crossing the road on odometry, the re-detection of the road edge, triggers continuation of the path, following the top of the road profile. At places the gravel road profile has uneven or multiple tops, but the top of the profile is preferable to the edges, as these are uneven and impassable, by the robot, at many places.

On asphalt roads an adaptive threshold were used to detect transition to less smooth road edges. On the castle square a fixed threshold were used, as the adaptive tends to find smoother parts and see e.g. horse tracks as obstacles. The fixed threshold detects pedestrians and rough road edges only.

The laser sensor do not like falling snow, as this gives random detections at shorter distances, but also rain is a challenge. Rain makes the roads wet, and this tends to reflect the laser beam (especially water filled pits and wet asphalt), resulting in a maximum range measurement, i.e. for the laser scanner it seems like the road is interrupted by deep holes. To compensate for this, all calculations ignore maximum range measurements.

4. CONCLUSION

The ability to detect different road types, has successfully enabled the robot to follow a simple guidance script through the 4 km planned distance. The script is preplanned, and guides the robot when too look for which terrain features, and where to drive on the found road.

The planned goal of being able to drive across the park autonomously and consistently is almost fulfilled, and it is expected that the consistency, will be improved as new features, e.g. detection of detours, and new sensors, especially camera, gets integrated.

REFERENCES

Andersen, N. and O. Ravn (2004). SMR-CL, a real-time control language for mobile robots. In: *2004 CIGR International Conference*. Beijing, China.

Blas, M. R., S. Riisgaard, O. Ravn, N. Andersen, M. Blanke and J. C. Andersen (2005). Terrain classification for outdoor autonomous robots using 2D laser scans.. In: *2nd int. Conf. on Informatics in Control, Automation and Robotics, ICINCO-2005.*. Barcelona, 14-17 Sept.. p. 6.

Klöör, P. L., P. Lundquist, P. Ohlsson, J. Nygårds and Å. Wernersson (1993). Change detection in natural scenes using laser range measurements from a mobile robot. In: *proceedings of 1st IFAC International Workshop*. IFAC. Swedish Defence Research Inst. Linköbing, Sweden, Linköbing Sweeden. pp. 71–76.

Urmson, Chris, Joshua Anhalt and ... Michael Clark (2004). *High Speed Navigation of Unrehearsed Terrain: Red Team Technology for Grand Challenge 2004*. CMU-RI-TR-04-37, Carnegie Mellon University.

Laser Guided Vehicles, LGV's:
Industrial Experiences and Needs for the Future

Åke Wernersson[1,2], Kalevi Hyyppä[1], Karl Åström[3], Arne Hedström[4], Jonas Rahm[4]

[1] Computer Science & Electrical Engineering, Luleå Univ of Technology, (n.n@ltu.se)
[2] Laser Systems, FOI 32, Linköping (ake.wernersson@foi.se)
[3] Center for Mathematical Sciences, Lund U, http://www.maths.lth.se, kalle@maths.lth.se
[4] Danaher Motion Särö AB, http://www.ndc.se, (n.n@danahermotion.se)

Abstract
Freely navigating vehicles using directions to identical retroreflective tapes has been in industrial use since 1991. In this paper we give
- some experiences from more than 5000 *installations*, in industry, in mines, harbours etc
- outline the AutoSurveyor that *automatically* gives a map of the reflectors, SLAM.
- mention Teach-In where the robot, by a manual operation, learns the trajectory from, say, a loading station to the "navigation highway".

Among needs for the future we mention
- The need for sensing obstacles. This also includes objects hanging down from the ceiling!
- Sensing and navigation in cases where retroreflective tapes *can not* be used.
- Outdoor sensing that is robust booth to snow flakes, hoarfrost, rain and bright sunshine.

It is very essential that the vehicle / robot have reliable and predictable properties. This is both a question of safety and economy. The paper outlines some properties and discuss the future.

Keywords: navigation, retroreflective tapes, landmark association, auto surveying (SLAM), teach in, multi-sensor systems, association errors, robustness

1. Introduction – identical beacons

The basic idea behind the LazerWay™ navigation system is beacons using retroreflective tapes. The angles to the navigation beacons are detected in a *robust* way since the illuminating laser and retroreflectors gives a very good S/N ratio. A typical error is 1 milliradian.

Indoors, the navigation repeatability is typically +/- 2 mm at a speed of .5 m/s. In mines with Load-Haul-Dumps, LHD, the repeatability is a few centimetres at 20 km/h. A large LHD is 80 tons with 25 tons load. The wheels are soft and the steering articulated. No rate gyros.

In many applications there are the "navigation highways" and loading stations. By, say, a manual operation the robot can *learn* the trajectory - a Teach-In procedure. More details on early results can be found in [Hyy87] while [NDC] gives user info. It should be mentioned that this principle was used earlier to detect stars with optronics onboard *spinning* sounding rockets. The pattern of the detected stars was used for finding the *attitude* of the rocket along the trajectory.

2. Automatic Generation of the Reflector Map (SLAM) – AutoSurveyor

During early tests the reflector map was measure manually using teodolites! Of course, an automatic map generation method is needed. An algorithm solving the large number of non-linear equation was found in 1991 and the AutoSurveyor went into industrial use 1993. A fully automatic version AutoSurveyor II was launched in 1996. More details on these early cases of SLAM can be found in [Åst 91, 93] and [ÅstO 00].

3. The Association Problem – Ambiguities

Given a reflector map and a vector of detected angels. Since the reflectors are identical, the problem is then to match them together, the association problem. Using a data driven state estimator, missing occluded reflectors are not a problem. One bad case is if a false detect is associated with a reflector in the map. Another bad case is if detects are paired together in a wrong way. Thus, a large number of reflectors is not good since it increases the risk of association errors.

To illustrate the non-linear nature of the problem consider just two reflectors observed by the robot. Since there is no compass onboard, only the difference in angle carries useful information. The position uncertainty for the robot is the circle going through the two reflectors and the robot. This have in fact happened in practice, the robot was running and stopped. All reflectors except two were occluded. Due to linearization err's in the Kalman filter the estimate started to move along the mentioned circle. When the occlusion disappeared and the robot was to start again it was lost since the orientation drifted away! We conclude by mentioning some "facts"; The *combined* problem of association and state estimation is outside the framework of the Kalman filter. If the linearised estimator is used, the association uncertainty introduces *additional* terms in the Riccati equation.

4. Conclusions, needs for the future

The strength of this is reflector based navigation is *both* the *S/N ratio* in the detector and a *direct sensing* of the *orientation* of the vehicle. In many applications we can not use reflectors or beacons. How should we use multisensing (lasers, cameras and gyros) to build a reliable system? When the vehicle moves a few times it's own length it has to make a decision. This requires an error rate $<10^{-6}$ in many cases!

Using lasers and cameras it is possible to extract robust features from geometrical primitives like walls, corners, tree trunks etc. Navigation will then be relative to these landmarks. We also need a good angular stability. A rate gyro can do this job. For LaserWay the directions to the reflectors (+/- 1 millirad) gave the angular stability for the vehicle. Recall that the LHD's could navigate without rate gyros. It is very essential that the vehicle / robot have reliable and predictable properties. This is both a question of safety and economy. This application area is also in need for better theories to give model based self monitoring control systems.

References

[Hyy87] Hyyppä; *Optical navigation system using passive identical beacons*, Intelligent Autonomous Systems, Amsterdam, 8-11 December 1986, pages 737-741. Hertzberger and Groen, editors, North-Holland, 1987.

[NDC] http://www.ndc.se/ , http://www.lazerway.com/ , http://www.teach-in.com/

[Åst91] K Åström: *Where am I and What am I Seeing? Algorithms for a Laser Guided Vehicle*. Masters thesis. Dept of Mathematics. Lund University, 1991.

[Åst93] K Åström: *Automatic Mapmaking*, 1st IFAC International Workshop on Intelligent Autonomous Vehicles, Southampton, UK, Pergamon Press, 1993.

[ÅstO 00] Åström, K., Oskarsson, M.: Automatic geometric reasoning in structure and motion estimation. Pattern recognition letters 21: 1105-1113, 2000. More on websida

[WAH89] Wernersson, Wiklund, Andersson, Hyyppä, *Vehicle Navigation using "Image Information": on Association Errors*. Intelligent Autonomous Systems - 2, pp 814 - 822, Dec. 1989.

Spherical mobile robots for extreme conditions

Viktor Kaznov [1]

[1] *Rotundus AB, c/o UUAB, Uppsala Science Park, SE-751 83 Uppsala, Sweden,*
Email: viktor.kaznov@rotundus.se

Rotundus AB develops a revolutionary spherical robot concept, the so-called “SecuriSphere”. A moving mobile robot with all mechanics and sensors completely enclosed within a durable spherical shell. Size can vary from 10 cm to 3 m. Robot runs for 8-10 hours, at speeds up to 20-30 km/h. Weighs a few kilograms depending on size and sensors. Can operate in sand, snow, mud and water.

The robots’ extreme durability and ability to run in almost any outdoor environment makes it highly fit for:

- Security and surveillance of large high-security sites
- Military surveillance and reconnaissance
- Rescue and safety operations
- Planetary surface exploration

The current research is focused on security and surveillance purposes with two major interests, namely long time and distance excursions and sensor fusion analysis. The Rotundus robot is unique in its remarkable features. The current prototypes are 50 cm in diameter and weigh just 5 kg. These platforms can traverse for 8 hours in 20-30 cm of snow on one charging of the batteries, reach speeds of 10 km/h and in excess on harder surfaces. Applying “SecuriSphere” to security and surveillance applications offers a possibility to patrol large areas under any kind of whether and on any kind of surface. The first prototype has been tested on the NASA/Jet Propulsion Laboratory Mars Yard during the summer of 2004.

The robustness is one of the main features of the robot. All mechanical and electrical parts of the robot are placed inside the spherical shell and are protected against impacts, sand, dirt, water etc. The driving unit has no contact with the shell besides the contact with the diametric main axle via the transmission system. That makes the robot to an impact safe system. There are only two points where a direct impact could cause some damage. These are the attachment points of the main axle to the shell. The developed system solves this problem by offering a telescopic main axle with elastic joints, which make these points impact safe.

Rotundus studies a combination of the neural networks and the classical control algorithms to actively compensate and stabilize the system. This approach minimizes the need of computational power and enables a powerful low mass platform. Due to the sensor fusion approach utilized by the control system, the precision of the single sensors is not critical for the normal functioning, which reduces the single point failure risks.

Rotundus AB
c/o UUAB
Uppsala Science Park
751 83 Uppsala

www.rotundus.se
info@rotundus.se
+46 70 318 77 67

Evaluations of a UGV in MOUT missions from a Man-System Interaction perspective

Patrik Lif, Hans Jander, Jonathan Borgvall
Department for Man-System Interaction, Swedish Defense Research Agency

Unmanned Ground Vehicles (UGVs) has been used operatively in Military Operations in Urban Terrain (MOUT) for many years now (e.g., by the US Army in Iraq). Most probably, the use of UGVs will increase since there are many advantages with UGVs, especially concerning intelligence and safety during MOUT missions. The Swedish Army Combat School (MSS) has tested a UGV system for a couple of years. The current system is called SNOKEN 2 (see Figure 1). There are many technical issues to investigate, ranging from sensor technology to human-machine interface (HMI). However, there are important organizational matters to evaluate as well. Introducing a new system meant to increase a unit's ability might have other effects than presumed. Tactics might have to be adjusted, demand of personal resources might change (both in numbers and in competence), and the workload of not only the operator and his unit but also at the higher organizational levels might be effected. If issues like these are not properly investigated there is a risk of provoking a decreased overall performance of a unit, even though specific tasks might be performed more safely and efficiently with the new system. Therefore, evaluations of SNOKEN 2 at different levels have started in cooperation between MSS and the department for Man-System Interaction (MSI) at the Swedish Defense Research Agency (FOI). The focus has been on the following issues:

- How is the unit effected by using a UGV?
- Is it necessary to add resources to the unit for them to be able to conduct both ordinary assignments and handling a UGV system?
- Is it necessary to change the units working methods?
- Are organizational changes necessary?
- Does an UGV affect the ability to collect surveillance information at group and platoon level?
- How is the group and the group members affected by the system?

Figure 1. Section leader sitting to the left, UGV operator with control panel in the middle, and carrier of the vehicle (when not in operation) to the right.

Of secondary focus was to evaluate SNOKEN 2 from a HMI point of view. Technical questions were not of interest here, but the evaluation of the tactical and MSI aspects is of course connected to technical improvements and solutions.

Data were collected through interviews with operators, section leaders, and platoon commanders, and observations conducted by researchers and military personnel. All the participants identified both advantages and disadvantages of using the system. Positive were the ability to collect intelligence and minimizing personal damage. On the negative side, the time demanded to collect intelligence increased. The operator handling SNOKEN 2 decreased his ability to carry other materiel and solve his ordinary assignments which affected the section in a negative way. The mental workload was not affected according to the participants' subjective ratings. According to operators, the risk of detection by the enemy was also seen as less likely when using the UGV system. The section leaders and the platoon commanders also had the opportunity to watch the camera image from the UGV by looking at the operators screen. This enhanced the quality of the intelligence and increased their ability to make successful decisions. Thus, the platoon commanders was indirect affected since they could get better intelligence, but the system did not affect their ordinary duties as was the case for the operators. The results should also be connected to the four basic combat situations used by the Swedish armed forces, and pros and cons can be coupled to the tactical situations:

- Advance
- Enemy contact
- Attack the target
- Defence

During the MOUT exercise, it was clear that the pace during the advance phase varied a lot. In some situations the unit moved fast and the use of the UGV was not possible at these occasions. However, during the advance phase the unit also secured crossings which meant that they stopped. During these situations, the UGV was used successfully for intelligence and surveillance. In the enemy contact phase, the soldiers will always focus on the combat and thereby value of the UGV system is reduced, especially if additional personnel for handling the UGV are not introduced to the unit. When attacking a target the use of UGV can be of great importance for target identification and damage assessment. In the defence situation (e.g. defending a crossing), the UGV system was used with success in gathering new information to stay ahead of upcoming situations. While defending one crossing the system was used to gather information of the next crossing, to get an understanding of the terrain but also to search for enemies in the area.

According to the evaluation some functions probably need to be automated, for example during the advance phase were the soldier can not operate the vehicle. Here, an automated function could be implemented to make the UGV follow the soldier automatically. Another situation where automation should be considered is during enemy contact and during certain surveillance missions.

In summary, the UGV system can be used with success in a number of situations but it is also necessary to understand the disadvantages so the commander can decide in which situations he or she needs to use the UGV. One big disadvantage is that the pace of the group is limited, but on the other hand it could save lives and gather information that otherwise would not be possible.

MILITARY USE OF UNMANNED GROUND VEHICLES IN URBAN TERRAIN

Carl Lundberg
National Defence College (FHS),
Department of Military Technology (MTI)
Box 278 05
115 93 Stockholm, Sweden
carl.lundberg@fhs.mil.se

Henrik I Christensen, Andreas Hedström
Centre for Autonomous Systems (CAS),
Numerical Analysis and Computer Science (NADA),
Royal Institute of Technology (KTH)
100 44 Stockholm, Sweden
hic@nada.kth.se, ah@kth.se

Abstract

Robots have a potential to be a significant aid in high risk, unstructured and stressing situations such as experienced by police, fire brigade, rescue workers and military. In this project we have explored the abilities of today's robot technology in previously the mentioned fields. We have done this by studying the user, identifying scenarios where a robot could be used and implementing a robot system for these cases. We have concluded that highly portable field robots are emerging to be an available technology but that the human-robot interaction is currently a major limiting factor of today's systems. Further we have found that operational protocols, stating how to use the robots, have to be designed in order to make robots an effective tool in harsh and unstructured field environments.

Session Navigation and Sensors

Topological Map Building for Mobile Robots Using Omnidirectional Vision *

Christoffer Wahlgren and Tom Duckett
Centre for Applied Autonomous Sensor Systems
Örebro University, SE-701 82 Örebro, Sweden
{christoffer.wahlgren, tom.duckett}@tech.oru.se

I. Introduction

The long term aim of our research is to find an on-line, probabilistic mapping algorithm for mobile robot navigation in both indoor and outdoor environments. Needless to say, there are many challenges. Outdoor environments can be huge, unstructured and dynamic. Many of the existing solutions to the robotic mapping problem indoors will not be applicable outdoors [1], simply because the methods employed do not scale well to larger and more complex environments. Our first approach is to use a topological representation of the world. Later, metric information or additional topologies could be added to form a hybrid map, which would allow the robot to perform specialized tasks.

Topological approaches to the robotic mapping problem use landmarks to define the nodes (places) of the map. The gathering of landmarks into distinct places is a largely unsolved problem. Existing approaches include defining the places by hand, unsupervised clustering algorithms [2], and detection of “distinctive places” [3]. If the whole set of landmarks is available, the mapping problem can be viewed as a search problem to find the best topological map that fits the observed landmarks (and other sensor data, such as odometry). The landmarks can be artificial landmarks such as beacons, or natural occurring features extracted by sensors such as laser range finders, sonars or cameras.

When building the topological map, one must also consider the correspondence problem and perceptual aliasing (multiple places have similar appearance); we wish to maintain a one-to-one correspondence between nodes in our map representation and places in the real world. The classical – and very successful – approaches to this problem are various probabilistic techniques [1]. However, probabilistic approaches can be computationally expensive, often growing exponentially with the size of the map [4]. The method we propose can be considered probabilistic (because the link likelihood is determined by visual similarity of image sequences) and avoids the computational problem of considering multiple hypotheses.

II. Topological Mapping Using Vision

This section describes our proposal for topological mapping using vision. The approach utilizes local features extracted from panoramic images combined with a segmentation technique to derive a topological map from a sequence of images obtained by a mobile robot. At each time step, the new image is compared to the previous images and the result (i.e. the number of feature matches between the two images) is stored in a matrix. Figure 1 shows an example for a run of 602 images. Note that there are two indications of closed loops in the matrix from images around 40 to 420 and from the first to last images. When a map is required by the planning system, the topological map is created in two steps. First, the sequence of images is divided into smaller sections, where each section represents a place. This place detection need not consider the correspondence problem at all; instead, one area might very well be represented by two or more nodes. Second, the links between the nodes are generated by considering the result matrix and the result from the place detector. Although much work remains, the approach has been shown to give good preliminary experimental results in a fairly complex indoor environment (Figure 3).

A. SIFT - feature matching applied on panoramic images

SIFT, the Scale Invariant Feature Transform, first presented in 1999 [5], was developed and patented by David Lowe. The main characteristic of SIFT is that it uses a feature description that is invariant to scaling and rotation. It is also insensitive to changes in illumination and camera location.

The landmarks extracted from the images are the Modified Scale-Invariant Feature Transform (MSIFT) features, as described by Andreasson [6]. The images are obtained with an omnidirectional camera, which simplifies calculations, since we need not worry about the direction of the robot (Θ), only its position (x, y). An example of feature matching is shown in Figure 2.

B. Place detection

In our approach, a place is a collection of images that are considered similar enough, i.e., there is a sufficient number of feature matches between the images. The place detection algorithm relies heavily on the assumption that the images are acquired in sequence, and that the distance travelled between each image is not too great. The place detector uses a quick and greedy algorithm that simply searches for the longest image sequences. For an example of another approach, see [7].

*This work is partially supported by The Swedish Defence Material Administration

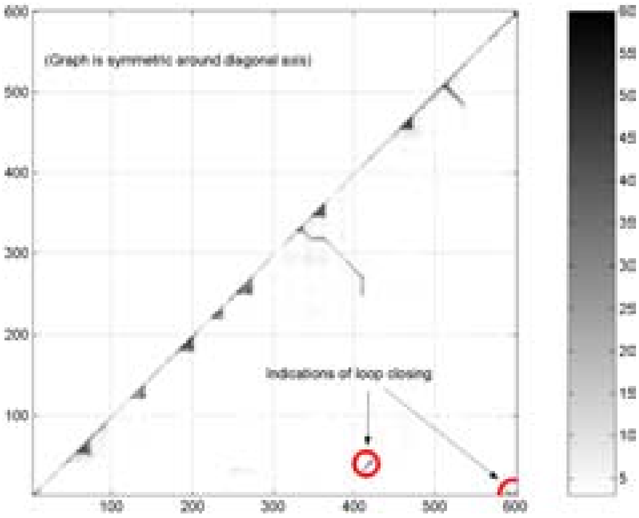


Fig. 1. Result matrix, illustrating the number of feature matches between images of a 602-image run.

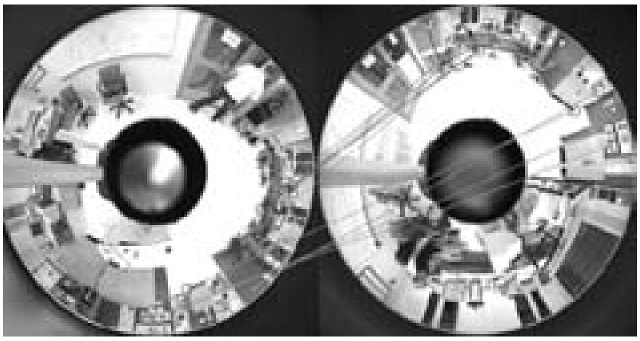


Fig. 2. Some of the features matched between two panoramic images.

By using only local feature matching as the criterion for determining boundaries between places, we do not require any knowledge about the geometry of the environment. This gives the algorithm the potential to work well in unstructured environments. On the other hand, there is also a chance that we will misclassify an image because of noise or occlusion. This can be solved by introducing a filter that removes images that do not "fit" into the sequences.

C. Building the topological map

To get a complete topological map, the places detected need to be connected to each other. Because the images are assumed to be obtained in sequence, there are *weak links* (following Lu and Milios [8]) between consecutive nodes in the sequence. In addition, a *strong link* exists between node A and node B when an image in node A matches an image in node B. This means that we expect strong links to form between nodes in larger areas, because some features are common to some images in the nodes. Also, we expect that if we revisit a previously visited area, a link will be created between the revisited node and the new node. Unfortunately, the feature matching is not perfect, so that *false links* will be created. Working

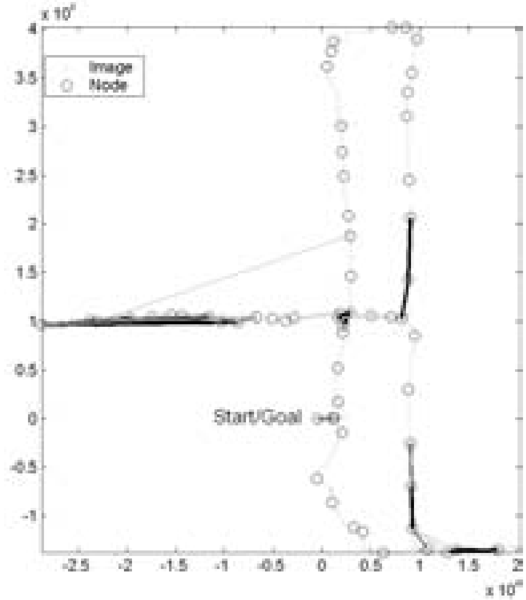


Fig. 3. Resulting probabilistic topological map. Link likelihood is illustrated by line boldness. Weak links are not shown.

towards a probabilistic approach, we propose the concept of link likelihood. Simply put, the link likelihood measures the probability that two nodes share common features. The link likelihood can be seen as function of the number of image matches between node A and B. An example of the resulting topological map can be seen in Figure 3. Darker lines indicate links with high likelihood; weak links are not shown. It is interesting to note that the two loops indicated in the result matrix are in fact found by this method. There is also one obvious false link. This link exist because of perceptual aliasing; it could be removed by increasing the matching threshold or by other improvements to the matching algorithm.

REFERENCES

[1] S. Thrun. Robotic mapping: A survey. Survey CMU-CS-02-111, School of Computer Science, February 2002.

[2] U. Nehmzow. "Meaning" through clustering by self-organisation of spatial and temporal information. In C. Nehaniv, editor, *Computation for Metaphors, Analogy and Agents*. Springer Verlag, 1999. Lecture Notes in Artificial Intelligence 1562.

[3] B. Kuipers. The spatial semantic hierarchy. *Artifical Intelligence*, 119:191–233, 2000.

[4] A. Ranganathan and F. Dellaert. Inference in the space of topological maps: An MCMC-based approach. In *Int. Conf. On Intelligent Robots and Systems*, 2004.

[5] D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. Int. Conf. Computer Vision ICCV*, pages 1150–1159, 1999.

[6] H. Andreasson and T. Duckett. Topological localization for mobile robots using omni-directional vision and local features. In *Proc. IAV 2004, the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, 2004.

[7] Z. Zivkovic, B. Bakker, and B. Kröse. Hierarchical map building using visual landmarks and geometric constraints. Technical report, Intelligent Autonomous Systems Group, University of Amsterdam, 2005.

[8] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

Tracking for Human Augmented Mapping *

Elin A. Topp and Henrik I. Christensen
Centre for Autonomous Systems
Royal Institute of Technology (KTH)
SE - 100 44 Stockholm, Sweden
{topp,hic}@nada.kth.se

I. Introduction

A key competence of a service robot is the ability to navigate in its initially unknown, arbitrary human inhabited environment, which requires a map. This map has to correspond to the user's representation of the environment to allow future task specification dialogues with respect to places, specific locations and objects. Thus we assume a form of tour scenario, in which the user introduces labels into a concurrently built map, while guiding the robot around – "Human Augmented Mapping". Hence the robot needs to follow the user, which requires a tracking method. Since we assume an arbitrary indoor context, the system should be able to handle also the presence of bystanders appropriately. We present the results of our tracking approach with respect to these two main purposes, a) following a particular person and b) handle the presence of other persons appropriately. In previous work we give a more comprehensive description of Human Augmented Mapping and tracking in that context [6].

II. Tracking for Following and Navigating

We will discuss two quality criteria for a tracking approach with respect to the two purposes named previously. Two different types of tracker failures can occur with respective effects on the reliability of the system. These two types are the loss of a target and a confusion of different targets due to ambiguous data association. For the purpose of following a loss could be handled by a respective interaction system, for example a dialogue system. Confusing two targets on the other hand can lead to critical situations, since a person unaware of the situation might be followed by the robot. For the passing of bystanders on the other hand, a confusion is not as critical as a loss, since still all targets (persons) would be tracked and could be handled. A loss would mean that a person is suddenly not a person, which would make the system fall back to plain obstacle avoidance. These reflections make a multiple target tracking approach appear as an appropriate solution to both problems.

A. The tracking system

Our laser range data based tracking system is based on the approach presented by Schulz et al. [4]. Their system uses leg detection and occupancy grids to detect people and distinguish them from other objects by movement. Detected features are associated to tracked targets with a sample based joint probabilistic data association filter (SJPDAF). Using this they can track multiple targets with a mobile robot, while motion compensation is done by scan matching.

We adopted the idea of using the SJPDAF approach for tracking and associating, but in contrast to Schulz et al. our detection and tracking allows handling of people standing still, which is useful for interaction. In contrast to numerous other approaches to people detection based on laser range data [3]–[5], we allow different types of features to generate our hypotheses, since depending on clothing the appearance of people in a laser scan varies significantly. We have three types of patterns that describe our features: *single leg, SL, two legs appropriately separated, (TL)* and *person-wide blob, (PW)*, together with a set of rules to keep the number of false alarms to a minimum.

*The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Robot Companion") and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

Each detected feature $z_j \in \{z_0, z_1, z_2, \dots z_n\}$ is assigned a posteriori probability β_{ij} that it was caused by the target $x_j \in \{x_1, x_2, \dots, x_m\}$. The computation of the β_{ij} is based on a sample representation for the targets. Each target x_i has its own sample set for state prediction and is updated according to β_{ij}.

As our main interest is not the tracking approach itself, but its performance in the context of person following and person passing, the tracker results need to be interpreted accordingly. With the help of a person handler module the targets are classified and marked with different flags. One of them is the user flag that is currently assigned to the target that is a) moving and b) closest to the robot in a certain region. This is a strategy used to overcome the lack of a dialogue system or other interactive means to determine the user.

To actually test the method in the context of following, the speed and turn rate to be send to the motor controller are computed depending on a constantly updated goal point in a certain distance to the user with the help of Nearness Diagrams to avoid obstacles.

B. System setup

Our testbed is an ActivMedia Performance PeopleBot. The robot is equipped with a SICK LMS-200 laser ranger finder mounted at a height of about 33cm. Other available sensors such as the camera, sonar or infrared sensors are currently not in use. For communication with sensors and the robot controller we use the Player/Stage[1] robot control software libraries.

III. Experimental results

To test the tracking approach in the discussed context, we used three different scenarios. One tests following and tracking of multiple targets in an artificially emptied "room". The other two reflect the behaviour of the tracker in a "real world" context. As we observed differences in the quality of the results, we present the two test types separately, referred to as setup #1, #2, and #3 respectively.

A. Experimental setup #1

In order to make sure that the number of persons present was controllable at any time during experiments, we defined an empty area by setting up a number of large plywood planks and cardboard pieces as "walls" for the experiments that involved a moving robot. We then defined a number of test scenarios that represented different combinations of a static or moving robot with a varying number of test persons present. With these tests we aimed to test the tracker under different test conditions. Regarding our quality measurements presented in section II we were mainly interested in problematic situations that might lead to confusions or the loss of a target. We therefore asked our test persons to walk at different speeds, cross each other in the field of view of the robot on purpose, "meet" in the middle of the room, "chat" and separate again, or perform unexpected changes in their moving direction.

During the tests all occlusions were handled correctly and no target was lost. This result could be confirmed by different tests under similar circumstances with the same models for movement and state prediction.

B. Experimental setup #2

As we cannot assume such test conditions all the time, we tested the tracker on data collected during a comprehensive user study conducted at our laboratory. The user study was a Wizard-of-Oz experiment and is described in detail by Green et al. [1]. The scenario for the experiment was a guided tour through a living room. 22 subjects got the task to ask the robot to follow, present different locations and objects in the room and test the robot's understanding by sending it to learnt locations and objects. Figure 1 shows a raw scan taken from a typical start position during the tests.

Running the tracking system on the data from the experiments showed that performance in this kind of real world environment

[1]http://playerstage.sourceforge.net

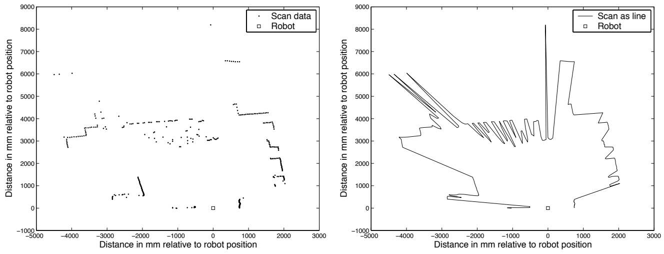


Fig. 1. The raw laser data (top) and the same data represented as polyline to show the data points in their angular order. The two peaks right in front of the robot are caused by the subject's legs, while the other peaks result from the table and chairs, that belonged to the experimental scenario.

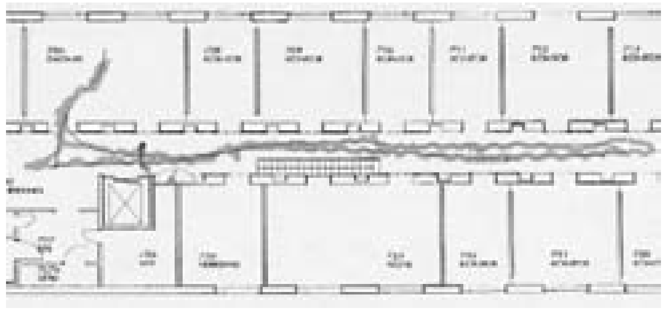


Fig. 2. The trajectories of the user (grey, thick line), the following robot (black, thin line) and a bystander crossing the way of the robot (small squares) between the robot and user.

was significantly worse than expected after the results from the previously reported tests. The user target got confused with other targets rather frequently.

The images in figure 1 show a clear resemblance between some of the patterns and the subject's legs, even if some of them appear too pointy. Still, such patterns can fall under the classification thresholds for legs, which gives a number of false hypotheses. Since these hypotheses do not appear as stable as a real person – especially when the robot is moving – the association is not as confident as needed, which might lead to confusions of the user target with other, false hypotheses. This is rather likely, when the user target is passing such another "target" closely.

As the task for the subjects was to show the robot around in a furnished room, it is scenario immanent that the user moves around between objects in the room. On the other hand, we could state, that in situations where the user was clearly distinguishable from disturbing objects, and those disturbing objects were detected reliably, the tracker and data association performed as expected. Occlusions were also handled properly in these situations.

C. Setup #3: Following through the office building

With this test we showed, that the system is still suitable for "real world" conditions if the disturbances can be reduced to a minimum by the choice of environment. The robot followed the test person out of the laboratory and along the hallway, covering a distance of about 25 meters, and returned – still following – to the laboratory. On the way back a bystander was asked to cross the way between the robot and the user. Figure 2 shows the part of the office building together with the trajectories. The experiment took approximately four minutes and a distance of about 50 meters was covered, including two door passages. A total number of 26 targets was detected throughout the whole time period, one was accidentally classified as "moving", but did not get confused with the user. The user target was tracked reliably over the complete time period and one occlusion of the user by a crossing bystander was handled as expected. The bystander target was classified as moving person correctly, so a respective person passing method could have handled the situation appropriately. In our test case the robot slowed down a bit, due to the influence of obstacles on the speed.

Summarising these tests on "real world" data we observed that

- the approach for tracking and data association is still a valid method for tracking multiple targets in the context of following a user or passing persons.
- problematic situations occur in "real world" scenarios, i.e., cluttered environments, when vicissitudinous false alarms lead to confusions.

Judging from these observations, we assume that we can improve the system for following and passing persons significantly by introducing other means for the detection of targets. Within the context of "showing the robot around" we have to deal with an unknown, cluttered environment. From preliminary analysis of the user study we can allege that persons in this context move more randomly compared to results from observations in long term experiments on a larger scale. Therefore we think that improving the detection to eliminate confusing false alarms is a better way to improve the system for our purposes than applying a more sophisticated motion model.

IV. Conclusion and Future Work

In this paper we presented our approach for tracking multiple targets with a mobile robot in the context of following a user and navigation in a human populated, but not too busy, environment as office buildings or regular houses. In this context we introduced the term *Human Augmented Mapping* with which we subsume aspects of various disciplines such as human robot interaction, people tracking and following, localization and mapping. We showed experimental results for the tracking approach under test conditions in an environment that did not contain any disturbing objects. Further we described the results from tests on data collected during a user study in a typical scenario and presented an experiment with our robot following a user around our laboratory for a certain time period.

As a general outcome we can state that our method is capable of handling a typical scenario. The multiple target tracking approach allows differentiation between the user that has to be followed, and bystanders who are not immediately involved in the interaction of user and robot, but still need to be detected and included in navigation planning.

A couple of confusing situations occurred during our tests in the "real world". The analysis of these situations revealed a need for an improvement of the detection of people in cluttered environments. In general it might be helpful to investigate the effect of a context dependent tracking model that includes knowledge about the environment, which requires map information. This and the fact that our approach is still valid for the purpose of person following indicate that it is useful to continue with the next steps toward a Human Augmented Mapping system and to work on improvements of the tracking system iteratively. These next steps involve the integration of a mapping method together with some interactive means of labelling the achieved map. This is the main subject of our current work.

References

[1] A. Green, H. Hüttenrauch, K. Severinson-Eklundh, "Applying the Wizard-of-Oz Framework to Cooperative Service Discovery and Configuration", in Proc. of IEEE RO-MAN, 2004.

[2] H. Hüttenrauch, and K. Severinson-Eklundh, "Fetch-and-carry with CERO: Observations from a long term user study with a service robot", in Proc. of IEEE RO-MAN, 2002.

[3] M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lömker, G.A. Fink, and G. Sagerer, "Person Tracking with a Mobile Robot based on Multi-Modal Anchoring", in Proc. of IEEE RO-MAN, 2002.

[4] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association", in Proc. of IEEE ICRA, 2001.

[5] E.A. Topp, D. Kragic, P. Jensfelt, H.I. Christensen, "An Interactive Interface for Service Robots", in Proc. of IEEE ICRA 2004.

[6] E.A. Topp, H.I. Christensen, "Tracking for Following and Passing Persons", in Proc. of IEEE IROS, 2005, *to appear*.

Automatic Building Detection for Mobile Robot Mapping *

Martin Persson and Tom Duckett
Centre for Applied Autonomous Sensor Systems
Örebro University, SE-701 82 Örebro, Sweden
{martin.persson, tom.duckett}@tech.oru.se

I. Introduction

Our research interest is within semi-autonomous mapping, SAM, in an outdoor environment using a mobile robot. In SAM prior information is used to enhance the mapping process. This information could be extracted from a satellite image or a hand-drawn map. Our scenario is an autonomous robot working in an urban area that needs to be mapped. In this process building detection has high priority. We have previously investigated automatic building detection from aerial images [1]. The experience from that work was that obtaining good results from aerial images is very hard, especially when digital elevation data is not available. In order to improve this automatic detection, information from mobile robot sensors can be used. This can, for example, include recognition of building facades. The idea is that a laser range scanner detects possible buildings, which are verified with building detection in images produced by a vision sensor.

Methods for automatic recognition of buildings have been studied before. Fritz et al. presented a new method for recognition of buildings [2]. They used 20 geo-referenced images of building facades from the city of Graz. The objective was to recognize the correct building from a photo taken by, for example, a camera in a mobile phone. The method is based on local features and entropy information in imagettes. Geo-referencing is used to limit the search space. By knowing where the image of the object to be recognized is located, the number of possible solutions in the database can be heavily reduced.

Another method, developed for building recognition in CBIR, Content-Based Image Retrieval, uses consistent line clusters with different properties [3]. CBIR is used for automatic classification of images on the Internet, where search engines can find images with relevant content based on recognition.

This paper discusses the problem of building detection in monocular images and gives some preliminary results. Two methods based on different feature extraction strategies are presented. The first uses invariant local features in the images and the second classifies images based on the image edge properties. Two classes of images are used; buildings and nature.

II. Classification Based on Invariant Features

In this section a method using invariant local features calculated with the SIFT algorithm [4] is presented. These features, also called keypoint descriptors, are 128 value arrays. When keypoints have been calculated, they can be matched to a keypoint database. The best match is found by minimizing the Euclidean distance to the other keypoints. In this matching known objects can be found in images. Our strategy is however different. We train an ensemble of SOMs [5], ESOM, with the different keypoint types and use the ESOM for discrimination between nature and building images.

A. SIFT - Feature Matching

SIFT, Scale Invariant Feature Transform, first presented in 1999 [4], was developed and patented by David Lowe. The main characteristic of SIFT is that it uses a feature description that is invariant to scaling and rotation. It is also insensitive to changes in illumination and camera location. SIFT finds distinctive features in gray scale images. The number of features in an image can be high. In order to reduce the computational time, remove the least distinctive features, and avoid finding keypoints in clutter, it is recommended to down sample the image to a small size.

Each SIFT keypoint has four values for location; position in image (row and column), scale and orientation. The keypoint descriptor itself has 128 values. This figure comes from a 4×4 array of orientation histograms centered on the keypoint, where each histogram has 8 bins.

B. ESOM trained with SIFT keypoints

SOM, the Self-Organizing Map, was developed in the early 1980's by Teuvo Kohonen [5]. A SOM is an unsupervised neural network that can cluster data. A number of neurons are organized in a grid and the map is trained with input vectors. A SOM toolbox for Matlab has been developed at Helsinki University of Technology. This has been used in our experiments.

ESOM, an Ensemble of SOMs, can be used for classification. One SOM is trained for each data class and in the classification the SOM with the best match is the winner. One benefit of this method is that SOMs can be added or removed from the ensemble without retraining of the already existing ones. Our experience from ESOM is that it is preferable to use the same grid size for all SOMs in order to be able to compare the quantization errors.

The optimal map size has been determined as 20×20 by training and evaluation. With this size, the average of correctly classified building and nature images is maximized and false positives are minimized.

*This work is partially supported by The Swedish Defence Material Administration

Fig. 1. The images used for training. Buildings are marked with 1 and nature with 2.

Fig. 2. Classified images, where label 1 is a building, 2 is nature and 0 is Unknown.

SIFT finds a number of keypoints in our low resolution images (120×90 pixels). On average 111 keypoints were found with the number of keypoints in the interval of 24 to 263. Evaluating the best matching unit gives a quantization error for each individual keypoint. These quantization errors are then averaged for both SOMs and the class with the lowest error was chosen as the winner. A threshold of 0.01 was used to check whether the ratio between nature and building is lower than 1%. In that case the image was classified as *Unknown*.

C. Result

Our data set consists of 30 nature images and 32 images of buildings. Of them, 10+10 images were randomly selected for training. The training images of one run are shown in Figure 1. The result of the classification of that run is presented in Figure 2. Running 9 batches, on average 93% of the building images and 84% of the nature images were correctly classified with 10% and 0.5% false positives respectively.

III. Classification Based on Edge Orientation

One typical feature of buildings in images is that, in many cases, they show mainly horizontal and vertical edges. In nature, on the other hand, edges tend to have more randomly distributed orientations, with the exception of bare tree trunks. Inspection of histograms based on edge orientation confirms this observation. Experiments that show the potential of classifiers based on edge orientation have been performed. Two classification methods have been evaluated, MDC and ESOM.

We use the same method for finding and labeling edges as in [1] and then calculate the absolute value of the line orientations. Evaluating the histograms of the line orientation from nature and building images, significant peaks around 0 and 90 degrees are detected for the buildings. Therefore, for each image, we calculate a histogram of the line's orientation. Different edges of the histogram bins have been tested, e.g.: $[0\ \ 0.15\ \ 1.42\ \ 1.57]$ and $[0\ 0.2\ \ldots\ 1.4\ 1.6]$ radians. Each bin contains the number of lines with a certain orientation. To be able to compare this statistic between images with a different number of lines, each bin value is divided by the total number of lines in the image, giving a relative value of the number of lines.

A minimum distance classifier, MDC, has been used to classify the data. The prototype vector is calculated by taking the mean values of the relative number of hits in selected bins of the histogram. A distance value $D_j(x)$ for each class j (building and nature) is calculated as $D_j(x) = \|x - m_j\|$ and the shortest distance indicates the winning class.

An alternative to MDC is to train an ESOM with input vectors based on the relative histogram values. We have used a grid size of 2×2 and classification is based on the quantization error.

A. Result

The used images are the same as those used in Section II. We used 10+10 images for training. The two methods were tested with the remaining 22 building images and 10 nature images, in total 42 images. Both methods classified on average 88% of the images correctly.

References

[1] Martin Persson, Mats Sandvall, and Tom Duckett. Automatic building detection from aerial images for mobile robot mapping. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Espoo, Finland, June 2005.

[2] Gerald Fritz, Christin Seifert, and Lucas Paletta. Urban object recognition from informative local features. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 132–138, Barcelona, Spain, April 2005.

[3] Yi Li and Linda G. Shapiro. Consistent line clusters for building recognition in CBIR. In *Proceedings International Conference on Pattern Recognition*, volume 3, pages 952–957, Quebec City, Quebec, CA, Aug 2002.

[4] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, Sep 1999.

[5] Teuvo Kohonen. *Self-Organizing Maps*. Springer, third edition, 2001.

Closed Contour Reconstruction using Iterated Smoothing Splines

June 30, 2005

Maja Karasalo, Xiaoming Hu	and	Clyde F. Martin
Optimization and Systems Theory		*Mathematics and Statistics*
Royal Institute of Technology		*Texas Tech University*
{*karasalo, hu*}*@math.kth.se*		*clyde.f.martin@ttu.edu*

Abstract

When building a map of an unknown environment several constraints must be met. If the environment is largely unknown we can only assume local information, and navigating and mapping must be based on sensor data alone. In this paper we consider the problem of sevoing around and mapping objects of unknown shape using splines. Interpolating splines for measurement data will give a poor result if the data is noisy, as the resulting curve will go through every measurement point. It has been shown in [1] that smoothing splines, where the curve is found through minimizing a cost function, is better for noisy measurements. In this paper we assume that we can travel several times around the object, making new measurements for every revolution. This allows for an iterated smoothing spline approach.

We are given a closed curve of unknown shape, for instance representing an unknown object in some environment that we want to map. Our aim is to find the shape of this curve, using only local information and range sensors with measurement errors. Each set of measurements is obtained by making one complete revolution. The servoing can be implemented for instance as in [2], using a planned path approach. Let

$$D_k = \{(\theta_i, r_i) : i = 1, ...N\}$$

be the measurement data obtained during revolution k. Using the spline generator

$$\begin{cases} \dot{x_1} &= x_2 \\ \dot{x_2} &= u \\ r(\theta) &= x_1(\theta) \end{cases} \tag{1}$$

the smoothing spline is given by the solution to the polar second derivative L_2 smoothing problem:

$$\begin{array}{ll} \min & J(u, x_0) = \int_0^{2\pi} u^2 d\theta + \delta^2 x_{20} + \epsilon^2 \sum_{i=1}^{N} w_i (r(\theta_i) - r_i)^2 \\ \text{s.t.} & u \in L_2[0, 2\pi] \\ & x(0) = x(2\pi) = x_0 \end{array} \tag{2}$$

Denote by $S_k(\theta)$ the spline given from solving Equation 2 using data obtained from k revolutions.

Solving Equation 2 for $\tilde{x}_1 = \tilde{r} = r_{k+1} - r_k$ will instead give iterated smoothing splines $r_{k+1}(\theta) = r_k(\theta) + \tilde{r}$. In this paper we will study the convergence of these iterated curves as $k \to \infty$, and evaluate whether they make for a good approximation of the true curve $S(\theta)$.

Using this approach, the new path for servoing will be a better fit to the closed curve for each new revolution, resulting in better measurements. Simulations to investigate performance of the approach will be presented at the workshop.

References

[1] Magnus Egerstedt and Clyde Martin, "Statistical Estimates for Generalized Splines", ESAIM:Control, Optimisation and Calculus of Variations, August 2003, Vol. 9, pp. 553-562.

[2] M. Karasalo, L. Johansson, X. Hu, K, Johansson, "Multi-Robot Terrain Servoing with Proximity Sensors", Proc. of ICRA 2005.

Real–Time Hough using Reconfigruable Hardware

Lars Asplund and Henrik Bergkvist and Isak Savo

Mälardalen University

1 Introduction

Many algorithms used in digital signal processing for image analysis require large computational resources. The most commonly approach is to use a DSP (Digital Signal Processor), such as Texas Instruments TMS320C6201 and C6701, Philips TriMedia TM1100 and Analog Devices Sharc ADSP 21160M. These processors are variations of SIMD architectures, and they contain several processing units. By the internal pipelining and by using the processing units in an optimal way quite high throughputs can be achieved. Standard PCs are naturally used for image analysis, but for real-time applications these systems has in the past not been powerful enough.

There have also been attempts to use reconfigurable hardware for Image Processing, [1], [4] and [2].

The current project aims at building a vision system (a vision system using image analysis in configurable hardware – FPGA, Field Programmable Gate Array) in the new robot design, Aros. ChipVision will analyze the output from four digital cameras in order to find lines in the environment. The output from the system will be data to the main computer in the robot about distance and angles to found lines.

A system utilizing reconfigurable hardware of 8 million gates and four CMOS cameras is used in an image analysis system. The system is a part of a sensor system for a robot, and can deliver data about the robots position using the Hough transform in real time.

The size of an FPGA today is far above several million gates. This amount of gates allow the full system to be implemented on one FPGA. This can be compared to the ARDOISE system [3], which is a modular system based on smaller FPGA's.

2 Overview of algorithms

The following steps are the main algorithms of the image analysis. First the RGB representation is transferred into HSI representation. One reason for this is that the analysis should not be sensitive for differences in light or shading effects. The HSI representation includes a separate variable/channel for intensity I. H is the variable which represents color tone. H is usually represented by a disc with 0-255 levels . By this representation the color is described with in one dimension instead of three. From this representation the interesting objects could be thresholded out using their color values.

Every pixel is given a number indicating their class, depending on their class and the class of the neighbors the pixels are labeled. Noise reduction is performed to minimize the number of pixels with erroneous object classification.

Mathematical formulas for transformation from RGB to HSI. These formulas take for granted that the RGB values are normalized to [0, 1].

For some application, the Saturation is not required since the colors are well separated in the Hue-circle. Thus only the H channel (and maybe the I channel) are relevant for finding interesting objects. S never needs to be calculated. H is a disc with radius 0-360 and can be represented as 256 gray levels (8 bits).

The segmentation will take place concurrently as the calculation of H.

Each pixel is now represented by its class ID. There are eight different color classes and one background class. These can be represented by 4 bits. What is left is a picture matrix with 16 gray levels.

This algorithm describes how noise reduction is performed. This step is performed once the segmentation is completed. A median filter sets a pixels class value based on its neighbor's class values. Of this follows that the image will be more homogeneous in terms of object classes. This since erroneous pixels will be adjusted. If this step is performed, over segmentation, which would complicate the classification, will be avoided.

For each pixel, look in a $n \times n$ neighborhood and assign the current pixel the group belonging of which most of its neighbors belong to.

So, given a pixels x and y coordinate, loop around its neighbors and let counters describe the number of pixels belonging to each class. Set the current pixels class belonging to the class which has the largest number of members in the current surrounding.

The algorithm for the edge detection is quite simple. Adjacent pixels both row–wise and column–wise are compared. Since the output from the Noise Reduction stage is very distinct, it is enough to set a pixel as member of the Edge–memory if one of the pixels belongs to the green carpet and the other belongs to a white line.

After a pixel has been assigned as being an edge–pixel its coordinate is justified due to the Fish–Eye characteristics of the camera used. The cameras are quite simple and cheap cameras, and the lens is far from linear. The algorithm used for this unfishify is by linear shifts in the X and the Y coordinate. This is not totally optimal, but is adequate for our purposes.

All stages for each camera from the input stage to the Unifishify handles all pixels at the full pixel–rate, and the picture is never stored. After the Unfishify–stage the edges are stored in Block–RAM, which is an on–chip memory.

The demand on the memory space for the Hough transform is high if a high resolution in angle and distance are required. Our approach to this is to do the transformation in two steps. In the first step, a lower resolution is used in both θ and ρ.

After the number of edge–pixels have acquired (the camera is turned up–side–down and all interesting lines appear close to the robot first) or when end–of–frame is reached, the Hough–space is searched. In this search *areas of interests* are defined, and a peak in Hough–space will be used to eliminate other peaks in the vicinity. The number of *areas of interest* are maximized to 32. The 32 possible *areas of interest* are filtered down to a maximum of six lines. After all *areas of interests* have been defined the second round of Hough transformations are performed. In this search an area of 15 degrees for θ and 12 units for ρ are allowed, giving a size of the Hough–space of only 180 points.

All edge–pixels are tested against all *areas of interest*. A final search in each of the *areas of interest*. The communication between the mother–board holding the FPGA and the four cameras are either by means of a serial link or using an USB–controller (FT245BM), with a bandwidth of 1 MByte/s.

3 Results and Discussion

For any kind of robot the sensor system is crucial for its observation of the environment. Of various sensors, vision is the most powerful. The main way vision is implemented today is to use ordinary computers. Although a modern PC has very high performance there is always a trade–off between frame–rate and resolution.

The results from this study shows that; by using an FPGA, it is possible to achieve a frame–rate of 25Hz in a setup with four cameras, where up to six lines are detected by each Camera. There are plans of implementing a SLAM system based on the found lines. Currently tests are performed in a well–defined environment.

Future improvements include improved algorithm for unfishifying the image. There are also improvements in detecting lines in the first stage of the Hough transform. Also the second stage can be further parallelized.

4 Acknowledgments

A special thanks to Xilinx Inc., for their support in terms of circuits.

References

[1] L Asplund, H Bergkvist and T Nordh, "ChipVision2 – A Stereo Vision System for Robots Based on Reconfigurable Hardware", to appear at RoboCup International Symposium 2005, Osaka, Japan.

[2] K Benkrid, D Crookes, J Smith and A Benkrid, "High Level Programming for Real Time FPGA Based Video Processing", IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM2001, April 2001.

[3] D Demigny, L Kessal, R Bourgiba, N Boudouani, "How to Use High Speed Reconfigurable FPGA for Real Time Image Processing", Fifth IEEE International Workshop on Computer Architectures for Machine Perception (CAMP'00), 2000.

[4] L Petit and J.D. Legat, "Hardware Techniques for the Real-Time Implementation of Image Processing Algorithms", Microelectronics Laboratory, University Catholique de Louvian, Belgium.

A Comparative Study for WiFi Localization in a Dynamic Environment

Boris Durán.
Intelligent Systems Lab, IDE, Halmstad University.
{boris@yodir.com}

Extended Abstract

Self-localization in mobile vehicles has been in research for several years. Many different approaches have been used, testing not only different kind of sensors but also different kind of algorithms; and trying always to be as precise as possible. WiFi localization is a new approach for solving this old problem. Not only private and public institutions but also home users are adopting, in one way or another, wireless technology. This is precisely the main reason for using, somehow, the already installed wireless hardware as a sensor system for localization; in that way, reducing costs and complexity of the final structures.

This research starts as a new effort in solving the WiFi localization problem by the systematic acquisition of real measurements wireless access points in a semi-controlled environment. Most of the research in WiFi Localization has focused on reading the Received Strength of the Signal Indicator (RSSI) from different base stations in order to develop localization methods, which have been based mostly on Bayesian algorithms. All efforts for minimizing the average of errors in WiFi localization have been reported to be around 3 meters and 97% of the cases around 9 meters. Moreover, most of the experiments have been developed under specific constraints and without considering the effects of obstacles in the overall performance of the algorithms.

The collected datasets would be used to test the performance of the Transition Matrix algorithm, which is a probabilistic method for localization that uses the occurrence of the transition of two signal strength readings, instead of using the readings themselves. Probability Density Functions are created by analyzing all the transitions produced by consecutive readings in a reference map. Once this matrix has been created, the algorithm follows the same methodology as in the Bayesian case; the position in which the system believes to be is given by the maximum likelihood estimator.

This work also presents some alternative approaches, non linear techniques based on Neural Networks and Fuzzy Logic which earlier have not been studied in WiFi localization. Learning Vector Quantization Networks and Self Organizing Maps from Neural Networks, together with Subtractive Clustering from Fuzzy Logic are the AI techniques studied in this research. The possibility of identifying natural groupings within the input data and produce a concise representation of a system's behavior was the main reason for testing these methods in WiFi localization.

All experiments were done in one of the corridors of the third floor of D building at Halmstad University. A distance of 15 meters was used within this corridor as the experimental space. A grid of 29 positions was created by collecting 256 signal strength measurements every 0.5 meters. Even though all methods were tested using only two Access Points, increasing their number suggests that the accuracy could also be increased; thus, extending the research to two and three dimensions.

The similarity of the mean errors, using Bayesian Inference, obtained in this project and the ones reported by other papers validate the methodology and results of this research. Table 1 brings together the medians, means and standard deviations of all the methods used in this project. Transition Matrix localization showed approximately the same mean errors as Bayesian Inference localization. However, the Transition Matrix algorithm showed the same results in a time almost 30 times faster than the Bayesian algorithm.

Method	**Obstacle-free**			**Obstacle at 10 m**		
	Median	**Mean**	**Std**	**Median**	**Mean**	**Std**
Bayesian	2.50	2.862	1.863	2.10	2.464	2.123
Trans. Matrix	2.55	2.867	1.835	2.00	2.462	2.085
LVQ	1.55	2.188	1.952	2.45	3.026	2.413
SOM	1.80	2.181	1.655	1.00	1.667	1.517
Fuzzy Clusters	**1.469**	**1.765**	**1.199**	**0.967**	**1.341**	**1.214**
Bayes, RADAR, Other probabilistic approaches	Mean error: ~ 3 meters 97%: ~ 9 meters.					
Signal propagation model and Extended Kalman Filter	Mean error: 2.60 meters.					
RF propagation model and location fingerprinting	Mean error: 5 – 7 meters.					

Table 1. Comparison table for WiFi localization. [*All values are in meters*]

The improvement obtained when using Neural Networks and Fuzzy Logic is evident. Fuzzy Clustering decreases the mean errors in more than 1 meter compared to the Bayesian based methods. Moreover, in 90% of the cases, errors were around 3 meters either with or without obstacles, see Figure 1. This is definitively one step up when comparing to other reported mean errors, see Table 1. It is also important to notice the possibility of using the median values instead of the means since the first ones give a better approximation in all cases.

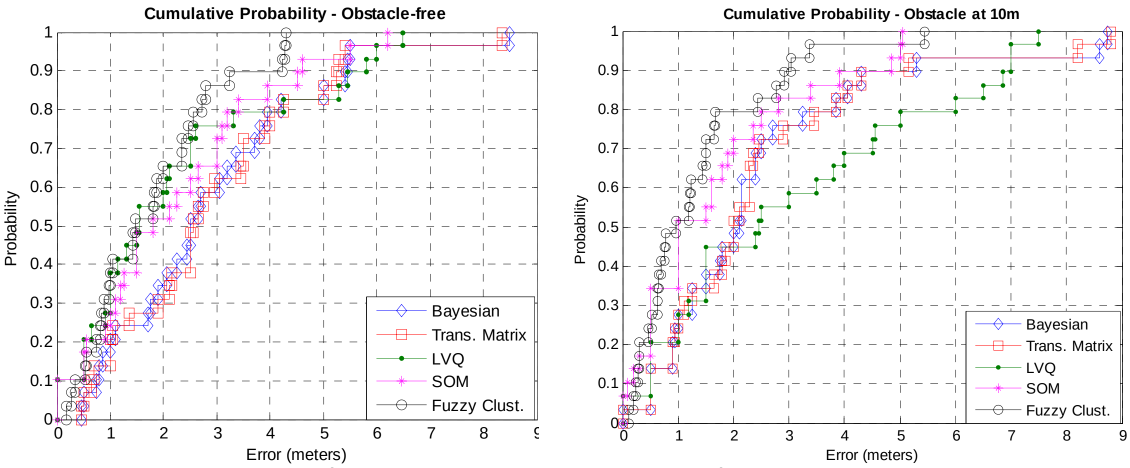


Figure 1. CDF for all methods without obstacles (left) and with an obstacle (right).

All goals proposed at the beginning of the project were accomplished after a careful and intense study of hardware and software in order to guarantee the reliability of the datasets. The analysis of localization mean errors from the Bayesian and Transition Matrix algorithms validated the results reported by other researches. They also showed the difficulties presented when trying to model a localization system given only WiFi radio signals. Therefore, new approaches capable of dealing with such non linear systems were studied and tested. Neither Neural Networks nor Fuzzy Logic has earlier been applied to WiFi localization. All of them showed improvements being Fuzzy Clustering the one with the best approximations. The preliminary results shown here are promising and encourage further research using AI techniques for WiFi Localization.

The continuation of this work will focus primarily on collecting measurements in two and three dimensions, both indoors and outdoors. Danaher Motion Systems is one of the private companies interested in the results of this research since some of their projects involve WiFi localization in both indoors and outdoors environments.

Fault detection for increased roubustness in navigation

Paul Sundvall and Patric Jensfelt
Centre for Autonomous Systems, KTH

Abstract— **This paper describes a method for detecting unexpected events during navigation such as collisions and wheel slippage. Some of these events are hard to detect but can be devastating for the navigation performance. Furthermore, when collisions go undetected it can lead to damage or injury.**

The method uses two or more sources of position information that are fed into an extended Kalman filter tracking the position. A CUSUM test is used to detect filter divergence and triggers an alarm upon unexpected events. Experimental results based on position information from odometry and a scan matching routine are presented. It is demonstrated that the system can detect that the robot is being pushed.

I. Introduction

For service robots, a crucial part of the system is to navigate reliably so as to avoid running into things. Also, if a collision occurs, the robot should try to minimize the damage, which makes it important to detect these events in a robust manner. Especially for applications in a domestic setting, users may not be supervising or may not even be able to help the robot if such a situation occurs. Most systems work well under normal operating conditions, but their performance degrade significantly upon unexpected events.

There are several things that can make the navigation system fail, e.g. collisions or sliding against something which makes the robot rotate, slippage when running over a cable or a threshold, or because users push the robot. An example of a situation where the navigation will fail is shown in Figure 1. In this case, there are no sensors on the robot that can detect that the top of the robot is in contact with the table and the robot is so light that the wheels are spinning on the floor. To decrease the sensitivity to such faults, one can use extra sensors like a gyro, see e.g. [1]. In the present paper, the approach is to use any two (redundant) methods for localisation, and merge their outputs to detect deviation. It is demonstrated that this strategy works, using odometry and scan matching.

Fig. 1. "Goofy" in collision (see top right corner) with a table. This is not observable by the laser scanner (blue), sonars or the bumper switches since they are vertically displaced compared to the table.

II. Approach

The method requires two or more measurement systems delivering estimates of the robot position ("pose provider"). These estimates need not be in the same coordinate system. The term pose provider will be used rather than measurement or sensor, because it can represent outputs that are more refined than raw sensor data. One can think of having multiple pose providers as having several GPS systems with different coordinate system and characteristics.

The role of the pose provider is to provide information of the motion of the robot. It is not necessary to provide an absolute position - this means that methods with drift, e.g. inertial systems, odometry or integrating relative motion, can be used, as well as absolute[1] pose providers like SLAM or map based navigation.

The results from the pose providers are fed into a filter that tracks the poses and decides if the poses are changing unexpectedly. The filter assumes all measurements follow the same underlying control signal, namely the robot speed. The difference between the poses will change slowly in the normal case. On collision, slip or similar event which causes the pose providers to disagree, this difference will change abruptly.

It is important that the pose providers provide redundancy. Many navigation modules assume that the odometry is quite reliable and will fall back to odometry when the sensor readings do not match. If such a pose provider is used, and one uses odometry as a complementary pose provider, they will behave similarly in the case of a large slip, and there will be no redundancy left for the filter to exploit.

The filter uses a set of three states $r_i = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$ for every pose provider i. As measurements (reports from the pose providers) come in, the state estimate is updated.

The purpose of the filter is not to track the state, but rather to detect abrupt changes in it. An extended Kalman filter is used to monitor the outputs of the pose providers. The innovations of the filter are fed to a detector, which serves the purpose of removing short transients to avoid false alarms.

A. Model and state tracking

A standard dynamic model is used for the evolvement of the output from the pose providers:

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t(x_t) u_t \quad & cov(u_t) = Q_t(x_t) \\ y_t &= C_t x_t + v_t & cov(v_t) = R_t \end{aligned} \tag{1}$$

where x is the aggregated state vector $\begin{bmatrix} r_1^T & r_2^T \end{bmatrix}^T$ in the case of two pose providers. The state is assumed to evolve only depending on the control signal and process noise u_t, by setting $A = I$. Since the measurements are the readings of the pose providers, $C = I$. The noise covariance matrices Q and R are state dependent[2], reflecting the fact that the uncertainty in robot motion will be large when the robot moves and small when standing still.

Because of the state dependent noise intensities and non-linear relation between input u_t and state x_t, the standard Kalman filter cannot be used. Instead the equation has to be linearized, around the current heading angle, $\theta(t)$, of the respective pose in the state vector. The input u_t and measurement noise v_t are assumed to be Gaussian and independent. The drift model is defined by B_t and Q_t as follows:

$$B_t = \begin{bmatrix} B^1 & B^1 & 0 & I_3 & 0 \\ B^2 & 0 & B^2 & 0 & I_3 \end{bmatrix} B^i = \begin{bmatrix} \Delta T \cos(\theta_i) & 0 \\ \Delta T \sin(\theta_i) & 0 \\ 0 & \Delta T \end{bmatrix} \tag{2}$$

[1]in the sense that the position error does not grow over time

[2]In the implementation Q is scaled to an estimate of the speed of the robot, using odometric information.

$$u = \begin{bmatrix} u^c \\ u^{e,1} \\ u^{e,2} \\ u^{c,1} \\ u^{c,2} \end{bmatrix} Q = \begin{bmatrix} Q^c & & & & \\ & Q^{e,1} & & & \\ & & Q^{e,2} & & \\ & & & Q^{c,1} & \\ & & & & Q^{c,2} \end{bmatrix} \quad (3)$$

where θ_i is the heading angle, ΔT is the time step and u^c is the common control signal $\begin{bmatrix} u_x & u_\omega \end{bmatrix}^T$. The drift $u^{e,i}$ is modelled as an additive input on the control signal. $u^{c,i}$ represents noise that models a drift in all states simultaneously, and has the effect of alleviating problems arising from linearization. There are in total 12 inputs for the model with two pose providers. It is straight forward to add more pose providers, at the cost of expanding the state vector. The process noise intensity Q is chosen diagonal, where the elements referring to u^c are scaled to the current speed of the robot.

The measurement noise intensity R is selected to be diagonal. Since the filter is implemented in a non realtime system, there will be a scheduling jitter that causes the readings to not be on distinct time instants. Even if the results from the pose providers is read within machine precision, the measurement noise is set to a nonzero value to represent the misreadings arising from scheduling jitter. A reasonable value for the measurement noise intensity is given by multiplying a small constant with the distance the robot travels during one time step, running at nominal speed.

The noise intensities can initially be set by studying the output of the pose providers that are used. An estimate of the robot speed can be compared for the different models, while running the robot and while at standstill. To increase the performance of the tracking, the parameters have to be tuned to measured data. This can be done by looking at the filter residuals for fault free and faulty data. The parameters are then adjusted to give high residuals for the faulty data and small residuals for the fault free data.

B. Detector

The output from the Kalman filter is an estimate $\hat{x}$ and an associated covariance matrix P. The prediction error is $e = y - Cx$, which is Gaussian and white, given that the model assumptions hold. If this is the case, the Mahalanobis distance $s_t = e^T(CPC^T + R)^{-1}e$ will be χ^2 distributed when there is no fault, and large otherwise. The measure is scalar, making it easier to handle than the raw residual.

A simple test of when to alarm for faults is to set a threshold on s_t. However, this is sensitive to rapid variations in s_t, which would cause false alarms. To avoid this, s_t is fed into a detector that add a bit of slowness to the detection. Here, the CUSUM test (see e.g. [2]) is used. There are two parameters to tune in the test, drift $v > 0$ and alarm threshold $h > v$. A simplified description is that v is related to what level the input to the detector normally has, and h is adjusted to trade off false alarms and risk of missed detection. Since the Kalman filter after a disturbance will slowly adapt to the new data, faults will only give residuals under a limited time. Therefore, the detector must not be too slow. As for the parameters of the Kalman filter, the parameters have to be tuned for the CUSUM test as well. This is done by studying the behaviour of the detector using fault free and faulty input respectively.

III. Implementation and experimental results

The filter is implemented on "Dumbo", a robot equipped with odometry and a SICK laser scanner. Since at least two pose providers are needed, a memoryless scan matching algorithm is implemented, inspired by [3]. The scan matcher integrates the relative movement found between consecutive scans, and the result is fed into the filter.

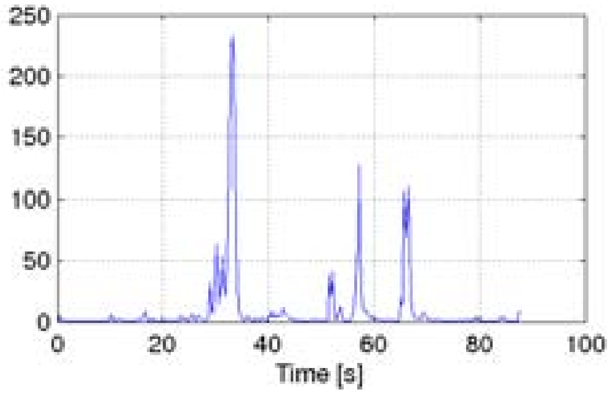


Fig. 2. Weighted residuals s_t from the Kalman filter from test with robot being pushed several times. The pushes are clearly visible as peaks in s_t. The first 25 seconds are fault free, and the residual is small.

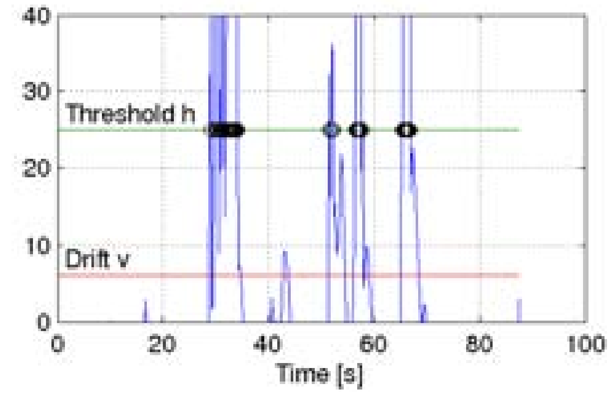


Fig. 3. Output from CUSUM test with robot being pushed several times. When the test statistic g_t exceeds the threshold, an alarm is generated (marked with circles) and it is reset. The weighted residual from Figure 2 is used as input for the detector.

The filter has successfully been tested during several sessions, with both fault free and faulty data. During the experiments, the robot has moved around autonomously using Nearness Diagram [4]. An example of output from an experiment with faulty data is shown here, where the robot has been pushed. The Mahalanobis distance s_t from the Kalman filter is shown in Figure 2. One can clearly see when the robot has been pushed. The corresponding test statistic and its threshold is shown in Figure 3. Each time the test statistic exceeds the threshold, g_t is reset to zero and an alarm is raised. The alarm times are marked with circles in the figure.

The left part of the data shows the no fault condition, and it is seen that the residual s_t is small. When the robot is pushed, the residual gets larger and the CUSUM test triggers an alarm.

References

[1] B. J. and L. Feng, "Gyrodometry: A new method for combining data from gyros and odometry in mobile robots," in *International Conference on Robotics and Automation (ICRA)*, 1996.

[2] F. Gustafsson, *Adaptive filtering and change detection*. Wiley, September 2000.

[3] C. Früh and A. Zakhor, "Fast 3D model generation in urban environments," in *Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, 2001.

[4] J. Minguez and L. Montano, "Nearness diagram navigation(ND): A new real-time collision avoidance approach," in *International Conference on Intelligent Robots and Systems (IROS)*, 2000.

Fast Laser Based Feature Recognition

Johan Larsson
AASS, Örebro University, Örebro, Sweden
johan.larsson@tech.oru.se

Mathias Broxvall
AASS, Örebro University, Örebro, Sweden
mbl@aass.oru.se

I. Introduction

Today the use of mobile robots is accepted as a part of everyday life. Autonomous vaccum cleaners and lawnmowers are examples of products which are available for home use. In industry, large efforts are made to replace manually operated vehicles in dull or hazardous applications with autonomous or teleoperated vehicles. One example of an industry application is the LHD (Load-Haul-Dump) vehicles, which are typically used in mines to transport ore from the stope or muck-pile to a dumping point. The work presented here is a part of an effort to develop algorithms for a control system based on reactive navigation to be used in a LHD vehicle.

The requirements for navigating the tunnels of a mine are basically the same as the requirements for navigating the corridors of an office building. First, the robot has to be able to locate the walls of the corridor that is currently being traversed. Second, intersections or doors have to be identified and their location relative to the robot have to be established. The first requirement has to be fulfilled for the robot to be able to successfully follow a corridor, the second is crucial for the localisation and the ability of the robot to navigate to a particular goal. In this paper we present our methods for feature recognition used for high speed reactive navigation based on a topological map, with none or sparse metric information. Our methods for corridor and intersection detection yields execution times of only a fraction of what we have found previously described in the relevant literature, and have proven robust and reliable in experiments performed both in office environment and in our test mine [Larsson et al., 2005].

A. Related Work

Our Corridor detection algorithm is based on the Hough Transform, a method proposed in 1962 by Paul V.C. Hough [Hough, 1962]. The Hough Transform is a robust and effective method for finding lines in a set of 2D points, and is a commonly used method in image processing [Duda and Hart, 1972]. In mobile robotics the Hough Transform has been widely used both for vision based self localisation, see for instance [Iocchi and Nardi, 1999], and mapping based on laser range data [Giesler et al., 1998]. In [Barber et al., 2001] a complete control system for topological navigation is described. The system uses a Hough Transform to detect lines segments, and from those segments corners and doors are extracted. However, in the experiments described the robot travels at 0.2 m/s, a speed that does not require a fast control loop and thus any high speed feature recognition. The average time to invoke the detection service and get back the detection data in the distributed system running on 400 MHz AMD K6-II processors under Linux are several hundreds of milliseconds, which is far too long to allow high speed navigation.

In [Alempijevic, 2004] a new innovative method to extract lines directly from the range data of the laser scanner is presented. Although this method is fast due to the fact that the line detection is done in sensor coordinates, it has the drawback that to be able to detect a line the laser points on that line have to be consecutive. The execution times of this method is in the order of tens of milliseconds, implemented in Matlab and compiled using the Matlab C compiler.

II. Feature recognition algorithms

Since the intention is to be able to run the feature recognition algorithms on an embedded system controlling a vehicle at high speed, it is most crucial that the algorithms are effective and computationally cheap, but still robust and reliable in all situations. This have been achieved by using a modified Hough Transformation to detect corridors, and by detecting doors or intersections in a combination of sensor space and Cartesian coordinates. To be able to reach the short execution times, both the corridor and the door/intersection detection use the fact that the density of the sensor readings is directly relative to the measured distance, this enables us to exclude a vast amount of data points without losing any information. In our experiments we have achieved execution times for the feature recognition of 1–3 ms on a 1.7 GHz Pentium M running Linux. Figure 1 gives an example of the extraction of the edges and direction of a corridor from laser range data. The green line shows the centre and direction of the corridor while the blue lines indicates the location (dark) and direction (light) of two corridors intersecting the traversed one.

A. Corridor detection

In our experiments the Hough space of the corridor detection is discretized with the angular resolution 1°, and the radius in 100 steps, i.e. radius resolution is the desired detection-range/100. Since all the angles of the laser data and Hough Transform are known in advance, the sine and cosine functions of these angles are all precomputed and stored, and thus can be accessed by a single indexing operation. The Hough Transform is by far the most time consuming operation of our algorithm. To reduce its computation time, we have implemented a simple function to exclude redundant information: any laser point that is closer to its predecessor than the radius resolution is excluded. In indoor environments where we have used the detection range 10 m, this reduces the laser data points in the typical situation of Figure 1 to one third, giving a significant reduction of the computation time. The elimination of redundant information also gives

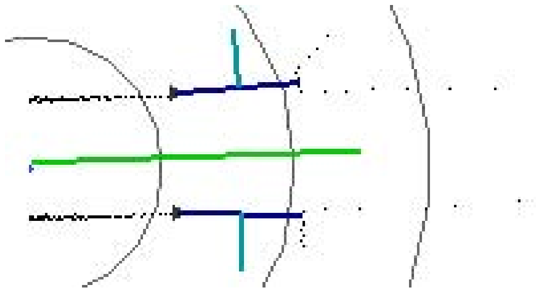


Fig. 1. Detected corridor line (green) and intersecting corridors (blue), robot (small blue triangle) on the left of the figure, heading right. Distances between scalelines (grey arcs) are 2.5 m.

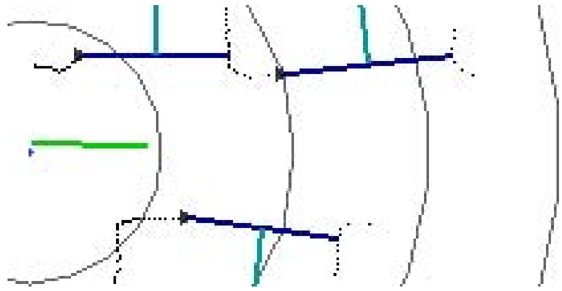


Fig. 2. Example of feature extraction in mine, the distances between scale lines (grey arcs) are 10 m. Tunnel width approximately 11 m.

another benefit, it eases up the identification of distant lines. By excluding laser points in close proximity to each other, the data point density is more evenly distributed and distant lines can be detected as well as lines close to the sensor.

The extraction of lines from the Hough space is implemented straight on as a search for the most intense candidates. These lines are then evaluated to find a line pair for which the directions differ approximately 180° and the distance between the lines is within the boundaries of the specified corridor width. The centre and direction of the corridor, relative the robot, are then easily calculated from the selected lines.

B. Intersection detection

In our application we use the opening detection to detect the presence and location of intersecting corridors/tunnels, but the algorithm can just as well be used to detect open doors. The algorithm is based on the fact that the presence of an intersecting corridor will result in a discontinuity in the laser range array, and that such an event therefore can be detected directly in sensor coordinates. By evaluating the difference in range of consecutive laser points (R_n, R_{n+1}) we have a sufficient condition to locate all potential intersection event candidates, but we can also get a large number of false positives. These are eliminated in two steps. First, the distance from the first point of the event is compared to all following points. If the distance in any case is shorter than our specified minimum opening the event can be dismissed. Second, the laser point (R_{n+x}) closest in euclidian space to the first point (R_n) of the event is located. If this point is identical to the initial second laser point (R_{n+1}) of the event (i.e. $x = 1$)), the event can be dismissed. If not, the two laser points R_n and R_{n+x} unambiguously reveal the presence and location of a corridor intersection.

For the algorithm to be robust there are a few other conditions that have to be considered as well, such as splitting the laser points in a left and right side. However, the detection of one or several intersections or doors can be performed fast with high reliability, and the execution time of the intersection detection is only fractions of a millisecond, well below 0.1 ms on our test system.

III. Experiments

The algorithms have been verified using our ATRV-Jr robot equipped with a SICK LMS 200 laser scanner. During the experiments the robot have successfully navigated at its full speed 1.7 m/s, both in the basement of the university building, and in our test mine. The only parameters that have to be changed between the different locations are the min and max width of expected corridors, and the detection range of the Hough Transform. Figure 2 shows tunnel and intersection detection in our test mine, note that the tunnel is correctly identified even though its walls are interupted by many side tunnels.

IV. Conclusion

The algorithms presented here have shown promising results to provide a robust and fast way to extract features from laser data for topological navigation of mobile robots. The functions have also proven to be flexible and easy to adapt to different environments such as mines tunnels and office corridors. For the future, we intend to evaluate the algorithms on a real LHD vehicle in the test mine.

Acknowledgment

This work is founded by the Swedish organisation Robotdalen. The authors would also like to thank Prof. Alessandro Saffiotti for support and advice.

References

[Alempijevic, 2004] Alempijevic, A. (2004). High-speed feature extraction in sensor coordinates for laser rangefinders. In *Proceedings of the 2004 Australasian Conference on Robotics and Automation*.

[Barber et al., 2001] Barber, R., Mata, M., Boada, M., Armingol, J., and Salichs, M. (2001). A perception system based on laser information for mobile robot topologic navigation. In *Proc. of IEEE International Conference on Robotics and Automation, Seoul, Korea*, pages 1702–1707.

[Duda and Hart, 1972] Duda, R. and Hart, P. (1972). Use of the hough transform to detect lines and curves in the pictures. *Communications of the ACM*, 15(1).

[Giesler et al., 1998] Giesler, B., Graf, R., Dillman, R., and Weiman, C. (1998). Fast mapping using the log-hough transformation. In *Proc. of IEEE/RSJ International conference on Intelligent Robots and Systems*, pages 1702–1707.

[Hough, 1962] Hough, P. (1962). Method and means for recognizing complex patterns, u.s. patent no. 3,069,654. Technical report, HelpMate Robotics Inc.

[Iocchi and Nardi, 1999] Iocchi, L. and Nardi, D. (1999). Hough trasform-based localization for mobile robots. In Mastorakis, N., editor, *Advances in Intelligent Systems and Computer Science*, pages 359–364. World Scientific Engineering Society.

[Larsson et al., 2005] Larsson, J., Broxvall, M., and Saffiotti, A. (2005). A navigation system for automated loaders in underground mines. In *Proc. of the 5th Int. Conf. on Field and Service Robotics (FSR-05)*, Port Douglas, Australia. Online at http://www.aass.oru.se/˜asaffio/.

Selection of Virtual Fixtures Based on Recognition of Motion Intention for Teleoperation Tasks

Daniel Aarno, Staffan Ekvall and Danica Kragić
Centre for Autonomous Systems
Computational Vision and Active Perception Laboratory
Royal Institute of Technology (KTH), Stockholm, Sweden
Email: {bishop, ekvall, danik}@nada.kth.se

I. Introduction

It has been demonstrated in a number of robotic areas how the use of *virtual fixtures* improves task performance both in terms of execution time and overall precision, [1]. However, the fixtures are typically inflexible, resulting in a degraded performance in cases of unexpected obstacles or incorrect fixture models. In this paper, we propose the use of *adaptive virtual fixtures* that enable us to cope with these problems. A teleoperative or human machine collaborative setting is assumed with the core idea of dividing the task, that the operator is executing, into several subtasks. The operator may remain in each of these subtasks as long as necessary and switch freely between them. Hence, rather than executing a predefined plan, the operator has the ability to avoid unforeseen obstacles and deviate from the model. In our system, the probability that the user is following a certain trajectory (subtask) is estimated and used to automatically adjust the compliance. Thus, an on-line decision of how to fixture the movement is provided.

In today's manufacturing industry, large portions of the operation has been automated. However, many processes are too difficult to automate and must rely on humans' supervisory control and decision making; in areas such as the identification of defective parts and process variations [2], or medical applications [3]. When such skills are required, humans still have to perform straining tasks. We believe that Human-Machine Collaborative Systems (HMCS) can be used to prevent ergonomic injuries and operator wear, by allowing cooperation between a human and a robotic system in a flexible way.

HMCS are gaining significant attention in areas such as medical surgery. In our previous work, [4], we investigated the possibility of segmenting complex tasks such as vitreo-retinal eye surgery into logical components or subtasks, given sensor traces of humans performing the task. It has been demonstrated that, if such segmentation can be obtained, it is possible to enhance the user's performance by choosing appropriate control modes and applying virtual fixtures. In [4], Hidden Markov Models (HMMs) were used for modeling subtasks where the states of the HMMs corresponded to motion primitives, or *gestemes*.

In our system, a high-level task is segmented to subtasks where each of the subtasks consists of moving a robotic manipulator along a straight line in 3D using a teleoperating device. Each state is associated with a virtual fixture automatically obtained from 3D training data. A state sequence analyzer learns what subtasks are more probable to follow each other. This is important for an on-line state estimator which estimates the probability of the user being in a particular state. A specific virtual fixture, corresponding to the most probable state, is then automatically applied.

II. Trajectory Analysis

This section describes the implementation of the virtual fixture learning system. The virtual fixtures are generated automatically from a number of demonstrated tasks. The overall task is decomposed into several subtasks, each with its own virtual fixture.

Virtual fixtures act as a guidance that aids the user in controlling the robot. The fixtures constrain the user's motion in undesired directions while allowing motion along desired directions.

When using adaptive virtual fixtures, it is important to detect the current state in order to apply the correct fixture. A combination of HMMs and SVMs (**S**upport **V**ector **M**achines) are used for state sequence analysis and probability estimation.

A HMM is a double stochastic model. An unobservable underlying stochastic model can be observed through a set of other stochastic models that generate a sequence of observations. The majority of applications have been in speech recognition [5] but successful results are also reported in many other fields.

One of the problems with using HMMs is the choice of the probability distribution for estimating the observation probability matrix B. With continuous input, a parametric distribution is often assumed because of issues related to the Baum-Welch method on data sets where $M \gg N$, [6]. This may decrease the performance of the HMM since the real distribution is hidden and the assumption of a parametric distribution is a strong hypothesis on the model [7].

Using probability estimators avoids this problem. These estimators compute the observation symbol probability [8] instead of using a look-up matrix or parametric model. In this work, SVMs are used to estimate the observation probabilities.

The first step is to filter the input data. Then, a line fitting step is used to estimate how many lines are required to represent the demonstrated trajectory. An observation probability function learns the probabilities of observing specific 3D-vectors when tracking a certain line. Finally, a state sequence analyzer learns what lines are more probable to follow each other. In summary, the demonstrated trajectories results in a number of support vectors, a HMM and a set of virtual fixtures. The support vectors and the HMM are then used to decide when to apply a certain fixture.

Once the task has been demonstrated, the filtered input data is quantized in order to find straight lines. Since the input data consists of normalized 3D-vectors, K-means clustering [9] may be used to find the lines. The position of a cluster center is equal to the direction of the corresponding line.

SVMs are used for observation probability estimation, instead of using the B matrix. The benefit is that the method does not require discrete observation symbols or an assumption on the probability distribution. SVMs are linear classifiers that separate data by a decision hyperplane in a high-dimensional space to which the data is transformed by applying a kernel function.

For each state detected by the clustering algorithm, a SVM is trained to distinguish it from all the others (one-vs-all). In order to provide a probability estimation for the HMM, the distance to the margin, from the sample to be evaluated is computed and transformed to a conditional probability using a sigmoid function [7]. The probability for a state i given a sample $\mathbf{x}$ can then be computed and by applying Bayes' rule, the HMM observation probability $P(\mathbf{x}|\text{state } i)$ may be computed. The SVMs now serve as probability estimators for both the HMM training and state estimation.

A fully connected HMM was used to model the task. The number of states is equal to the number of line types found in the training data.

With each line, there is an associated virtual fixture defined by the direction of the line. In order to apply the correct fixture, the current state has to be estimated. The system continuously updates the state probability vector $\mathbf{p}$, where $p_i = P(\mathbf{x_k}, \mathbf{x_{k-1}} \ldots, \mathbf{x_1}|\text{state } i)$ is calculated by the forward procedure from the HMM and the observation sequence so far. The state s with the highest probability p_s is chosen and the virtual fixture corresponding to this state is applied with the guidance $k = \max(0.5,\ p_s \cdot 0.8)$, where $p_s = \max_i\{p_i\}$. This automatic adjustment of the guidance factor allows the user to leave the fixture and move freely without having a special "not-following-fixture"-state.

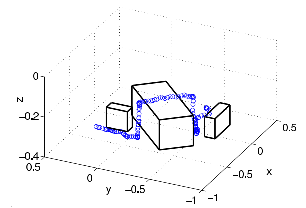
III. Summary and Experimental Evaluation

The system was evaluated with three experiments. The first experiment was a simple trajectory tracking task in a workspace with obstacles. The second was similar to the first, but the workspace was changed *after* training, in order to test the algorithm's automatic adjustment to similar workspaces. In the last experiment an obstacle was placed right in the path of the trajectory, forcing the operator to leave the fixture. This experiment tested the guidance adjustment as well as the algorithm's ability to cope with unexpected obstacles. A PUMA 560 robotic arm was used in all experiments.

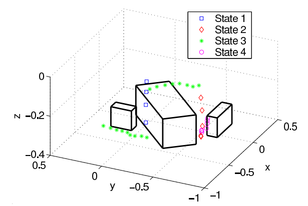
In the experiments, a magnetic tracker called Nest of Birds was used to control the PUMA 560. However, the system works equally well with other input modalities. For instance, we have also used a force sensor mounted on the end effector to control the robot.

A. Experiment 1: Pick and Place

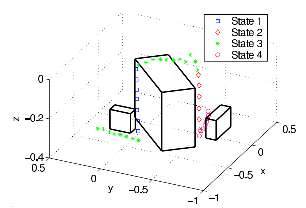
The first experiment was a simple pick-and-place task in a narrow workspace. The user had to avoid obstacles and move along certain lines to avoid collision. At start, the operator demonstrated the task five times. The system learned from training data, illustrated in Fig. 1(a). From the training data four states were automatically identified.



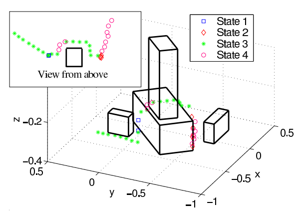
(a) A training example demonstrated by the user.



(b) Following trajectory using virtual fixtures.



(c) Modified workspace (obstacle size).



(d) Avoid obstacle not present during training.

Fig. 1. End effector position in example workspace. The different symbols (colors) corresponds to the different states recognized by the HMM.

The user then performed the task again, this time aided by the virtual fixtures generated from the training data. The path taken by the PUMA is shown in Fig. 1(b).

For clarity, we also present the state probabilities estimated by the SVM and HMM during task execution, in Fig. 2. The guidance factor used is also shown. This example clearly demonstrates the ability of the system to successfully segment and repeat the learned task, allowing a flexible state change.

B. Experiment 2: Changed Workspace

This experiment demonstrates the ability of the system to deal with a changed workspace. The same training trajectories as in the first experiment are used but the workspace was changed after training.

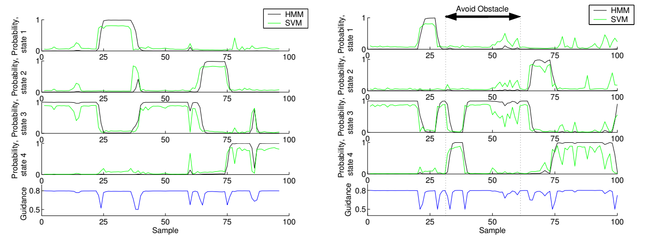


Fig. 2. **Left:** Estimated probabilities for the different states in experiment 1. **Right:** Estimated probabilities for the different states in the obstacle avoidance experiment 3. Estimates are shown for both the SVM and HMM estimator.

As it can be seen in Fig. 1(c), the size of the obstacle the user has to avoid has been changed so that the end-effector had to be raised twice as high as in the first experiment. The trajectory generated from the on-line execution shows that this does not introduce any problem for the control algorithm since appropriate guidance is provided at each state.

C. Experiment 3: Unexpected Obstacle

The final experiment was conducted in the same workspace as the first one. However, this time a new obstacle was placed right in the path of the learned trajectory, forcing the operator to leave the fixture. Once again, the same training examples as in the previous experiments were used.

Figure 1(d) illustrates the path taken in order to avoid the obstacle. The system always identifies the class which corresponds best with input data. However, as seen in Fig. 2, for unseen directions the guidance decreases, which enables the operator to regain control and safely avoid the obstacle.

Now, it can be seen that the overall task has changed and that new states were introduced in terms of sequencing. The proposed system not only provides the possibility to perform the task, but can also be used to design a new task model by demonstration if this particular task has to be performed several times.

References

[1] A. B. Kuang, S. Payandeh, B. Zheng, F. Henigman, and C. MacKenzie, "Assembling virtual fixtures for guidance in training environments," in *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 367–374, 2004.

[2] M. A. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. Moore, R. B. Gillespie, and P. Akella, "Cobot architecture," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 377–390, 2001.

[3] R. H. Taylor and D. Stoianovici, "Medical robotics in computer-integrated surgery," in *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 765 – 781, 2003.

[4] D. Kragić, P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Human-Machine Collaborative Systems for Microsurgical Applications," in *International Symposium on Robotics Research*, 2003.

[5] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proc. of the IEEE, vol 77, no. 2*, pp. 257–286, 1989.

[6] A. Elgammal, V. Shet, Y. Yacoob, and L. S. Davis, "Learning Dynamics for Exemplar-Based Gesture Recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 571–578, 2003.

[7] A. Castellani, D. Botturi, M. Bicego, and P. Fiorini, "Hybrid HMM/SVM Model for the Analysis and Segmentation of Teleoperation Tasks," in *IEEE International Conference on Robotics and Automation*, 2004.

[8] H. Bourlard and N. Morgans, "A Continuous Speech Recognition System Embedding MLP into HMM," in *Advances in Neural Information Processing Systems*, vol. 2, 1990.

[9] J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

A software infrastructure for sensors, actuators, and communication

Thomas Johansson, Thomas Hellström
Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden
{thomasj, thomash} @cs.umu.se

1. Background and motivation

Various tools for development of software for mobile robots have been proposed over the last two decades. These tools exist at many levels of abstraction, and are designed to support the development in different ways. At the highest level, both the hierarchical, reactive, and hybrid robotic paradigms are represented by a number of architectures that implement the general ideas within the respective paradigm. These architectures describe, on a high level, different ways to organize the main components to achieve the overall goal of designing and constructing an intelligent mobile robot. At the intermediate and low levels, there exists a number of general systems for development of robotic software

However, it is our firm belief that all levels of tools mentioned above overlook important issues regarding target machines, actuators, and sensors. Some of the intermediate level systems do include handling of sensors and actuators, but do not pay enough attention to them in a more systematic and general way. The work presented here attempts to fill this gap, and provides a link between physical sensors/actuators (or rather their software counterparts: the driver routines) and the overall control program. Furthermore, the proposed system deals with multiple target machines and communication between software modules placed on the same, or different, computers. These issues become more and more important as the complexity of the developed robots increases, and support for them should, in our opinion, exist in the development process.

The system has been developed as part of an ongoing project for an autonomous path-tracking forest machine. The forest machine is intended to move autonomously along a recorded path through a changing environment. The machine has a number of sensors (e.g. position, obstacle, speed) and actuators (steering angle, throttle) to its aid. The numbers and types of sensors and actuators have varied during the development process, partly as a result of investigation of different sensor types, and partly as a result of the use of different vehicles. The project uses a small Pioneer robot for fast development, a software simulator for system testing and a Valmet 930 Forwarder for outdoor real-life testing of the system. This dynamic environment puts special demands on the software used during development, mainly demands for ease of configuration, quick exchange of vehicles, and replaceable sensors.

2. Primary criteria

The following set of criteria was deemed important at the outset of the design process:

Interchangeability of robot vehicles: We anticipated the need for different software modules that should represent different robot hardware, in particular software for our small Pioneer Robot and the 8-meter-long forest machine (Valmet 830). It should be easy to switch between the two machines without any code changes. This is a practical and efficient approach, especially when developing systems for large autonomous vehicles. Furthermore, support for this level of interchangeability will become more and more important as generic robotics systems for many types of tasks and platforms are being developed.

Interchangeability of sensors and actuators: In a complex system, sensors, and sometimes also actuators, often have to be replaced by similar, yet not identical, components. This kind of replacement is often a major part of the development and research process where different kinds of algorithms, sensors, and setups have to be evaluated and compared. Furthermore, this functionality is often needed in a completed robot. A satellite navigator e.g. may have to be replaced by odometry if the satellite signals are occluded, or a laser scanner used for obstacle detection may have to be replaced by a radar sensor due to weather changes.

Distributed processing: The initial specification placed one computer on the moving vehicle and another at a supervisor's point, and freedom to place parts of the system on either machine was required. Placing the whole system on only one machine simplified debugging in the office. Freedom in the placement of software parts also means that extra computing capacity can be added to the system without changes to the code. The different parts communicate via a standard TCP/IP network, either Ethernet or a wireless LAN.

Event driven: The system should be driven by events, from either data obtained from a sensor, or control messages from a user or control software. There are no polling loops in the support system, even though there are some in the high-level control software.

3. Design

From these primary requirements a set of more detailed requirements was specified to facilitate the design and construction of the system:

Modularity: A software 'module' is the basic building block in the system. Modules exist in a type hierarchy, with subtypes being sensors, vehicles, and actuators for example. At the top level in the hierarchy all modules have a common interface, i.e. they have a common set of operations that can be performed by them. Examples are close, open, get status and so on. This property of modules is used by the system to load, start, stop, and interrogate modules on a high level without knowing the detailed function of a particular module. A common loader for all types, which executes an initialisation file with module names and actual types, loads the modules.

Extendibility and flexibility: New modules and module types should be easy to add to the system, without any changes to the existing software. An important feature of the system is the low coupling between modules, i.e. they are effectively isolated from one another with regard to internal representation of data and functions. Only the exposed external interface is shown, and if a new module of a certain type is added, it can be handled as any other module of this type. An example would be a Speed Sensor type, whose only function is getSpeed. This leaves it up to the implementers to design the module in any way they want, as long as the module delivers data via its getSpeed function. In the system there exist several speed sensors, which obtain data from the machine itself or from the GPS receiver, but the rest of the system does not know, and does not need to know, the actual sensor used at any specific moment. If a new type of speed sensor is installed, its corresponding speed sensor module can be added to the system without any changes to the existing code. Which sensor to use can be configured at start-up or dynamically during the run.

Cohesion: The modules are designed for high cohesion, i.e. a module does just one thing, but does it well. For example, a sensor that would require some form of filtering of its data. Instead of incorporating filter code into the sensor module (and thus into all sensor modules that require it), a special filter module is developed. The filter module would have the same type as the sensor, and in effect would be a *virtual sensor*, used in the normal sensor's place. The filter module then uses the actual sensor as a source for its data. This virtual sensor will for all intents and purposes look exactly like a 'real' sensor to the users of the sensor data.

Multithreading: Every module should have its own execution thread, and thus run independently of other modules. Polling loops are discouraged, instead an event-driven system is used with multiple independently executing threads. The treads normally sleep and only wake up if a message arrives, either from another module or from the network or a user.

Object Orientation: The system should be written in an Object-Oriented language. Java's ease of development, excellent development environments, support for modular software and multithreading, combined with our experienced staff members, settled the question of implementation language.

4. Current status and future work

At present the system is used daily for the development of the path tracking forest machine. It comprises some 100 modules divided into these groups:

Sensors – speed sensors, angle sensors, several GPS position sensors, heading, attitude, range, and obstacle sensors.
Actuators – throttle and steering angle actuators for different vehicles
Controls – control panels that make it possible to run the vehicle either autonomously or via teleoperation
Indicators – most sensor types have a corresponding indicator to show the current measured value on a screen, mainly for debugging purposes.
Proxies and servers – facilitates the communication over the network.
Simulators – simulate most sensor types, to ease debugging of communication systems

In a system of this size it is important to be able to get a view of the health of the modules. All modules can report a basic status plus, where appropriate, the delay time for the data from a sensor. A Health Monitor has the knowledge of all loaded modules on a computer, and also communicates health information with all other computers, so the overall health can be assessed on any machine.

The system relies heavily on configuration data stored in initialisation files. Every module has a file that specifies individual parameters for the module, for instance logging options, network port numbers or serial communication parameters, and most importantly, any other module used. There is also one large configuration file that specifies what modules to load when the system starts. To ease the problems of keeping track of hundreds of initialisation files there is a graphical tool, a Configuration Manager, that makes it possible to specify and connect modules to each other. At present this is a static tool used before the system is run, but plans exist of a dynamic version that can be used to change configuration in mid-run.

Also a more powerful inspection system is planned, that enables the user to see into every module while it is running, and also, with great care, change values of parameters.

So far, the developed system has proven to be a powerful tool that can serve as a generic framework when developing mobile robots and autonomous vehicles.

Acknowledgements

This work was financed by the Kempe Foundations, VINNOVA, Land Systems Hägglunds, Carl Tryggers stiftelse, LKAB and Komatsu Forest AB. We gratefully acknowledge their support.

A Knowledge Processing Middleware Framework and its Relation to the JDL Data Fusion Model

Fredrik Heintz and Patrick Doherty {frehe, patdo}@ida.liu.se
Dept. of Computer and Information Science
Linköpings universitet, 581 83 Linköping, Sweden

In the past several years, attempts have been made to broaden the traditional definition of data fusion as state estimation via aggregation of multiple sensor streams. One of the more successful proposals for providing a framework and model for this broadened notion of data fusion is the U.S. Joint Directors of Laboratories (JDL) data fusion model [1, 3] shown in Figure 1. The gap between such models, which describe a set of functions which should be components of a deployed system, and the actual instantiation of data fusion in a software architecture is very much an open and unsolved problem.

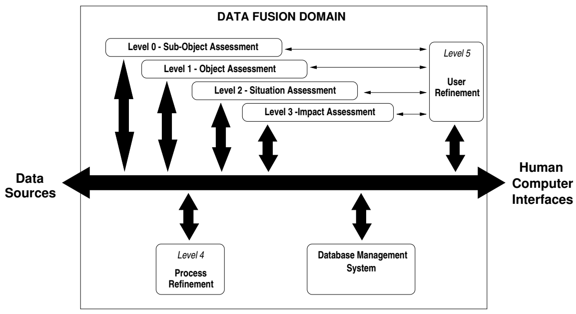


Figure 1: Revised JDL data fusion model from [1].

We have developed a knowledge processing middleware framework called DyKnow in an attempt to bridge the gap. DyKnow was designed, implemented, and tested in a prototype deliberative/reactive software architecture for a deployed unmanned aerial vehicle [2].

Conceptually, DyKnow processes data streams generated from different sources in a distributed architecture. These streams are viewed as representations of time-series data and may start as continuous streams from sensors or sequences of queries to databases. Eventually they will contribute to qualitative data structures grounded in the world which can be interpreted as knowledge by the system. Knowledge producing processes combine such streams, by abstracting, filtering and approximating as we move to higher levels of abstraction. In this sense, the system supports conventional data fusion processes, but also less conventional qualitative processing techniques common in the area of artificial intelligence. The resulting streams are used by reactive and deliberative services for control, situation awareness, monitoring, and planning to achieve mission goals. A knowledge producing process has different quality of service properties, such as maximum delay, trade-off between data quality and delay, how to calculate missing values and so on, which together define the semantics of the chunk of knowledge created. The same streams of data may be processed differently by different parts of the system relative to the needs and constraints associated with the tasks at hand.

Ontologically, we view the external and internal environment of the agent as consisting of physical and nonphysical *entities*, *properties* associated with these entities, and *relations* between these entities. The properties and relations associated with entities will be called *features*. Due to the potentially dynamic nature of a feature, that is, its ability to change values through time, a *fluent* is associated with each feature. A fluent is a function of time whose range is the feature's type.

Grounding and anchoring internal representations of external entities in the world is one of the great open problems in robotics. Consequently, middleware systems for knowledge processing must provide suitable support for the management of representations and their relation to the external entities they represent. In DyKnow an *object identifier* refers to a specific entity and a *link* between two object identifiers represents that they are hypothesized as refering to the same entity. The collection of object identifiers linked together represents the current knowledge about the identity of the entity. The object structures are one of the main concepts used on the object assessment level of the JDL model where they are used to reason about objects and object types.

The fluents associated with features are the target of the representation in DyKnow. A feature has exactly one fluent in the world which is its true value over time. The actual fluent will almost never be known due to uncertain and incomplete information, instead we create approximations. Therefore, the primitive unit of knowledge is the *fluent approximation*. There are two types of fluent approximations, primitive and computed. A primitive fluent approximation acquires its values from an external source, such as a sensor or human input, while a computed fluent approximation is a function of other fluent approximations. To do the actual computation a *computational unit* is used. The computational unit is a function taking a number of fluent approximations as input and generating a new fluent approximation as output. The fluent approximation is a very general concept and is used on all the JDL levels to represent dynamic information while computational units are used to encapsulate existing data fusion algorithms.

Since a fluent may be approximated in many different ways each feature may have many approximated fluents associated with it. Each fluent approximation is specified by a declarative *policy* which represents how it is created and what properties it has. The policy is the context in which the observations of the feature are interpreted. The policy can also be used by other services to reason about the fluent approximations. This is very useful at the process and user refinement levels where the system or the user should be able to adapt the current data fusion process as the world evolves.

Two important concepts in many applications are states and events. In DyKnow a *state* is a composite feature which is a coherent representation of a collection of features. A state synchronizes a set of fluent approximations, one for each component feature, into a single fluent approximation for the state. The need for states is obvious if we consider that we might have several sensors each providing a part of the knowledge about an object, but whose fluent approximations have different sample rates or varying delays. The state concept is the other important concept on the object assessment level since it is used to fuse data streams from different sensors related to a single object into a coherent representation of that object.

An *event* is intended to represent some form of change or state transition. Events can either be primitive or generated. Generated events can either be extracted from fluent approximations or computed from other events. DyKnow currently has support for two types of computed events. The first is the evaluation of linear temporal logic (LTL) formulas becoming true or false. The second is the recognition of scenarios, called chronicles, composed of temporally related events, expressed by a simple temporal constraint network. An LTL formula is evaluated on a state stream containing all the features used by the LTL formula, so the state extraction mechanism mentioned above is a prerequisite for the LTL formula evaluation. The chronicle recognition engine takes events representing changes in fluent approximations as input and produces events representing the detection of scenarios as output. These can be used recursively in higher level structures representing complex external activity such as vehicle behavior.

The event concept is important on the situation assessment level of the JDL model where relations between objects should be recognized. Situations can be represented by computed events such as temporal logic formulas or chronicles describing temporal relations between events. This way different features are fused together over time in order to extract more abstract situations. The computed events are also well suited to extract higher level events that can be reported to the user as an indication that something important has happend. This abstraction can be used on the user refinement level to reduce the cognitive load on the user.

To summarize, we believe that DyKnow provides suitable concepts to integrate existing software and algorithms related to data fusion and world modelling in general. Observe that the focus is not on individual data fusion techniques but on the infrastructure which permits the use of many different data fusion techniques in a unified framework.

References

[1] E. Blasch and S. Plano. Level 5: User refinement to aid the fusion process. In B. Dasarathy, editor, *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, 2003.

[2] P. Doherty, P. Haslum, F. Heintz, T. Merz, P. Nyblom, 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*, 2004.

[3] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White. Revisions and extensions to the JDL data fusion model II. In P. Svensson and J. Schubert, editors, *Proc. of the 7th Int. Conf. on Information Fusion*, 2004.

THE WORLD'S SIMPLEST MECHANISM SIMULATOR
or
Making Engineering Students Suddenly Discover that They Want to Devour Mathematics and Physics

Martin Nilsson

SICS/Mälardalen University/Lund Institute of Technology, POB 1263, SE-164 29 Kista, Sweden
E-mail: from.swar05@drnil.com

EXTENDED ABSTRACT

Mechanism simulation is an essential tool for robotics and engineering. Computer programs for simulating rigid-body dynamics are usually large and elaborate pieces of software. Some of the most well-known high-end commercial simulation packages are MSC.Nastran [1] (formerly Working Model), ADAMS [2] (originally developed by Mechanical Dynamics, Inc.; purchased by MSC in 2002), LMS Virtual.Lab [3] (formerly developed as DADS by Dassault), SIMPACK [4] (originally developed at DLR), Dymola [5], and Pro/ENGINEER Mechanism Dynamics [6], typically priced in the 10,000-USD/EUR range. There are also several decent free or GPL'ed simulators, for instance ODE [7] and DynaMechs [8]. There are many other systems, but the complexity of all these systems has led to an apparently common belief that rigid-body dynamics simulators must necessarily be large and complex, and that developing such simulators will require an expert team of programmers with Ph.D.'s in Lagrangian dynamics and differential geometry.

This is wrong.

We claim that the mathematics and physics knowledge of even a **first-year** engineering student suffices in order to implement a useful rigid-body dynamics simulator, powerful enough to handle kinematical loops. Of course, the student's simulator is unable to compete with the professional systems when it comes to features and efficiency, but it will be perfectly useful for many non-trivial applications, and *most importantly, the student will be able to completely master its theory and implementation.*

Many students in our environment have explicitly expressed a wish to implement a "physics engine" or "game engine", subsuming a dynamics simulator. On top of this, the feeling of being in total control of one's own implementation is clearly a strong motivational force for a student [9]. This can be taken advantage of: The development of a simulator implies a whole syllabus of fundamental mathematical and physical concepts, in a natural sequence for the student. Nobody **forces** the student to study mathematics or physics; the student will study these subjects because the student **wants** to, and because they **become interesting** (since they will solve the student's problems). *An especially attractive property of a dynamics simulator is its*

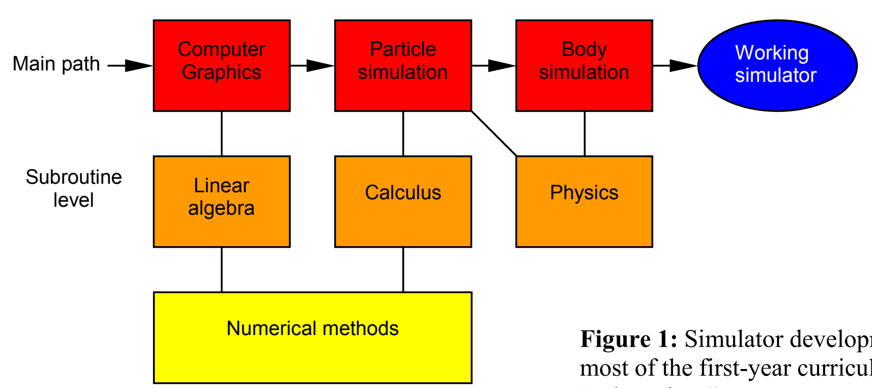


Figure 1: Simulator development path. It includes most of the first-year curriculum as a kind of "subroutines".

completeness in the educational sense. It will require practically all of the linear algebra and single-variable calculus of the first year, as well as mechanics up to and including motion of solid bodies (fig. 1). Furthermore, once the simulator is built, *it will provide an excellent tool for study and experimentation during the rest of the student's life at the university.*

The teacher's responsibility is to guide the students **strategically** into creating a good simulator structure. Obviously, students need to be directed towards a carefully thought-out structure in order to achieve the proper educational effect. We believe students should **not** be provided with ready-made components, but should essentially develop all code by themselves. However, it is equally important that their development is controlled, so that dead-ends are avoided, and so that the ground is appropriately prepared for a gradual transition to more advanced methods as their level of knowledge increases. Leaving the students too much on their own, as in some forms of Problem-Based learning (PBL), overwhelms them with difficult choices, and can become counterproductive in the long rum. We call our teaching concept **Controlled Problem-Based Learning** (CPBL) [9].

The big question is whether such a simulator is at all possible. The purpose of the rest of this presentation is to sketch a realization. Our description is based on a minimal simulator that we are currently developing for the **OpenMechatronics** platform [10], an open system for assisting education, research and other situations in mechatronics and robotics, when resources such as time, money or people are limited.

A first observation is that it is perfectly possible to base a simulator on nothing beyond the classical Newton-Euler equilibrium equations. Lagrangian dynamics is unnecessary. However, a key to simplifying the simulator is the formulation of the **constraints**, which should at least include the lower pair joint constraints. With a suitable formulation, it becomes possible to demonstrate how Lagrangian dynamics **develops** from Newton-Euler in a concrete and elegant way. The single fundamental constraint we use is **point-constrained-to-surface**. Each such constraint removes one degree of freedom, and several of them can easily be combined to express prismatic, rotational and universal joints.

Closed kinematical loops are essential in robotics, and do need to be handled by the simulator. The collision problem, on the other hand, can be complex, but although it need not be handled in its full generality for mechanism simulation, the simulator should be prepared for such development, suggesting e.g. momentum for a choice of state variables. Initial representations should be chosen intuitively clear rather than numerically optimal. For instance, rotations should first be represented as matrices, demonstrating the need for better numerical representation, paving the way for quaternions. Other examples of such a modular development path, or tree, are conversion from filled to sparse linear equation solvers, and for standard Runge-Kutta solvers to stiff ODE solvers.

An essential part of the development, not to be forgotten, is the **visualization** using computer graphics. Graphical presentation constitutes the student's first feedback and becomes a concrete illustration of many abstract concepts, not only in of linear algebra, but throughout the mathematical curriculum.

The importance of **modularity** needs to be emphasized. In order to maximize speed or generality, a simulator may need to "merge" or "interleave" different stages, e.g., integration and collision detection. However, this will make the structure of the simulator harder to understand and develop. We believe that it is important, at least initially, to design the simulator in such a way that modules can be individually upgraded, although this could mean a certain performance loss.

CONCLUSIONS

We propose that development of a mechanism dynamics simulator **a)** is highly motivating for an engineering student, **b)** awakens the student's interest in mathematics and physics, and **c)** provides a valuable tool for further study and experimentation. We also claim that such a simulator can be written with a minimum of knowledge, well within reach even for first-year students, and we have briefly sketched some essential features of such a simulator.

REFERENCES

[1] *MSC.Nastran*: http://www.mscsoftware.com . June 2005.
[2] *ADAMS*: http://www.mscsoftware.com . June 2005.
[3] *DADS/LMS Virtual.Lab*: http://www.lmsintl.com . June 2005.
[4] *SIMPACK*: http://www.simpack.de . June 2005.
[5] *Dymola*: http://www.dynasim.se . June 2005.
[6] *Pro/MECHANICA*: http://www.ptc.com . June 2005.
[7] Smith, R.: *ODE*: http://www.ode.org . June 2005.
[8] MacMillan, S.: *DynaMechs*: http://dynamechs.sourceforge.net . June 2005.
[9] Nilsson, M.: *Evaluation of the Computer Mechatronics course at LTH for the years 2003-2005*. Seminar at Lund Institute of Technology (LTH), September 2005.
[10] Nilsson, M.: *OpenMechatronics*, http://www.openmechatronics.org/ . August 2005.

Binary Agents for the Control of Autonomous Vehicles

Andrew Wallace
Frontios
AAWallace@iee.org

***Abstract*— Behavioural based (see [1] and [2]) systems are commonly used in robotic research. Arguably, the main area of activity has been in the area of integrating behavioural or reactive systems with deliberative systems in the form of hierarchical systems and in the area of action selection.**

Examples of combining architectures includes [3] who use layered architecture for an office delivery robot using reactive and a decision-theoretic methods for navigation and obstacle avoidance. Examples of the action selection problem include [4], [5] and [6] where evolutionary approaches are taken to evolve the action selection of a behavioural based system. In [7] and [8] learning momentum and Q-Learning are methods used for action selection. However, each method uses traditional types of behaviours, which have to be hand crafted. This paper, however, presents an alternative construction method for autonomous, behavioural based, vehicles such as mobile robots. The method centres around the use of binary agents and is intended to lend itself to the solving of the action selection problem by providing elementary building blocks, which through networking can form behaviours and a behavioural based controller. They are intended to form the foundations of a method to automatically construct behaviours for complex systems such as mobile robots, humanoid robots and information fusion systems.

This paper presents a novel architecture that is intended to lend itself to the action selection problem. The architecture centres around the use of binary agents where a binary agent can be considered to be the simplest form of agent being one that takes a single input of a binary value.

Taking an object oriented view, a binary agent can be viewed as a class, which inherits from a more general class of Agent, and consists of a set of states augmented with timers defining its behaviour so that when an agent has been turned on it advances from state to state after a time-out. States are, therefore, arranged in a circular pattern looping from end to start. Thus the transition from state to state is an ordered sequence, $S =< 1, 2, ...n >$ as shown in figure 1.

The binary agents are then connected in a network of agents to form a controller. Any one agent can be connected to one or more other agents. It follows, therefore, that for every state that the agent is in it will be able to send a binary signal to 0 or more agents, thus turning them on or off. The pattern of signals sent is referred to as the activation pattern and can be different for each state.

For example, if AgentA, AgentB and AgentC were binary agents so that AgentA was connected to both AgentB and AgentC. AgentA could turn AgentB on and AgentC off in one state. Its activation pattern for that state would then be 1, 0. In its next state it could turn AgentB off and AgentC on. The activation pattern for that state would then be 0, 1.

This paper presents research into the application of such binary agents to the control of an autonomous vehicle. A description of the agents is presented in section II.

An example system that was implemented on a physical agent (that is, a mobile robot platform) consisting of a light sensor, two touch sensors and two motor actuators is presented in section III. The robot was then given the tasks of exploring an unknown environment and searching for a light source.

To achieve its tasks the robot controller required agents to drive the robot forward, backwards and to turn so the robot was able to manoeuvre. It then required agents to be able to detect the light level for searching for a light and agents to detect the activation of the touch sensors.

For the motion of the robot four agents were used, one forward and one backwards agent for each motor. There are two tasks for the light sensor. One is to detect the light source and one is to detect the absence of the light source. There are two tasks associated with the touch sensors. One is to detect the activation of the sensor and the other is to perform the required directional changes of the motors. Therefore, this leads to the following agents (see figure 2):

Four motor agents that control the motion of the robot.

Two light sensor agents to detect light thresholds

Two touch sensor agents, one for each touch sensor

Two additional binary agents to control the light sensors and motor agents when the sensor agents had been turned on.

Each motor agent had the task of turning the motor on in a given direction. Therefore, there were two agents for each motor; one to drive the motor forward and one for reverse as follows:

MotorAFwdAgt

MotorCFwdAgt

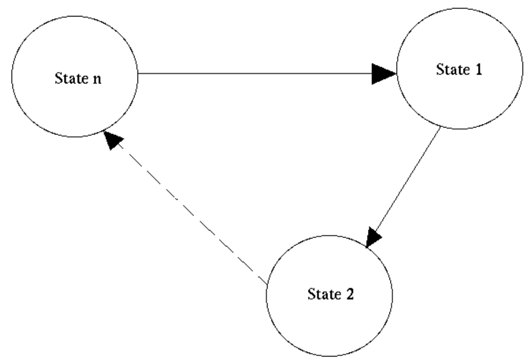


Fig. 1. State diagram for a binary agent

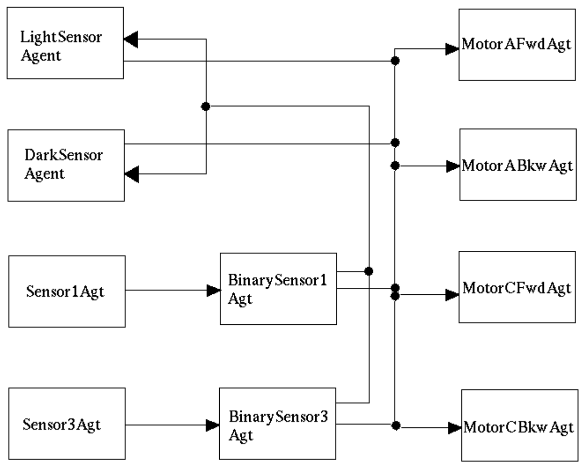


Fig. 2. Binary agents used in the robot controller

MotorABkwAgt

MotorCBkwAgt

To direct the robot to a light source there were two light agents. Each light agent was of the threshold type of binary agent. The DarkSensorAgent was turned on when the light level fell below a threshold and the LightSensorAgent was turned on when the light level went above a threshold level. The light sensor agents were connected to the light sensor and to the four motor agents. When each respective light agent was turned on they would turn on or off the motor agents. The DarkSensorAgent turned off one reverse motor agent and one on and one forward motor agent off and one on resulting in the robot spinning on the spot in a "search out the light" behaviour. When the light sensor agent was turned on it turned off both reverse motor agents and turned on both forward agents. This resulted in a "head towards the light" behaviour. Each agent had one state.

The touch sensor agents (Sensor1Agt and Sensor3Agt) where directly connected to the touch sensors and to two another agents. The two other agents were referred to as BinarySensor1Agt and BinarySensor2Agt. The sensor agents had the task of turning the binary sensor agents on when the touch sensor was closed. The BinarySensor*Agts had the task of turning off the light sensor agents and the motor agents. As the binary agents cycled through their states they first turn both reverse motor agents on and both forward agents off causing the robot to reverse. They then turned one forward motor agent on causing the robot to rotate. Finally they turned both forward agents on causing the robot to move forward. The light sensors were then also turned on. This resulted in a "move away from object" behaviour which occurred after a collision with an object.

The Sensor*Agts each had one state. The BinarySensor*Agts had four states.

To evaluate the controller the robot was given the task of exploring an unknown environment. The environment was a flat surface with a random set of object placed at various locations and was surrounded by a wall on all sides. The robot then explored the environment colliding with objects. As it did so the expected set of behaviours were observed. The robot reversed, turned and then advanced after each collision.

After the robot ran for one minute the light sensors were activated. When a light source was not present or detected the DarkSensorAgent was active causing the robot to enter into a light seeking behaviour. When a light source was detected the LightSensorAgent was active causing the robot to enter into the "head towards a light" behaviour.

If a collision was detected the light sensor agents were turned off and the robot went into the "move away from object" behaviour.

The robot performed as expected with the switching of external observable behaviour achieved by switching agents on or off.

A description of the experimental run with the robot is given in section IV. The binary agent controller was able to control a mobile robot platform and externally observable behaviours were evident despite there being no explicit behavioural representation being programmed. The resulting behaviours can be seen as emerging from the interaction of the binary agents or they can be viewed as encoded in the states of the binary agents and the activation patterns and the network of interconnected agents.

The application and the construction of the agents and their states were engineered by hand. As the agents are formed in a network where activation signals are sent from one agent to one or more other agents it can be considered to be similar to a network of neurones in a Artificial Neural Network. The activation pattern in such a network can be learnt as well as the interconnections and the network itself. Therefore, there is a possibility that the state machines, the on / off activation patterns for each state and the network of binary agents could be learnt. Methods such as reinforce learning or genetic algorithms could be employed to learn the structure of a binary agent controller.

Another possibility for the automated formation of binary agent controller and / or the state machines and activation patterns is the use of cellular automata. Cells in a cellular automata network are either on or off and as such have a binary state similar to binary agents. The rules for determining the on or off states for each cell could be used to decide the connections between the binary agents or the activation patterns and state machines of each agent.

Other aspects of interest to study is the scalability of the system which is associated with handling more complex problems. Also it would be interesting to study how binary agents would handle noisy or imperfect data.

A discussion of the binary agents is presented in section V and a summary in section VI, conclusion in section VII and future work in section VIII.

REFERENCES

[1] R. A. Brooks, "A robust layered control system for a mobile robot," Tech. Rep. A.I. Memo 864, Massachusetts Institute of Technology Artificial Intelligence Laboratory, September 1985.

[2] R. C. Arkin, *Behavior-Based Robotics*. MIT Press, 1998.

[3] R. Simmons, R. Goodwin, K. Z. Haigh, S. Koenig, and J. O'Sullivan, "A layered architecture for office delivery robots," in *First International Conference on Autonomous Agents*, pp. 245–252, February 1997.

[4] M. Wahde and H. Sandholt, "Evolving complex behaviours on autonomous robots," in *Proceedings of the* 7^{th} *UK Mechatronics Forum International Conference*, 2000.

[5] M. Wahde, "A method for behavioural organization for autonomous robots based on evolutionary optimization of utility functions," *Journal of Systems and Control Engineering*, vol. 217, pp. 249–258, 2003.

[6] C. Jassadapakorn and P. Chongstitvatana, "Reactive planning with evolutionary computation," in *National Computer Science and Engineering Conference*, pp. 357–361, 2002.

[7] E. Martinson, A. Stoytchev, and R. Arkin, "Robot behavioral selection using q-learning," in *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), EPFL, Switzerland*, 2002.

[8] J. B. Lee and R. C. Arkin, "Adaptive multi-robot behavior via learning momentum," in *In Proceedings of IROS'03*, 2003.

Session Manipulation

Telerobotics for use in Contaminated Workspaces

Johan Forsberg[1] Åke Wernersson[2, 3]

[1] Mobile Robotics Sweden AB, jf@mobile-robotics.com
[2] Computer Science & Electrical Engineering, Luleå Univ of Technology, ake.wernersson@ltu.se
[3] Laser Systems, FOI 32, Linköping (ake.wernersson@foi.se)

Abstract: This paper is on telecommands for semi autonomous operation, i.e. give the robot local intelligencs. The Tracking Control System outlined below is developed for the demands of the nuclear decommissioning industry. The system is designed for high safety, high flexibility and high reliability. The fundamental technology is triangulation of markers on the robot detected by cameras in fixed positions.
Mobile Robotics Sweden AB specialises in robot control of heavy robust machinery using non-contact sensing. Below some *existing* modules are outlined. Other applications take up in the paper are for telemedicine and construction. More info on http://www.mobile-robotics.com

Workspace Protection System, WPS
The system is based on continuous tracking of the motion of the robot arm using cameras and markers on the arm. The same system also observes the walls and other obstacles. Obstacles can be programmed into the system or entered manually by the operator. The arm is automatically stopped if there is a risk of it colliding with an obstacle. For maximum safety the system continuously supervises itself and checks that the measurements are consistent and of sufficient quality.

The collision avoidance adapts to the velocity of the arm. When the arm is moving fast it isn't allowed to be as close to the work space boundaries as it would be when moving slowly. Therefore there is always enough margin for the arm to stop. This also allows the operator to slowly return the arm to the work space after an emergency stop with the collision avoidance still active.

The MR-WPS System gives the end user :
- Continuous self-testing of the safety system. If MR-WPS loses contact with the machine for more than 0.2 seconds the machine is stopped.
- MR-WPS takes the arm's speed into account.
- High flexibility and ease of use. Possibility to define irregular safety zones.
- Easy upgrading of existing machines.

Tool Point Control System, XYZ
MR-XYZ gives the user a more efficient, safer and easier way of steering the Brokk robot arm. It uses the Mobile Robotics Tracking System to let the operator directly control the movement of the robot tool in XYZ coordinates rather than the individual cylinders. This is especially useful when the robot is remotely controlled.
- Joystick motions are directly related to the motion of the tool, not the individual cylinders.
- The signals from the control box are computer processed before reaching the Brokk.
- The Brokk arm is equipped with the same reflectors as in the MR-WPS System.
- Switching between MR-XYZ and the conventional control box is done by pressing a button.

Following Surfaces
If you point the tool at three points on a flat surface and then press the surface following button the X and Y joysticks can be used to follow the surface.
The up/down joystick can be used to change the distance to the surface.

Teach and Repeat System
As with MR-XYZ the signals from the control joystick are computer processed before reaching the Brokk arm. Using the MR-XYZ control box you can teach the robot movements that it can repeat, for example lifting the arm over an obstacle.

Automatic Movements
In certain tasks within the nuclear decommissioning industry it is required for the hydraulic arm to be moved very precisely and slowly, movements which are virtually impossible to perform manually. MR-AM is developed for user friendly programming of automated movements. With the assistance of the Tracking Control System the hydraulic arm can move along precise paths in relation to the surroundings. One example of slow and precise movements is cutting steel with a blowtorch.

Factory test

References

Nygårds, Wernersson, Larsson and J. Forsberg, *"On Control of Large Machines: feedback from optronics/lasers in the tool"* 7th Scandinavian International Conference on Fluid Power, SICFP'01, Linköping, May, 2001

B Nilsson, J Nygårds, U Larsson, Å Wernersson, *Control of flexible mobile manipulators: position and vibration reduction using an eye-in-hand range camera*, IFAC, Control Engineering Practice 7(1999)pp741-751.

http://www.sm.luth.se/csee/ra/ old page with more info

Constrained Path Planning for Mobile Manipulators

*Frank Lingelbach, †Daniel Aarno and †Danica Kragić
Centre for Autonomous Systems
*S3 - Automatic Control
†Computational Vision and Active Perception Lab
KTH, Stockholm, Sweden

***Abstract*— This paper presents our ongoing research in the design of a versatile service robot capable of operating in a home or office environment. Ideas presented here cover architectural issues and possible applications for such a robot system with focus on tasks requiring constrained end-effector motions. A key component of such system – a path planner – is presented in detail along with an overview of the software architecture it fits into.**

I. Introduction

Tasks requiring motions with pre-defined end-effector paths arise frequently for autonomous robots in a home or office environment. Door opening is an evident example, where the end-effector, grasping the door handle, has to move on a circular path around the hinge. For an autonomous service robot, it is therefore of fundamental importance to be able to plan such motions on its own. Beyond this, there exist many more applications in the field of industrial robotics, like arc welding or spray painting, or in robot assisted surgery. Related problems, where the path of the end-effector is not completely specified but constrained to some degree, might appear as carrying a glass of water or keeping an object in the field of view of an eye-in-hand camera for visual servoing.

This paper does not consider any differential constraints. These emerge from the dynamics of the robot and are typically acceleration or velocity constraints. A car-like robot can, for example, move only along its steering direction.

Each subtask that implies a motion of the robot raises the question of how to get from the current configuration to the desired one. On its way to the new configuration, the robot must not collide with any obstacle. Furthermore, additional constraints might have to be taken into account. For example, when opening a door the robot must follow the natural trajectory of the door handle. The problem of getting from A to B on a feasible path is addressed in the field of path planning.

Path planning alone is often not enough to execute a task. This could be because the model of the environment contains errors, the environment is not completely known or contains moving obstacles. To overcome these issues, task-compliant on-line modification of the planned path is necessary.

II. Related Work

In our previous work, we have shown that it is possible to perform door opening with a hybrid dynamic system architecture. The door opener presented in [1] uses an on-line estimate of the door radius to move the base on an appropriate arc. The manipulator is only used to perform force relaxation. Vision is used to detect and grasp the door handle. This approach clearly fails if the appropriate arc is not collision free. A door in a corner of a room or a door opening outwards, where the robot has to pass through the door frame, cannot be handled by this approach.

In recent years, the field of path planning has been dominated by probabilistic methods like Rapidly-exploring Random Trees (RRT) [2] and Probabilistic Roadmap Methods (PRM) [3]. Recently, we presented a new path planning method called Probabilistic Cell Decomposition (PCD) [4]. An extensive overview over the field of path planning can be found in [5]. Due to its good performance for articulated robot platforms, PCD is used for path planning in this paper.

Path planning for redundant robots with end-effector constraints has been presented in [6], where Oriolo *et al.* sample the self-motion manifolds for configurations along a given end-effector path to ensure the correct end-effector pose. The progress along the end-effector path is monotonous. There might be problems that are solvable only if stepping backward on the end-effector path is permitted. Therefore, we propose to take up the progress along the end-effector path as an additional dimension for the path planning problem.

III. The Architecture

A typical task for an autonomous service robot could be: ”Get the milk from the refrigerator!”. The high level “get milk” task can be decomposed into a number of subtasks on a lower level like: go to the kitchen – open the refrigerator – get the milk – close the refrigerator – return to the user. Each of these tasks can then be divided into even finer grained tasks, for example, the open the refrigerator task can be divided into: locate the refrigerator – locate the door handle – approach the door handle – grasp the door handle – move the door handle along a specific trajectory – ungrasp the door handle.

Executing such tasks requires, among other things, knowledge of different areas such as software and hardware architectures, low-level and real-time programming, modeling, automated control, fault-control, human-robot interaction, localization, mapping and path planning.

At the Centre for Autonomous Systems an architecture for integrating research from different areas into a complete system is under development. The NoMan (**No**vel **Man**ipulation) architecture provides a simple to use architecture for testing ideas on real robots. One of the short term goals of NoMan is to combine relevant research from different areas into a system capable of navigating an office environment and performing the “get milk” task. The NoMan project uses a deliberate/reactive architecture.

The top layer of the NoMan architecture is the specification layer where tasks are specified. A task specification can be as simple as a predefined task (e.g. a surveillance robot) or it can be more complex, originating from a human-robot interaction (HRI) sequence (e.g. a verbal speech command or a gesture) [7], [8]. From the specification layer a *task description* is propagated to the task planner. Such a task description could be `Fetch(MILK)`. Using the *knowledge base* the task planner then decomposes the task into sub-tasks. For instance, the knowledge base knows that the milk is likely to be in the refrigerator and the refrigerator is in the kitchen. The task planner can then use the world model and the path planner to compose a necessary low-level task description for performing the `Goto(KITCHEN)` subtask. This low-level task description is then fed to the task execution layer. From the low-level task description the task execution layer starts a set of reactive behaviors and monitors their progress. In the `Goto(KITCHEN)` example the reactive behavior would be the *path adaptation* behavior which is composed of an obstacle avoiding and a path following behavior.

IV. Theoretical Background and Implementation

Path planning is carried out in the configuration space. For a manipulator mounted on a mobile base, the dimension of the configuration space is defined by the number of degrees of freedom (dof) of the platform. The platform used in our experiments is composed of a mobile XR4000 base (3 dof), a PUMA 560 arm (6 dof) and a Barrett hand (4 dof). For a given grasp, the joints of the hand are fixed. Thus, only 9 dof remain. The rotational axis of the base coincides with the axis of the first joint of the arm. Together with the fact that the shape of the base is rotational invariant, if the small gearbox on top of the base cylinder is neglected, we assume all configurations equivalent for which holds $x_{\text{rot}} - x_0 = \text{const}$ (where x_{rot} is the base rotation and x_0 is the state of the first joint)

For some tasks that the robot has to accomplish, there may be additional constraints on a feasible path. A very common one is, for example, a given end-effector path. If the pose of the end-effector is given, the inverse kinematics of the platform can be used to calculate a feasible joint configuration. For a redundant platform, there exist infinitely many solutions to the inverse kinematics. These can be grouped into a finite number of self-motion manifolds [9]. The joint configuration can change continuously along such a manifold without affecting the pose of the end-effector. We therefore propose to perform probabilistic path planning in the space $C_p = s \times p_s$,

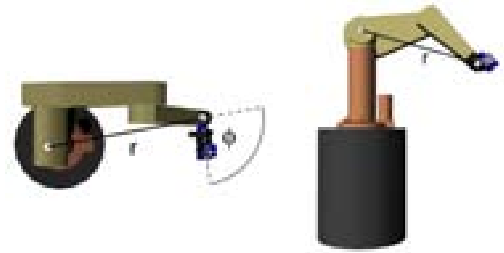
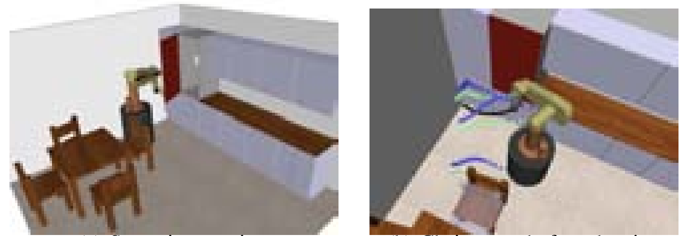


Fig. 1: Top and side view of the platform with parameterization used for constrained path planning



(a) Scenario overview (b) Chair moved after planning

Fig. 2: (a) Overview over the kitchen scenario; (b) Planned path (light green) and executed path using adaptive path following when the chair blocks the planned path (dark blue)

where s is the progress along the end-effector path and p_s is a parameterization of the self-motion manifold for a given s.

Using the example of opening a door, a given grasp, obtained from a grasp planner [10], fixes the position of the end-effector relative to the door handle. Thus, a given opening angle of the door determines the pose of the end-effector and consequentially restricts the feasible configurations of the platform. To open the door, the end-effector has to follow a path that is defined by the position of the door hinge, the distance between the door handle and the hinge and the chosen grasp. For the platform described above, there remain 2 dof after fixing 6 dof by the given end-effector pose. The problem can be parameterized in the following way: α is the door opening angle, i.e., the progress along the end-effector path and ϕ, r is the position of the base relative to the end-effector in polar coordinates. We denote this 3-dimensional configuration space by C_p. For each collision check, the configuration in C_p has to be mapped to the natural configuration space C_n. For a given triple (α, ϕ, r), the position of the end-effector and the base are fixed. The inverse kinematics of the manipulator then maps the relative base – end-effector position to a feasible configuration of the manipulator. Thus, in addition to the binary collision check, there arises another requirement for a configuration $x \in C_p$ to be feasible: the inverse kinematics must have a solution for x on the chosen manifold. For the PUMA 560 there exist eight self-motion manifolds characterized by elbow-up / elbow-down, right-arm / left-arm and flip / no-flip.

There may be problems that are not solvable without switching to another manifold. Thus, a path planning algorithm that maps a C_p configuration to only one specific manifold can not find a feasible path for this kind of problems. This can be seen as a limitation of our approach. On the other hand, to switch to another manifold, the manipulator has to pass a singular configuration like a fully extended arm. However, in such a configuration the manipulator can not react to external forces in all directions. Thus, unexpected deviations from the path may be harmful to the platform. Therefore, it seems natural to avoid these singular configurations and, consequentially, remain on one manifold. The choice of manifold is up to the task planner that provides the path planner with a feasible start and goal configuration. Here, a possible approach could be starting with a standard manifold or checking random samples on the different manifolds for collision to obtain a rating of the manifolds.

V. Experimental Evaluation

These preliminary experiments have been carried out with the newly developed Components for Path Planning (CoPP) presented in [11]. The distance computations has been performed with the Proximity Query Package (PQP) [12], [13].

Figure 2.a shows the setup for our simulation experiments. The path planner has to plan a constrained motion to open the refrigerator. The average planning time for this problem is well below one second. In a first experiment, the same world model is used for execution as for planning. Thus, the planned path is guaranteed to be collision free and the constraint on the end-effector is satisfied. In a second experiment, the world in which the task is executed deviates from the one used for path planning. The chair next to the robot is located 25 cm closer to the refrigerator. In Fig. 2.b the results for these two experiments are shown. The black balls correspond to the end-effector path which is the same for both experiments. The light green items show the planned path for some links of the robot, the dark blue items correspond to the path executed of the adaptive path follower when the chair is moved after planning.

VI. Conclusions and Future Work

We have shown how to integrate path planning and path adaptation for a door opening task and placed it in the context of a general architecture for mobile manipulation. One of the contributions of this paper is the idea of regarding the progress along the given end-effector path as an additional dimension of the path planning problem. This allows for solving problems where backstepping along the end-effector path is necessary.

We plan to extend our work to more general constraints. These may include holonomic constraints like pointing a camera on an object or carrying a glass of water. To be applicable on general mobile platforms it will be important to include non-holonomic constraints as well.

References

[1] L. Petersson, D. Austin, and D. Kragić, "High-level Control of a Mobile Manipulator for Door Opening," in *International Conference on Intelligent Robots and Systems*, 2000.

[2] S. LaValle and J. Kuffner, "Rapidly-exploring Random Trees: Progress and prospects," in *International Workshop on Algorithmic Foundations of Robotics*, 2000.

[3] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[4] F. Lingelbach, "Path Planning for Mobile Manipulation using Probabilistic Cell Decomposition," in *International Conference on Intelligent Robots and Systems*, 2002.

[5] S. M. LaValle, *Planning Algorithms*. [Online], 2004. `http://msl.cs.uiuc.edu/planning/`.

[6] G. Oriolo, M. Ottavi, and M. Vendittelli, "Probabilistic Motion Planning for Redundant Robots along given End-Effector Paths," in *International Conference on Intelligent Robots and Systems*, 2002.

[7] H. I. Christensen, D. Kragić, and F. Sandberg, "Vision for Interaction," in *Sensor Based Intelligent Robots* (G. D. Hager, H. I. Christensen, H. Bunke, and R. Klein, eds.), Lecture notes in Computer Science, pp. 51–73, Springer, 2001.

[8] E. A. Topp, D. Kragić, P. Jensfelt, and H. I. Christensen, "An Interactive Interface for a Service Robot," in *International Conference on Robotics and Automation*, 2004.

[9] J. W. Burdick, "On the Inverse Kinematics of Redundant Manipulators: Characterization of the Self-Motion Manifolds," in *International Conference on Robotics and Automation*, 1989.

[10] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic Grasp Planning using Shape Primitives," in *International Conference on Robotics and Automation*, 2003.

[11] M. Strandberg, *Robot Path Planning: An Object-Oriented Approach*. PhD thesis, KTH, 10 2004.

[12] "Proximity Query Package (PQP)." [Online]. `http://www.cs.unc.edu/~geom/SSV/`.

[13] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast Proximity Queries with Swept Sphere Volumes," Tech. Rep. TR99-018, Department of Computer Science, University of N. Carolina, Chapel Hill., 1999.

Simulating Tactile Sensors in Robotic Grasping

Johan Tegin and Jan Wikander
KTH Machine Design
johant@md.kth.se

***Abstract*— A tactile sense can be used to improve control of robotic grasping as well as in grasp evaluation and fault detection. By adding tactile sensors to an existing grasp simulator, there is now a tool specifically designed for grasping using sensor feedback that can aid in evaluating sensor based control algorithms and different sensor layouts.**

I. Introduction

Tactile sensing in robotic grasping is used to further improve grasping beyond what is possible using only information such as actuator output, joint torque and position. For certain tasks, we need tactile information and also the knowledge of how to use this information. Even though much information can be retrieved from other sensors and actuator drivers, it is only by touch that we can collect information from the very point of contact. An ideal simulation environment would enable us to test and evaluate different control algorithms as well as different sensor layouts.

II. Background

There are several robot simulators available today, but to our knowledge only one that features grasp planning and evaluation. GraspIt! developed by A. Miller [1] comprises several object and robot models, a grasp planner, grasp evaluation methods, a physics engine, and a graphical interface.

More advanced grasping has been impossible to perform in GraspIt! due to the lack of sensor models. Also, more advanced joint controllers are needed to make good use of this sensory information. These features have now been added and we are able to perform more advanced control of the grasp formation process, see Figure 1.

Key to secure grasping is knowing that the grasp is of sufficient quality so that the object does not slip or drop. GraspIt! can, using the contact positions and forces from simulation, analytically evaluate grasp quality. But an executed grasp, both in simulation and in reality, will differ somewhat from the planned grasp. Unfortunately, it is not easy to find the contact points and forces of the grasp executed in reality. Either sensing that supply the data for analytical grasp quality evaluation, or some kind of algorithm that compute a quality measure based on what we actually do know, is needed.

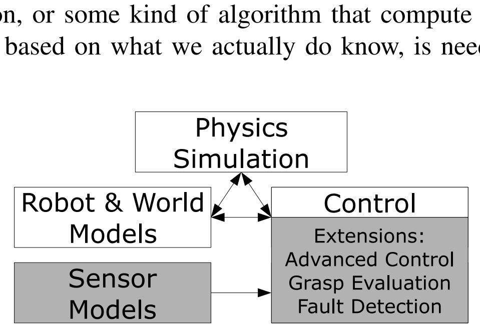


Fig. 1. Expanding GraspIt! with tactile sensor information opens up new opportunities for control.

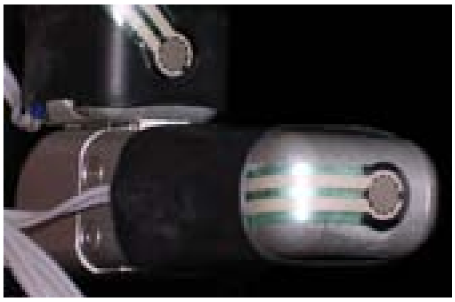


Fig. 2. Force sensing resistors mounted to a finger link and to the fingertip on the finger of a Barrett hand.

The hardware required to find contact points and forces is difficult to design, control and also costly. It is our belief that simple grasping/fixturing can be securely performed using less complex hardware.

Data more easily collected in real grasping is typically joint torques, joint positions, and simple tactile sensor information. This is enough to allow simple feedback control and to compute a crude grasp quality estimate. Since all contact information is available in the simulation, this estimate can be compared to the actual grasp quality computed by GraspIt! and also to the quality of the planned grasp.

A Barrett hand equipped with small force sensing resistors is used in the simulations reported here.

III. Grasp Sensing

The sensor layout depends on numerous factors: robot hand, object, task, environmental conditions etc. For more advanced manipulation, as in manipulating an object without moving the palm, the robot hand typically needs to be dextrous and fingertip force/torque sensors may be suitable or even needed to aquire necessary information. But, as mentioned, we will focus on using more simple hardware.

A. Sensor Hardware

Force sensing resistors (Figure 2) are among the most simple tactile sensors. They are affordable and easy to mount to the robot hand. Unfortunately, they only measure normal force and are rather small, see Figure 2. But since our intention is to perform grasping using rather simple hardware, we are also interested in the limitations posed by the sensor hardware. For this, it is of good use to see how the sensor information changes during grasp formation in order to find out whether better performance can be achieved by using a better controller, grasp replanning, or if the sensor layout must be improved.

B. Sensor Models

The tactile sensor was assumed to have an outer diameter of 10 mm and a thickness of 1mm. The sensors were added to the last link of each finger and can be seen in Figure 3.

From GraspIt! it is possible to extract contact information in the form of contact points and 3-D forces. Several contact points, and corresponding forces, may exist on one sensor

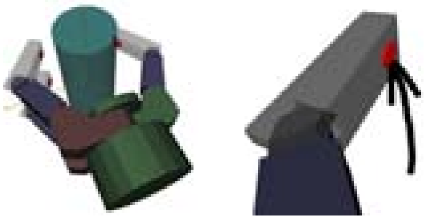


Fig. 3. The modelled sensors are placed at the fingerpads.

surface. By summation of the contact forces normal to this surface, one sensor reading per sensor is achieved. This data processing is the first step in simulating the force sensing resistor. Hysteris, drift, the effect of forces parallel to the sensor and other sensor imperfections can be added if desired.

IV. Scenario

A simple example was created to exemplify sensor use for grasp evaluation and to show some simulated sensor readings. The task is grasping a box in a fingertip grasp. We use the sensors to ensure that we end up with the intended grasp where all sensors are touching the box. The box size, position and orientation are known but with uncertainty of a few centimeters. We neglect gravity compensation and the joints are modeled ideal with no friction. The very simple closing algorithm is:

1) Use a constant torque to start closing the fingers
2) When the preset maximum velocity is reached, stop acceleration, i.e. set the joint torque to zero
3) When the velocity is below a threshold, that is, when there is contact, set a constant torque to the joint

By using low speed, the momentum of the fingers is kept low to keep the box from moving at impact.

When the fingers have closed, we can verify that the planned grasp was performed by checking that the sensor values correspond to the expected. If the fingers have closed and any sensor reading is zero, we cannot know for sure that we reached the planned grasp.

V. Results

As can be seen in the first trial, Figure 4, the sensors never come in contact with the object. This is also clear if we look at the data from the tactile sensors in Figure 5.

If we look at another trial, Figure 6, we see that the sensors make contact with the box. The force peaks in Figure 7 arise from inertial effects. After the initial peak/impulse, the finger has stopped and we can see the forces balancing the joint torques. Thus, it is likely that we have reached the desired grasp.

Acknowledgment

This work was supported by the Swedish Foundation for Strategic Research through the Centre for Autonomous Systems at the Royal Institute of Technology, Stockholm, Sweden.

References

[1] A. T. Miller. *GraspIt!: A Versatile Simulator for Robotic Grasping*. PhD thesis, 2001.

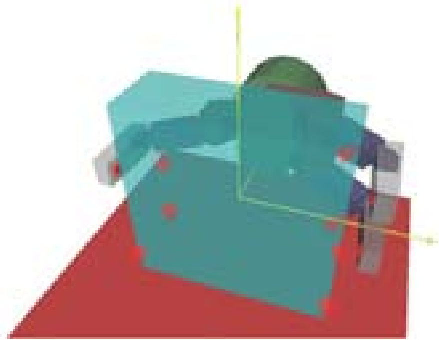


Fig. 4. The finger motion is stopped by reaction forces as the fingers (the proximal links) hit the box and prevent the sensors from touching the box.

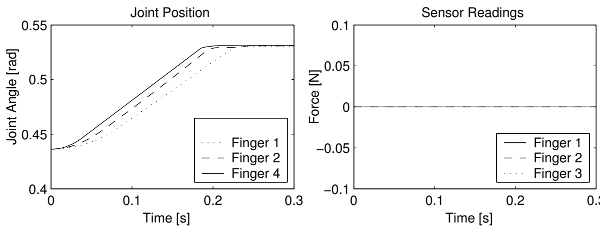


Fig. 5. The joint angles and the sensor values when not reaching the desired grasp.

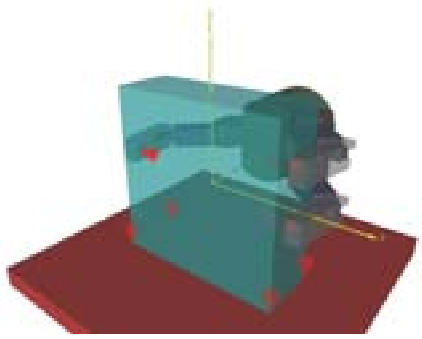


Fig. 6. The grasp is formed according to plan and all three sensor contact the box.

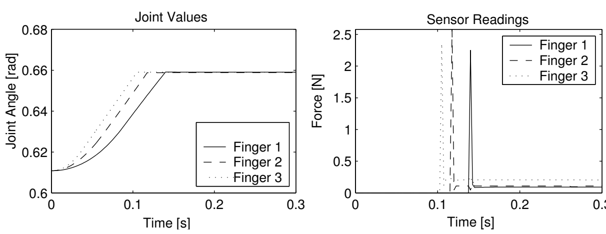


Fig. 7. The joint angles and the sensor values while reaching the intended grasp.

Productive robots and the SMErobot project

K. Nilsson[1], R. Johansson[2], A. Robertsson[2], R. Bischoff[3], T. Brogårdh[4], M. Hägele[5]

[1]Dept. of Computer Science , Lund Inst. of Technology, Sweden, klas@cs.lth.se
[2] Dept of Automatic Control, Lund Inst. of Technology, Lund University, Sweden
[3] KUKA Roboter GmbH, Germany
[4] ABB Automated Technologies Robotics, Sweden
[5] Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Germany

Extended Abstract

The need for keeping manufacturing and workplaces in Europe calls for new ideas and concepts for productive robots with focus on their usefulness for Small and Medium sized Enterprises (SMEs). The activities from the Robotics Research at LTH, [3], [4], [2], and in particular those related to the SMErobot project [1] comprise efforts in this direction. The SMErobot project is a recently started four-year-project of the 6th Framework Programme of the EC "to create a new family of SME-suitable robots and to exploit its potentials for competitive SME manufacturing". To this purpose the goal is not to create fully autonomous robots for all possible tasks, but rather to create semi-autonomous robots and to allow for human-robot interaction in a safe way. An introductory motivation and some of the main issues of the SMErobot project now follow, [1].

More than 228 000 manufacturing SMEs in the EU are a crucial factor in Europe's competitiveness, wealth creation, quality of life and employment. To enable the EU to become the most competitive region in the world, the Commission has emphasized research efforts aimed at strengthening knowledge-based manufacturing in SMEs as agreed at the Lisbon Summit and as pointed out at MANUFUTURE-2003. However, existing automation technologies have been developed for capital-intensive large-volume manufacturing, resulting in costly and complex systems, which typically cannot be used in an SME context. Therefore, manufacturing SMEs are today caught in an *automation trap*: they must either opt for current and inappropriate automation solutions or compete on the basis of lowest wages.

In total, the SMErobot initiative is set to offer an escape out of the automation trap through:

- Technology development of SME robot systems adaptable to varying degrees of automation, at a third of today's automation life-cycle costs;
- New business models creating options for financing and operating robot automation given uncertainties in product volumes and life-times and to varying workforce qualification.
- Empowering the supply chain of robot automation by focusing on the needs and culture of SME manufacturing with regard to planning, operation and maintenance.

One of the key components in this development is the possibility of humans and robots to share tasks and to share the same workspace in a safe way.

The Safe Human-aware Space-Sharing Robot: Currently, robots have to work behind fences to ensure operator safety because of risks due to high-energy motions and control/software faults. For SMEs – which typically are engaged in small-series production – this poses a significant problem in terms of installation cost and time and lack of interaction and tuning during production. Intrinsically safe robots are identified to be the cornerstone of future manufacturing concepts such as the space-sharing co-worker for human-robot cooperation. Initial research towards safe robot systems has mainly investigated safety issues of specific robot components. It is recognized that an intrinsically safe robot requires research towards safe mechanics, human motion perception, and safety-conformable layouts and controls. New robot structures and kinematical ideas for high-performance, low-inertia, low-cost robots are of major interest in this area, see Fig. 1. In applications where this is not physically possible, a complementary approach based on active dependable control and human-aware motions is needed.

Another main issue of the project is programming and task description.

The Robot Capable of Understanding Human-Like Instructions: Today, programming depends on explicit specification of coordinates, motion commands and process parameters. The programming effort is tremendous and drastically increases the life-cycle costs for a typical workcell. In the absence of highly skilled robotic programmers, relatively easy tasks take an average of 40 hours of programming for the average SME, but programming should be as simple

as telling a colleague to perform a certain task. Therefore, future robot instruction schemes require the use of intuitive, multi-modal interfaces and preferably human communication channels, such as speech and gestures. Identification and localization of work pieces, automatic generation or adaptation of programs and process parameters are also required for minimizing programming efforts.

Other issues include easy installation by Plug-and-Produce technologies, task description techniques, calibration, sensor-fusion, and generation of robot programs. The combination with safe robots that understand human-like instructions should form a new generation of robots, with the continued need for further development and integration with other types of robot technology.

There will be a need for robot work cells to adapt to variations in production flow or in maintaining the production rate in the presence of uncertainties. When the physical or logical layout of the production process has to be restructured or when new equipment is installed, SME employees, who today do not engineer robotized production, should be able to compose and configure the equipment. Work cells should be adaptable in case of work-piece variations or reconfigured when production volume or product changes occur. The goal is to create work cell designs resulting in "zero-changeover time" when adapting or reconfiguring to product variants, and the deployment of a typical robot work cell within *less than three days* (instead of weeks). Several concepts of self-configurable work cells have been suggested in the past with little success, mainly owing to high costs of the modular components, lack of compatibility with third-party components, shortcomings in the robot's intelligence for dealing with variable product geometries/locations/processes, and owing to lack of integration of planning and configuration tools within typical CAD systems.

Mechanisms and Architecture for Plug-and-Produce: To be able to plug and produce without operator intervention, there is a need for standardized protocols and software interfaces, which allow the automatic configuration of all components upon setup of the workcell. Grippers, tools, sensors, part transport and feeding devices have to be automatically interfaced to the cell, initiated and started. The consortium has the power to set European and worldwide standards for these interfaces and plug-and-produce mechanisms. The software mechanisms needed to support such radically improved flexibility differs from current practices in that fragile/hard-coded implementations are replaced by secure software techniques.

Robot Task Generation based on Product/Process Data: Programs and process information should be generated from CAD data, in addition to intuitive instruction schemes. Both user and CAD input need to be combined and converted into task instructions that can be understood by the robotic workcell. The generated motions from ideal CAD data have to be adjusted to current workpiece and process requirements. Thus, the adaptation of the robot requires adaptation and optimization of trajectories and process data. First algorithms are available, but need to be better integrated with sensor-based motion control and the programming system in general. The consortium has novel ideas, which make fast, easy and intuitive calibration of CAD data to cell equipment, tools and work objects possible without any measuring devices. The robot operator can make use of process models and CAD geometries without knowing anything about CAD systems or the process model representations. This also means full coherence between on-line and off-line representation, and SMEs can make direct use of process and geometry data from more advanced customers.

Fig. 1. Portable LTH-prototype of the Gantry-Tau parallel kinematic manipulator (PKM) concept, based on the ideas and patents of T. Brogårdh, ABB Robotics, see [5] and references therein.

Acknowledgment This work has been funded by the European Commission's Sixth Framework Programme under grant no. 011838 as part of the Integrated Project SMErobot.

References

[1] SMErobot (2005) `http://www.smerobot.org`
[2] `http://www.robot.lth.se`
[3] R. Johansson , A. Robertsson, K. Nilsson, T. Brogårdh, P. Cederberg, M. Olsson, T. Olsson, G. Bolmsjö, "Sensor Integration in Task-level Programming and Industrial Robotic Task Execution Control", *Industrial Robot: An International Journal*, Vol. 31, No. 3, pp. 284-296, 2004
[4] A. Blomdell, G. Bolmsjö, T. Brogårdh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K.Nilsson, M. Olsson, A. Robertsson, and J.J. Wang, "Extending an Industrial Root Controller – Implementation and Applications of a Fast Open Sensor Interface" (In press), *IEEE Robotics & Automation Magazine*, September 2005.
[5] H. Cui, Z. Zhu, Z. Gan, and T. Brogårdh, "Kinematic analysis and error modeling of TAU parallel robot" *Robotics and Computer-Integrated Manufacturing*, vol.21, No. 6 pp. 497–505, 2005

Session Multi Vehicles and Control

Symmetries in the Coordinated Rendezvous Problem

Ruggero Carli* Fabio Fagnani† Alberto Speranzon‡ Sandro Zampieri†

ABSTRACT

Coordination algorithms for cooperative autonomous vehicles has attracted large attention recently, e.g., [1, 2, 3]. For a survey see [4]. One of the control problems that have been considered is rendezvous namely, the problem of autonomously driving a number of vehicles to the same point. This is a fundamental coordination problem for multi-vehicle systems, and it can be seen as one of the simplest instances of formation control.

The aim of this contribution is to characterize the relationship between the amount of information exchanged by the vehicles and the control performance for the rendezvous. We would expect that a more connected communication network would yield a better performance.

The model we consider is as follows. We assume that each vehicle can communicate its position to other vehicles through a given communication network. We restrict our investigations to the case when the rendezvous point is the barycenter of the initial configuration of the vehicles, which can be considered as a minimum effort approach.

We assume that the vehicles are described by an elementary model, which allows us to investigate how communication affects the coordination. Let $N \geqslant 2$ be the number of vehicles. We model the vehicles as

$$x_i(t+1) = x_i(t) + u_i(t) \qquad i = 1, \ldots, N\,,$$

where $x_i \in \mathbb{R}$ is the x-coordinate of vehicle i. If we restrict to a linear state feedback, $u(t) = Kx(t)$, where $K \in \mathbb{R}^{N\times N}$, the system dynamics become

$$x(t+1) = (I+K)x(t)\,.$$

Under the assumption that $I+K$ is a doubly stochastic matrix, the vehicles rendezvous at the barycenter of the initial configuration. Such matrix can be interpreted as the transition matrix of a Markov chain, whose underlying directed weighted graph $\mathcal{G}$ represents the communication network.

In order to compare various feedback controllers that solve the rendezvous problem, we consider as performance index the spectral radius

$$\rho_{\mathcal{G}} = \inf_K \max\{|\lambda| : \lambda \in \sigma(I+K), \lambda \neq 1, K \in \mathcal{K}(\mathcal{G})\},$$

where $\mathcal{K}(\mathcal{G})$ is the family of all matrices $N \times N$ which are compatible with the graph $\mathcal{G}$. The spectral radius $\rho_{\mathcal{G}}$ bounds the convergence rate to the rendezvous point. Since $I+K$ represents a Markov chain, $\rho_{\mathcal{G}}$ is related to the mixing rate of such a chain. This connection allows us to use results available in the literature for bounding mixing rates of Markov chains (see for example [5] and references therein.)

Let us consider communication networks that are represented by graphs with symmetries. In particular, consider so called Cayley graphs on finite Abelian groups [6]. Figure 1 shows two examples of such graphs.

The main result, proved in [7], is the following. Let $\mathcal{G}$ be a graph as defined before, and let $\nu > 0$ be the number of fixed transmitting vehicles to each vehicle c.f., Figure 1. Then we can bound the spectral radius as

$$1 - \frac{C}{N^{2/\nu}} \leqslant \rho_{\mathcal{G}} \leqslant 1$$

*University of Padova, Department of Information Engineering, via Gradenigo 6/B, 35131 Padova, Italy, {carlirug,zampi}@dei.unipd.it

†Dipartimento di Matematica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129 Torino, Italy fabio.fagnani@polito.it

‡(Corresponding author) Royal Institute of Technology, Department of Signals, Sensors and Systems, Osquldas väg 10, 10044 Stockholm, Sweden, albspe@s3.kth.se

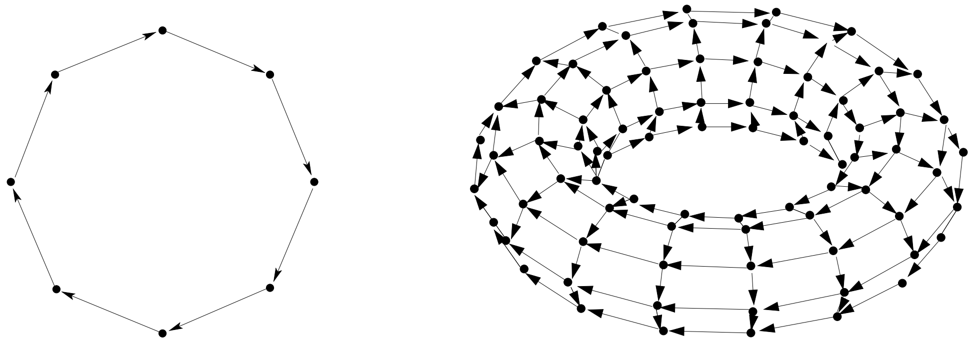


Figure 1: Examples of communication networks. On the left is shown a communication network in which the vehicle receives information from only one other vehicle, and thus the in-degree is $\nu = 1$. On the right is shows a toroidal communication network in which $\nu = 2$.

where $C > 0$ is a constant.

The bound shows that symmetries in the communication network yield rather poor performance in terms of convergence rate. As we can see, if we keep a constant in-degree ν, as N grows the spectral radius tends to 1. This means that the convergence to the rendezvous point becomes slower and slower as the number of vehicles grows.

The result of this contribution shows the importance of the structure of the communication network on the performance for systems with a large number of vehicles. In particular if the communication network has symmetries then the coordination strategies do not scale well with the number of nodes in the network.

References

[1] J. A. Fax and R. M Murray. Information flow and cooperative control of vehicle formations. *IEEE Trans. Aut. Cont.*, 49(9), 2004.

[2] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Aut. Cont.*, 49(9), 2004.

[3] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Aut. Cont.*, 48(6), 2003.

[4] A. Speranzon. On control under communication constraints in autonomous multi-robot systems. Licentiate thesis, Royal Institute of Tehcnology, 2004.

[5] L. Saloff-Coste. Random walks on finite groups. In *Encyclopaedia of Mathematical Sciences*. Springer, 2004.

[6] C. Godsil and G. Royle. *Algebraic Graph Theory.* Springer, 2001.

[7] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri. Symmetries in the coordinated rendezvous problem. Submitted to Workshop on Networked Embedded Sensing and Control, Notre Dame, USA, 2005.

Modification via averaging of passivity-based control for orbital stabilization

Anton S. Shiriaev and Leonid B. Freidovich
Department of Applied Physics and Electronics
Umeå University, SE-901 87 Umeå, Sweden
{Anton.Shiriaev,Leonid.Freidovich}@TFE.UMU.SE

I. Introduction

In certain industrial applications, autonomous robots are required to perform various cyclic motions. Dynamics of robots can be described by a set of nonlinear differential equations in Euler-Lagrange form with control inputs, corresponding to several mechanical degrees of freedom. Unfortunately, to show existence of periodic solutions rigorously only a few methods are known. Moreover, they are applicable only for special restricted classes of systems. Consequently, design of controllers that ensure practical orbital stabilization of periodic motions is a challenging problem. We consider this problem for the class of mechanical systems with all but one actuated degrees of freedom. The particular well-known examples from this class are a pendulum on a cart and Furuta pendulum,as well as the Pendubot and the Acrobot, two-link robots with actuators either only in the elbow or in the shoulder, respectively. We illustrate our approach on the example of *Furuta pendulum* and show some experimental results[1].

II. Description of idea

Recently, motivated by applications, several researchers have proposed an approach of feedback design that ensures creation and partial stabilization of a motion, periodic only with respect to a few practically essential degrees of freedom. The proposed methodology is based on the following idea. Instead of traditional motion planning (choosing desired trajectory in the form of periodic function of time), partial feedback linearization [4] is used to derive a family of differential equations, solutions of which define a set of desired trajectories for the essential coordinates. After that, passivity can be used for orbital stabilization of the desired periodic motion with respect to the corresponding subset of generalized coordinates [1]. The drawback of this approach is that, typically, the other coordinates are not guaranteed to be bounded. Escape of some coordinates to infinity is unacceptable for practical implementation. For known examples, it is not hard to show that proposed strategies do lead to unbounded solutions, except for the case when initial conditions satisfy certain restrictive relations. In this paper, we propose a way to introduce an addition to the earlier proposed control actions avoiding the escape problem. For our design we constructively employ an averaging method [3, Sec. 10.4].

III. Controlling Furuta pendulum

Like the other examples, mentioned above, the Furuta pendulum is a an underactuated two degrees of freedom mechanical system, see Fig. 1.

Fig. 1. Furuta Pendulum.

Here, the pendulum is freely attached to the rotating horizontal arm, which is directly actuated by a DC-motor. The motor produces torque, assigned through dSPACE® system by our program, coded in Simulink®. Our goal is to achieve stable oscillations of the pendulum around one of the equilibria.

Our control design is model-based. Ignoring fast dynamics of the motor and sensors, the motion of the Furuta pendulum can be described by a system of two coupled second order differential equations with respect to the angle of the arm, ϕ, and the angle of the pendulum, θ. The control torque, u, enters only the ϕ–equation.

[1]See [2] for movies of the experiments and information.

The first step of the design is a feedback transformation. We partially compensate the interactions between the links and obtain a new control input, which is a force that is applied to the pendulum. If this new input is kept zero, the pendulum would freely move independently from possible motions of the arm. In this case, theoretically, the pendulum would oscillate with an amplitude, depending on the initial position, provided friction forces are compensated completely.

Alternatively, we could continue with the second step. Following [1], the new input could be used in order to design passivity-based feedback control law that exponentially stabilizes certain level of the energy of the pendulum. Correspondingly, under this composite control the pendulum would oscillate with a desired amplitude, provided the quality of friction compensation is sufficient. However, due to the back force from the pendulum to the arm, the latter would rotate in the direction depending on initialization.

The third step of the design is our main contribution. We introduce an additional control action that is simultaneously small enough not to destroy the periodic motion of θ and large enough to eliminate the drift in ϕ and to make the arm oscillate around a given desired position. We construct an auxiliary time-periodic linear system that plays the role of nonstandard linearization along the desired periodic trajectory of θ. We furthermore simplify this system using the averaging technique and consequently obtain an averaged time-invariant linear system. Finally, for this system we use pole placement to design a regulator for ϕ. Employing singular perturbation technique, we show that constructed in this way additional control action ensures boundedness of the trajectories of the original nonlinear system and practical exponential orbital stabilization of the desired periodic motion of the pendulum.

The three-step-design procedure has been tested experimentally. The results for oscillations of our Furuta pendulum around the downward equilibrium are shown below. On Fig. 2, the desired limit cycle corresponds to the closed red curve and the system's trajectory for 30 first seconds are shown in blue. On Fig. 3, we show the control signal, applied to the arm, and the achieved oscillations of the arm around the desired angle of $\pi/2$ radians.

IV. Conclusion

In this paper, we have considered a problem of creating stable periodic motions in underactuated mechanical systems. We have proposed a way to introduce an addition to the earlier proposed passivity-based control actions. Our design is based on averaging and allows to avoid unboundedness of the solutions, which is important for practice. It should be noticed, however, that there is a price that we pay for the guarantee that all the generalized coordinates remain bounded. We only achieve exponential orbital stabilization for an unknown trajectory, periodic with respect to the essential coordinates, that is in a small vicinity of the desired motion. The theoretical idea has been implemented for Furuta pendulum example and shown to be effective and robust with respect to unavoidable uncertainties.

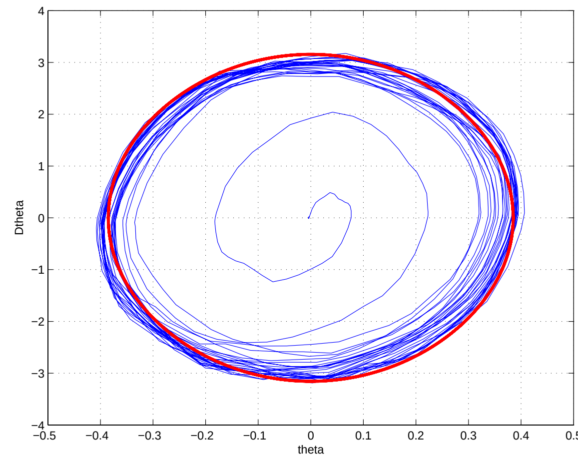


Fig. 2. Oscillations of the pendulum (θ vs. $\dot{\theta}$).

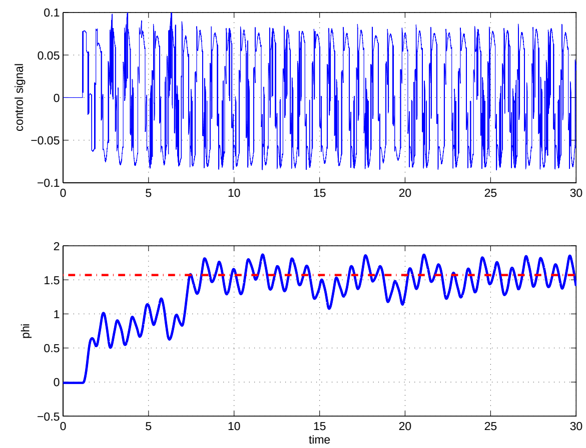


Fig. 3. Control signal and angle of the arm (u and ϕ).

References

[1] Aracil, J., Gordillo, F., Acosta, J.A., "Stabilization of Oscillations in Inverted Pendulum", 15th IFAC World Congress, Barcelona, Spain, 2002.

[2] http://www.tfe.umu.se/forskning/Control Systems/Set_Ups/Furuta_Pendulum/Furuta_Pendulum_info.html

[3] Khalil H.K., *Nonlinear systems,* Prentice Hall, 3^{rd} ed, 2002.

[4] Spong, M.W., "Partial Feedback Linearization of Underactuated Mechanical Systems", IROS'94, Munich, Germany, pp. 314-321, 1994.

Stabilization of Vehicle Formations - A Case Study

Ather Gattami, Johannes Berglund
Department of Automatic Control, Lund University
Box 118, SE-221 00 Lund, Sweden
E-mail: ather@control.lth.se

August 9, 2005

We consider a practicle example of stabilization of vehicle formations, namely six vehicles from the Multi-Vehicle Wireless Testbed (MVWT) used at Caltech. Powerful theoretical results on homogeneous interconnected systems are used for stability analysis and controller design. Each vehicle has a rectangular shape seen from above, with two fans to control its motion, see Figure 1. The nonlinear dynamics of the vehicle are given by

$$\begin{aligned} m(\ddot{r} - r\dot{\beta}^2) &= -\mu\dot{r} + (F_R + F_L)\cos(\theta - \beta) \\ m(r\ddot{\beta} + 2\dot{r}\dot{\beta}) &= -\mu r\dot{\beta} + (F_R + F_L)\sin(\theta - \beta) \\ J\ddot{\theta} &= -\mu r^2\dot{\theta} + (F_R - F_L)r_f \end{aligned} \tag{1}$$

The nonlinear dynamics are linearized and we obtain a linear system for the error dynamics which has two inputs, the fan forces F_R and F_L and two outputs, the polar coordinates r and β. The task is to stabilize all vehicles in a prespecified formation. There is no common coordinate-system. Each vehicle can only measure the relative distance to a limited number of other vehicles. Using the fact that the system is homogeneous, existing results from [1] can be used for *separately* finding a decentralized controller for every vehicle. We show stability for the case where the interconnection graph is given in Figure 2. Every node denotes a vehicle, and for instance, the graph shows that vehicle 1 can sense the distance to vehicle 2 and 6, vehicle 2 can sense the distance to vehicle 1 and 3, and so on. Other interconnections can also be used using the same

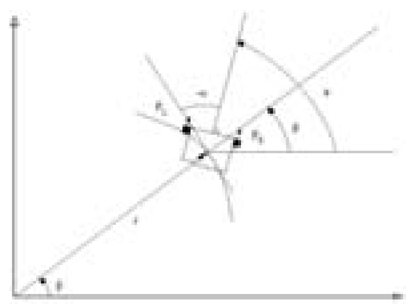


Figure 1: The Multi-Vehicle Wireless Testbed vehicle.

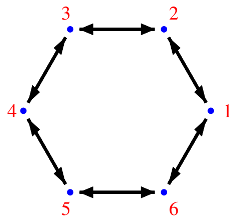


Figure 2: The interconnection graph.

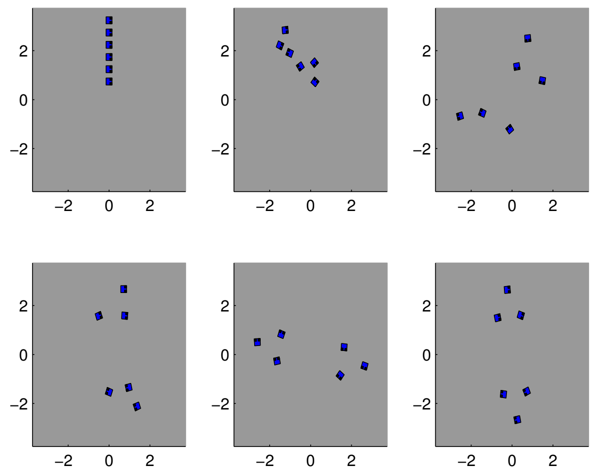


Figure 3: The vehicles starts in a row and the task is to make them rotate in two groups around a center that is agreed on online. The simulation shows that they end up rotating in the desired grouping counter-clockwise.

methods for analysis and controller design. A simulation is presented in Figure 3.

References

[1] Ather Gattami, Richard M. Murray. *A Frequency Domain Condition for Stability of Interconnected MIMO Systems*. In Proc. ACC 2004, Boston, USA.

[2] Johannes Berglund. *Stabilization of Vehicle Formations - A Case Study*. Master's Thesis ISRN LUTFD2/TFRT–5751–SE, June 2005. Dept. of Automatic Control, Lund university, Lund, Sweden.

[3] L. Cremean, W. Dunbar, D. van Gogh, J. Hickey, E. Klavins, J. Meltzer, R. M. Murray. *The Caltech Multi-Vehicle Wireless Testbed*. In CDC 2002.

Motion Planning and Dynamical Positioning for a Fleet of Underactuated Ships

Anton Shiriaev† Anders Robertsson‡ Leonid Freidovich†

†Department of Applied Physics and Electronics
University of Umeå, SE-901 87 Umeå, SWEDEN
{Anton.Shiriaev|Leonid.Freidovich}@tfe.umu.se

‡Department of Automatic Control, Lund Institute of Technology
PO Box 118, SE-221 00 Lund, SWEDEN
Anders.Robertsson@control.lth.se

I. Introduction

This paper is devoted to the problems of motion planning and feedback stabilization for dynamical underactuated ship. We use the models taken from [1], where it is suggested to describe the behavior of a ship by three independent variables: the angle that the ship makes with the direction to the *north* and and two coordinates that define the position of the ship in the horizontal plane, which are usually taken as the *north-east* coordinates.

It is important to notice that the dynamics of the ship differs from the dynamics of a rigid body on a plane. This is due to hydrodynamical effects (behavior of ambient water) and due to the presence of friction terms from the motion in the water with both linear and quadratic velocity dependencies.

It is assumed that the ship has only two control inputs that correspond to two generalized forces acting on the ship. They are the propeller thrust and the angle of the rudder. The absence of the 3rd control input makes the ship underactuated.

In this paper we consider a formation of several ships. We assume that the specifications for desired behavior for each ship or a family of ships is given by a geometrical path and an orientation along the path. In general, the specifications can be updated off-line or/and on-line. They are seen as implicit instrumental tools for further controller designs.

Projecting dynamics of the ship (the family of ships) onto such a geometrical path (paths), we obtain a description of all feasible motions of the ship (formation) along such path(s) without its explicit time parametrization. For the dynamical positioning problem, the search of equilibria of the projected dynamics on such path(s) and the analysis of their stability (instability) are of the highest interest. Below we briefly discuss possible solutions with different topologies of a ships' formation.

II. Projected Dynamics for One Underactuated Ship Model

We start with presenting a path planning problem for one ship with three degrees of freedom and two control inputs. Feasible motions of the ship model are computed for an illustrative example taken from [1, Example 10.1, p. 410]. It describes a high speed container ship of length $L = 175$ m and displacement volume $21,222$ m^3, which is equipped with one rudder and one forward thrust propeller.

Let

$$\eta = [n,\, e,\, \psi]^T \tag{1}$$

denote the North-East position and the yaw angle, respectively, and

$$\nu = [u,\, v,\, r]^T \tag{2}$$

be the velocity vector in the body frame. The kinematics equation is

$$\frac{d}{dt}\eta = R(\psi)\nu, \quad \nu = R(\psi)^T\frac{d}{dt}\eta, \tag{3}$$

where

$$R(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

is the rotation matrix in yaw. The surge speed equation and the steering equations (sway and yaw) are assumed to be decoupled. The ship dynamics, written in the body frame, are given by

$$M\dot{\nu} + N(\nu)\nu = B(\nu)\tau + R(\psi)^T w \tag{4}$$

where $\tau = [T,\, \delta]^T$ is the control input with T being the propeller thrust and δ being the angle of the rudder, w is the vector of environmental disturbances, the matrix functions M, N, and B are

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0_{1\times 2} \\ 0_{2\times 1} & I_{2\times 2} \end{bmatrix} \tag{5}$$

$$N(\nu) = \begin{bmatrix} -X_u - |u|X_{|u|u} & 0_{1\times 2} \\ 0_{2\times 1} & -\frac{U}{L}\begin{bmatrix} a_{11} & La_{12} \\ \frac{a_{21}}{L} & a_{22} \end{bmatrix} \end{bmatrix} \tag{6}$$

$$B(\nu) = \begin{bmatrix} (1 - t_d) & 0 \\ 0 & \frac{U^2}{L}b_{11} \\ 0 & \frac{U^2}{L^2}b_{21} \end{bmatrix} \tag{7}$$

Here $X_{(\cdot)}$, a_{ij}, and b_{ij} are the hydrodynamic coefficients[1], t_d is the thrust deduction number, $t_d \in (0,1)$, and $U = \sqrt{u^2+v^2}$ is the total speed.

We assume that the vector w of environmental disturbances in (4) is constant. Suppose that the control inputs T and δ are chosen to preserve a given path invariant for ship movements. We describe below all the possible motions of the ship.

Theorem 1: Suppose that the path and the yaw angle are defined by the C^2-smooth functions of the new independent variable θ:

$$n = \phi_1(\theta), \quad e = \phi_2(\theta), \quad \psi = \phi_3(\theta), \tag{8}$$

and that there exists a control input $\tau^* = [T^*, \delta^*]^T$, which makes the relations (8) invariant for the ship dynamics (4). Then θ is one of the solutions of the dynamical system

$$\alpha(\theta)\ddot{\theta} + \beta_1(\theta)\dot{\theta}^2 + \beta_2(\theta)\dot{\theta}|\dot{\theta}| + \gamma(\theta) = 0 \tag{9}$$

The explicit formulae of the functions $\alpha(\theta)$, $\beta_1(\theta)$, $\beta_2(\theta)$, and $\gamma(\theta)$ are given in the proof presented in [2]. ■

A test for asymptotic (in-)stability of equilibria of (9) is given in the next statement.

Theorem 2: Let θ_0 be an equilibrium point of the system (9), that is the point where $\gamma(\theta_0) = 0$. Suppose that the functions $\alpha(\theta)$, $\beta_1(\theta)$, $\beta_2(\theta)$, and $\gamma(\theta)$ are such that the constant

$$\omega_0 = \frac{d}{dt}\left[\frac{\gamma(\theta)}{\alpha(\theta)}\right]\bigg|_{\theta=\theta_0} \tag{10}$$

is positive and that the inequality

$$\frac{\beta_1(\theta_0)+\beta_2(\theta_0)}{\alpha(\theta_0)} > \frac{\beta_1(\theta_0)-\beta_2(\theta_0)}{\alpha(\theta_0)} \tag{11}$$

holds. Then, the equilibrium θ_0 of (9) is asymptotically stable. Moreover, if the sign of inequality (11) is opposite, then the equilibrium θ_0 of (9) is unstable. ■

Further details and a feedback design method for dynamical positioning of one ship can be found in [2].

III. Dynamic Positioning for Formations of Underactuated Ships

Equipped with the dynamical positioning algorithms elaborated for one ship, one can readily consider DP-problems for formations of ships with different topologies in communication and control strategies.

One of the solutions for an automatic reallocation and bias correction in a chain of pontoons used for shaping a landing run on sea is illustrated on Fig. 1 and 2. On the plots we show the process of building a formation of 3 ships. The first ship is assigned to remain on a particular geometrical path in the inertia frame. The two other ships form a chain behind the first one. The geometrical constraints for the second and the third ships are chosen as circles with the centers corresponding to the position of center of mass of the ship previous in the chain. In the beginning phase, the formation reaches the stable equilibrium formed by environmental forces w_1 acting from the west. After some time, during the second phase, the environmental forces abruptly change direction. As a result, the formation reaches a new stable equilibrium after transition.

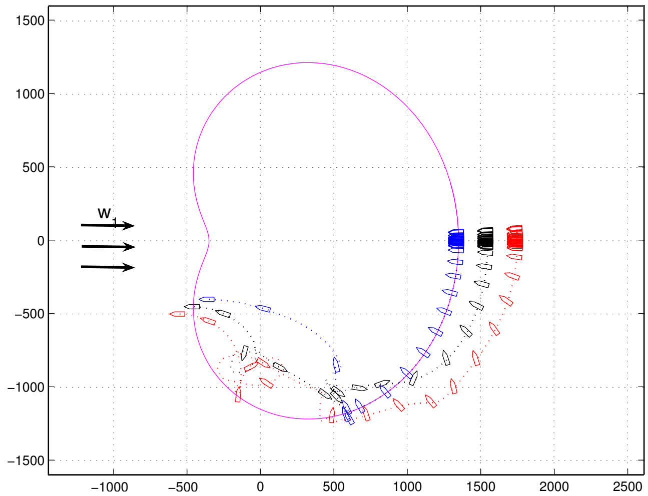


Fig. 1. The process of building a formation of 3 ships. The formation reaches a stable equilibrium due to environmental forces w_1, acting from the west.

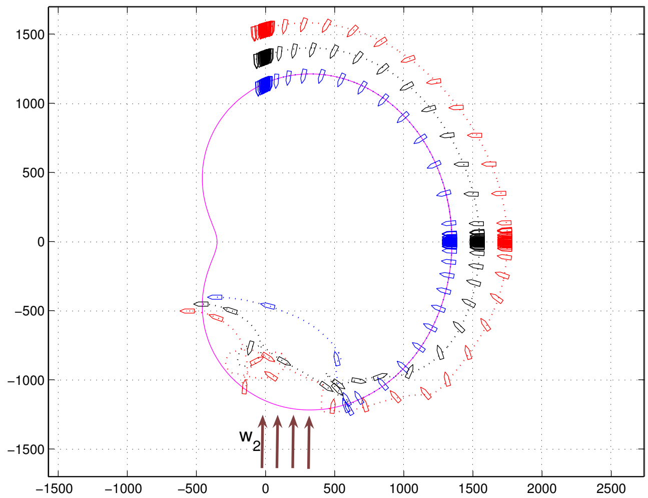


Fig. 2. During the second phase, the formation reaches another stable equilibrium due to environmental forces w_2, acting from the south.

[1] The numerical values for the example are $m = 21.2 \cdot 10^6$, $X_{\dot{u}} = -6.38 \cdot 10^5$, $a_{11} = -0.7072$, $a_{12} = -0.286$, $a_{21} = -4.1078$, $a_{22} = -2.6619$, $b_{11} = -0.2081$, $b_{21} = -1.5238$.

References

[1] Fossen T.I. *Marine Control Systems.* Marine Cybernetics, 2002.

[2] Shiriaev A., A. Robertsson, P. Pacull and T.I. Fossen. 'Motion Planning and its Feedback Stabilization for Underactuated Ships: Virtual Constraints Approach,' in the *Proceedings of the 16th IFAC World Congress*, Praha, 2005.

Flocking with Obstacle Avoidance: A New Distributed Coordination Algorithm Based on Voronoi Partitions

Magnus Lindhé and Karl Henrik Johansson
Signals, Sensors & Systems
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
lindhe@s3.kth.se, kallej@s3.kth.se

Petter Ögren
Autonomous Systems
Swedish Defence Research Agency
SE-164 90 Stockholm, Sweden
petter.ogren@foi.se

1 Introduction

We have studied the problem of navigating a group of autonomous vehicles in terrain with obstacles of general shape. The group should form a flock that maintains desired inter-vehicular distances while avoiding collisions with obstacles or each other when moving towards the goal. We propose a decentralized algorithm that only requires every vehicle to sense the locations of its neighbors within a limited radius. Our solution [1] is based on a modified navigation function [2] and a coverage control algorithm [3].

The navigation function, f, provides a computationally efficient way to achieve motion to the goal. It maps every obstacle-free point in the plane to the length of the shortest unobstructed path from there to the goal. In practice, only points in a rectangular grid are considered, and values for all other points are linearly interpolated. Computed this way, f has no undesired local minima and its negative gradient always points along the shortest way to the goal.

The coverage control algorithm relies on the concept of Voronoi regions. The Voronoi region associated with a vehicle can be described as the set of points that are closer to that specific vehicle than any other vehicle. An example is illustrated in Figure 1. In the original coverage control algorithm, a Voronoi region is updated continuously for each vehicle and the control makes the vehicle move to the centroid of its region. The centroid is computed using a density function, used to distribute the vehicles evenly for sensing purposes.

Our algorithm also computes the centroids, but we intersect the Voronoi regions with the obstacles and use f as the density function. This results in obstacle avoidance as well as protection against inter-vehicular collisions. Furthermore, we introduce artificial mirror neighbors to achieve formation maintenance when the vehicles are moving in an open space.

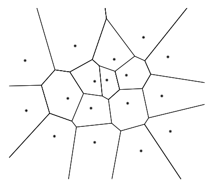


Figure 1: Example of Voronoi regions.

2 The Proposed Algorithm

We assume that every vehicle has sensors that can detect neighboring vehicles within a limited sensing radius. We also assume that the N vehicles have the discrete dynamics

$$p_i(k+1) = p_i(k) + u_i(k), \qquad i = 1, \ldots, N,$$

where u_i is bounded and finally that all vehicle controllers have global knowledge of the obstacles. This is not strictly needed, as the navigation function can be calculated based on partial knowledge and then updated as soon as a car detects a new obstacle.

Informally, our algorithm can be described as:

1. *[Sensor reading]* Use sensors to find the positions of neighbors within a limited sensing radius.

2. *[Mirror neighbors]* If the vehicle is not inside the convex hull of its neighbors, place mirror neighbors.

3. *[Voronoi region]* Derive the Voronoi region, based on real and mirror neighbors and obstacles.

4. *[Centroid]* Compute the centroid of the Voronoi region, weighted according to the navigation function.

5. *[Control actuation]* Move towards the centroid if it is closer to the goal than the present position, otherwise stand still. If the step to the centroid is longer than half the sensing radius, restrict the step length to that distance.

3 Properties of the Algorithm

The proposed algorithm offers a clear priority between the three desirable properties of goal convergence, collision safety and flock cohesion. Safety has the highest priority, since the vehicle stands still if the centroid is not deemed safe. No step longer than half the sensing radius is allowed, since this could lead to collisions between vehicles that have not detected each other. It can be proven that this algorithm is collision safe.

The next priority is given to goal convergence, since no steps that move vehicles farther from the goal are allowed. This will lead to crowding near narrow passages or sharp corners, but speeds goal convergence at the expense of maintaining desired inter-vehicular distances. Note that even if a vehicle stands still, the movement of others may create feasible points to go to in a subsequent step.

Finally, flocking has the lowest priority but when moving over open fields the group tends to a hexagonal lattice formation with the desired distance between all vertices. This is due to the mirror neighbor mechanism, that stops vehicles on the boundary of the flock from floating away from the others.

4 Simulation Results

We have simulated the proposed coordination algorithm in two settings. First we have studied the case with simple integrator vehicle dynamics and simulated a group of twenty vehicles in Matlab. The results illustrate group cohesion, goal convergence and obstacle avoidance in an environment with irregular and non-convex obstacles. Snapshots from one such simulation are included in Figure 2.

Secondly, we tested a more realistic setting with car-like vehicle dynamics. Using a hierarchical controller structure, a low-level controller drives the

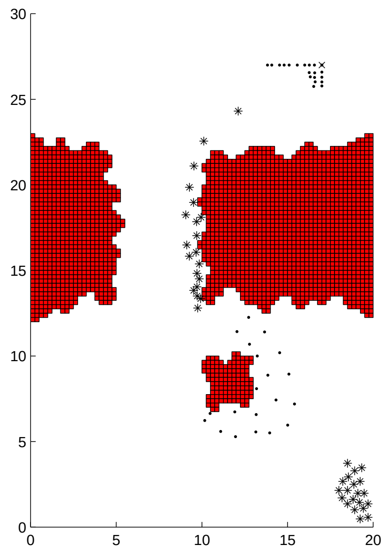


Figure 2: Four snapshots of twenty vehicles moving around irregular obstacles. The goal is in the upper right corner.

vehicles to waypoints computed by the high-level coordination algorithm, while for collision safety not leaving the Voronoi region. The simulations are documented as animated films with up to 20 cars, in the form of US Army HMMWV jeeps.

References

[1] M. Lindhé, P. Ögren, and K. H. Johansson. Flocking with obstacle avoidance: A new distributed coordination algorithm based on Voronoi partitions. ICRA, Barcelona, Spain, 2005.

[2] P. Ögren and N. E. Leonard. A convergent dynamic window approach to obstacle avoidance *IEEE Transactions on Robotics.* Vol. 21, Issue 2, April 2005.

[3] J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage Control for Mobile Sensing Networks *IEEE Transactions on Robotics and Automation.* Vol. 20, April 2004.

Session UAV

A low-cost UAV-system for forestry and agricultural applications

Olle Hagner[1] and Ola Friström[2]
[1]Remote Sensing Laboratory, Swedish University of Agricultural Sciences
[2]SmartPlanes AB

Personal Aerial Mapping System

A low-cost UAV-system specifically designed for applications in forestry and agriculture has been developed by SmartPlanes AB in close cooperation with the Remote Sensing Laboratory, Swedish University of Agricultural Sciences and the Regional Forestry Boards in Västra Götaland and Västerbotten.

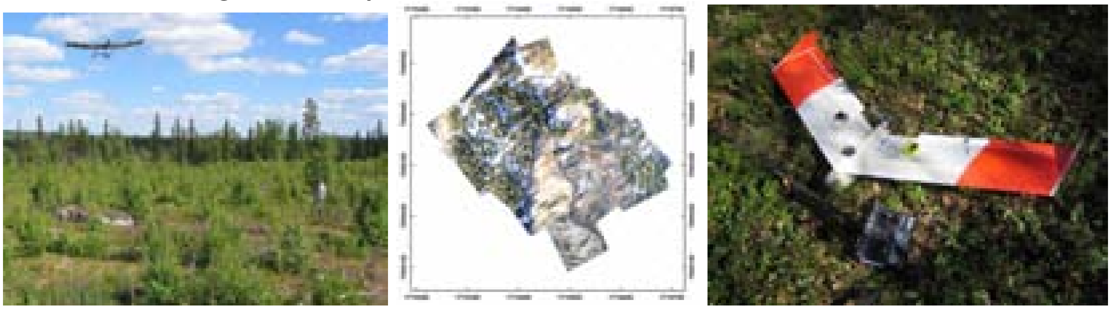


Figure 1. Forest mapping, orthophoto mosaic, SmartOne mini-UAV (prototype)

The design is based on a vision of a personal aerial mapping system (PAMS) consisting of a robust, easy-to-operate, safe and affordable airframe integrated with advanced photogrammetric software tools that enables the user to produce high quality photo maps direct on-site. Some examples of the application areas where the PAMS concept would be useful are:

- Crop monitoring
- High resolution forest mapping
- Detection and documentation of wind-throw
- Monitoring of regenerating forest
- Mapping in support of logging operations
- Detection of areas important for biodiversity

The main requirements dictated by the applications are:

- Safe operation
- Easy to use,
- Low-cost
- Colour video or high resolution digital camera and GPS
- Portable and operated by a single person
- Ability to operate from small clearings in forest terrain in adverse weather conditions

IFLYG

A research program *Intelligenta flygplan* IFLYG funded by Kempestiftelserna has been initiated around the PAMS concept involving a network of researchers, technicians, users in forestry and agriculture and industrial partners. The work is coordinated by SLU. Subprojects address topic such as:

- Avionics
- Photogrammetry
- Sensors
- Flight safety
- Mobil GIS
- Forestry applications
- Agricultural applications

SmartOne UAV

The use of unconventional materials and building techniques has resulted in a very rugged airframe that can be flown in strong winds, rain and snow. Low cost has been achieved by electric propulsion and off-the-shelf-components for radio-controlled model airplanes. Safety is provided by close-range and low altitude operation combined with low weight and speed.

The *SmartOne* UAV system has been approved by the Swedish aviation authorities for flight testing within 9 test areas in Sweden. Initial restrictions dictate line-of-sight operation and a maximum flying height of 300 meters outside densely populated areas. Personnel from the regional Boards of Forestry are completing pilot training and will participate in the forestry oriented flight testing program. Flight testing for agricultural applications is conducted in cooperation with Jordbrukstekniska forskningsinstitutet (JTI), Svenska Lantmännen AB and SLU in Skara.

Figure 1. Example pictures: Fire monitoring, alpine tree line in winter, mini-UAV hand launch, emerging crop, farm houses, spruce forest, new clear-cut, regenerating forest, industrial construction site, soccer field, grazing deer (late evening).

Using equivalence classes of multi-target paths for sensor plan evaluation

Christian Mårtenson **Pontus Svenson**
Dept of Data and Information Fusion
Swedish Defence Research Agency
SE 164 90 Stockholm
Sweden
cmart,ponsve@foi.se

June 20, 2005

We present a new approach to sensor resource allocation by simulating *equivalence classes* of future multi-target paths. Our method is meant to be used for pre-planning sensor allocations, *e.g.*, for choosing between several alternative flight-paths for UAV's, or deciding where to deploy ground sensor networks. The method can be used to choose the best sensor allocation with respect to any fusion method; it can also be extended to be used for threat prediction.

In addition to the list of sensor allocations to evaluate, the algorithm requires knowledge of the terrain/road network of the region of interest. Additional doctrinal knowledge on the enemy's possible goals and rules of engagement can increase the speed of the method, but is not required.

Given a current situation picture, the method generates possible future paths for the objects of interest. For each considered sensor allocation, these futures are partitioned into equivalence classes. Two futures are considered equivalent with respect to a given sensor allocation if they would give rise to the same set of observations. For each such equivalence class, we run a fusion algorithm; in this paper, an emulated multi-target tracker. We use the output of this emulated filter to determine a fitness for each sensor allocation under evaluation. For small scenarios, we compare some different ways of calculating this fitness.

In order to determine the best of the sensor allocation schemes, we simulate a possible future path of the units of interest and apply all the sensor schemes to it. For each sensor scheme, this gives us a list of simulated observations that we can input to a fusion module. For each fusion output, we calculate a fitness by comparing it to the "true" simulated future. By averaging over many possible future paths, we can determine a total fitness for each sensor allocation scheme. Equivalence classes of future paths are used to speed up the simulation of such future paths.

In order to handle multiple objects, we use random sets [1] and probability hypothesis densities in our formulation of the method. Using $z(x(t), s(t), t)$ to denote the set of observations given by the sensor scheme $s(t)$ at time t for unit positions $x(t)$, we define two possible future ground truths $\hat{x}_1(t)$ and $\hat{x}_2(t)$ to be equivalent with respect to the sensor allocation scheme $s(t)$ if $\mathbf{z}(\hat{x}_1(t), s(t), t) = \mathbf{z}(\hat{x}_2(t), s(t), t)$ for all t, and write that $\hat{x}_1(t) \sim_s \hat{x}_2(t)$.

What this means is just that if there is no way for our sensor and fusion system to differentiate between two different predicted paths, there is no sense in simulating them both. Instead of averaging over all possible futures, we average over all equivalence classes of futures

$$s^{\text{best}} = \arg\min_{s(t)\in S} \int_{\hat{x}(t)\in X_s} P[\hat{x}(t)] H(\hat{x}(t), s(t)) d\hat{x}(t)$$

where X_s denotes the partition of X induced by the equivalence relation $\sim_s$ defined above and H is a fitness measure.

This idea can be extended further. We could for instance relax the restriction that the observations must be equal for all t, and generate equivalence classes for specific time-windows. Another option would be to consider the threat posed by a future $\hat{x}$ and define two futures to be equivalent if they entail the same threat for us. This and other extensions of the ideas presented here will be described in detail elsewhere.

Note that the definition of equivalence classes as given above is only correct for sensors with detection probability $p_D = 1$. In the algorithm and experiments described below, where we use $p_D < 1$, we also condition on a specific sequence of random numbers used in determining the fictitious observations; this will guarantee that our equivalence classes are proper. Formally, we express this by including the seed of the random number generator used in the sensor allocation scheme $s(t)$. Another way of exploiting the ideas presented here would be to group $\hat{x}_1(t)$ and $\hat{x}_2(t)$ together if their corresponding fictitious observations $\mathbf{Z}(\hat{x}_1(t), s(t), t)$ and $\mathbf{Z}(\hat{x}_1(t), s(t), t)$ are "sufficiently similar". This would, however, not give equivalence classes, since transitivity would not hold in that case.

We use two different fitnesses, the XY-difference, D_{xy}, which measures the mean L_1 distance between $y(t)$ and all instances $\hat{x}(t)$ of $\mathbf{X}(t)$, $D_{xy} = \sum_{\hat{x}(t) \in X(t)} P(\hat{x}) \cdot \|\hat{x} - y\|_1$ and an entropy measure $H_y = -\sum_a y_a \cdot \log y_a$. (Note that care must be taken to normalize $y(t)$ to 1 before inserting into this formula.)

We use two different simulation methods, the Exhaustive which generates all possible equivalence classes, and a Monte Carlo algoritm which samples from such equivalence classes. Results from the evaluation of 100 randomly generated sensor allocations, using 100 Monte Carlo samples, are shown in figure 1. (This scenario takes place around Härnösand; more results from it as well as from others are given in [2].) The allocations are ordered after the results of the Exhaustive method run with the D_{xy} measure, which can be regarded as an exact result. The Exhaustive method is also the most compute intense. In a worst case scenario, the number of equivalence classes to consider grows exponentially both with the number of sensors and with the number of targets.

Two conclusions can be drawn from the results presented in [2]. The first is that the entropy measure H_y approximates the relative values of D_{xy} rather well. The second is that the Monte Carlo technique reduces the overall computation time without significant loss in precision. In other words, the Monte Carlo method using the H_y measure gives us an acceptable approximation in the Härnösand scenario, with a large decrease in computation time. The time complexity for this combination is linear in all input parameters.

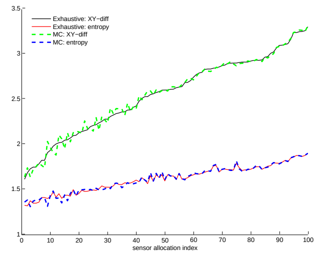


Figure 1: Results from the evaluation of 100 sensor allocations for the Härnösand scenario, ordered after the Exhaustive method with the XY-difference measure. The ranking induced by the entropy measure is similar to that induced by the XY-difference.

The method could also be used interactively by a user, who suggests partial or complete sensor schemes to the system. Here work needs to be done both on how to generate a complete plan from a partial one and on how to best interact with the user. It would be interesting to combine the method presented here with planning systems. Users can then input alternative plans and have them automatically evaluated by the computer system. This could be extended to be used for other things than sensor allocation, such as threat or possibility analysis, logistical planning, etc.

References

[1] I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen. *Mathematics of Data Fusion*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1997.

[2] Christian Mårtenson and Pontus Svenson. Evaluating sensor allocations using equivalence classes of multi-target paths. In *Proc 8th International Conference on Information Fusion*, 2005.

Combining Path Planning and Target Assignment to Minimize Risk in a SEAD Mission

Petter Ögren and Maja Winstrand
petter.ogren@foi.se
Department of Autonomous Systems,
Swedish Defence Research Agency (FOI),
SE-172 90 Stockholm, Sweden

***Abstract*— Suppression of Enemy Air Defense (SEAD) missions are dangerous and constitute the main tasks envisioned, together with precision strike, for the next generation of UAVs. Unmanned systems performing a SEAD missions must be able to autonomously plan and re plan their actions. When addressing planning problems, a common approach is to formulate optimization problems, and a natural choice of objective function is the risk. In this paper, two ideas are proposed to enhance the performance of a SEAD mission planning tool. The first is a natural way to propagate risk assessments from path segments to whole paths. In this way an estimate of the actual failure probability is minimized, not just some vague measure of badness. The second idea is a way to incorporate the offensive capabilities of the aircraft, i.e. target assignment, in the path planning. Simulation results are presented to illustrate the combined approach.**

I. Introduction

Today, there are a number of operational UAVs performing reconnaissance and surveillance tasks. The next generation of armed UAVs, so called Unmanned Combat Air Vehicles (UCAVs), [4], are intended to also be able to perform precision strike and Suppression of Enemy Air Defense (SEAD) missions.

In a SEAD mission, the objective is to reduce the threat from enemy surface to air missiles (SAMs) by forcing the enemy to shut his radars down and thus enable a safer passage. This is done by engaging the enemy radars controlling the SAMs with a signal seeking missile, e.g. a AGM-88 High-Speed Anti-radiation Missile (HARM) or a Marconi ALARM (Air Launched Anti-Radiation Missile). From this point on we will write HARM to mean either missile, for ease of notation. The word SAM is furthermore used to denote the entire surface to air missile system. When faced with a HARM the enemy SAM radar is likely to shut down to avoid being destroyed, thereby removing the threat posed by the SAM to the aircraft. Hence the term suppression of enemy air defense. In this paper we will focus on the mission planning part, i.e. the choice of flight path and targets to be engaged. We will not address the actual engagement nor the actions needed to negotiate unexpected events, such as pop-up threats, during the mission.

II. Main Results

Using the ideas of Zabarankin et al. [3], our goal is to formulate the SEAD problem as a so called Weight Constrained Shortest Path Problem (WCSPP).

Problem 2.1 (WCSPP): Given a graph $G = (V, E)$, weights $w_{ij} \in \mathbb{R}_+$, costs $c_{ij} \in \mathbb{R}_+$ and start and destination vertices s and d. The Weight Constrained Shortest Path Problem (WCSPP), is defined as follows:

$$\begin{aligned} \min_{P} \quad & \Sigma_{(i,j)\in E(P)} c_{ij}, \\ \text{s.t.} \quad & \Sigma_{(i,j)\in E(P)} w_{ij} \leq W, \\ & s, d \in P. \end{aligned} \tag{1}$$

That is, find the path P from s to d such that the sum of costs c_{ij} is minimized while the sum of weights w_{ij} is below the bound W.

Our task is now to establish a correspondence between the SEAD problem that we wish to solve and the WCSPP problem that we know how to solve. We can formalize the Pilot's SEAD problem as the following optimization problem.

Problem 2.2 (Formalized Pilot's Problem):

$$\begin{aligned} \max_{P} \quad & \Pi_{(i,j)\in E(P)} (1 - R_{ij}) \\ \text{s.t.} \quad & \Sigma_{(i,j)\in E(P)} w_{ij} \leq W \\ & s, d \in P, \end{aligned} \tag{2}$$

where R_{ij} and w_{ij} is the estimated risk and fuel consumption when flying from i to j. Furthermore, P is a path in the Voronoi graph constructed from the SAM locations.

Note that the fuel constraint and path criteria maps nicely into a WCSPP. The objective function is however a product above, reflecting the combined probability of surviving all the path segments, and not a sum of costs as in the WCSPP. In many papers, such as [2] and [3], this fact is ignored and the sum of kill probabilities, sum of inverse squared threat distances, or some other measure of badness, is minimized. Here however, we can find the solution that minimizes the actual risk estimate, by an elaborate choice of edge costs described in the following Lemma.

Lemma 2.3: Let $c_{ij} = log\left(\frac{1}{1-R_{ij}}\right)$. Then the Formalized Pilot's Problem (2), has the same solution as the WCSPP, (1).

Proof Note that log is a strictly increasing function and let "⇔" denote that two optimization problems have the same solution. Then $\max \Pi_{e_i \in Path}(1 - R_{ij}) \Leftrightarrow \max log(\Pi_{e_i \in Path}(1 - R_{ij})) \Leftrightarrow \max \Sigma_{e_i \in Path} log(1 - R_{ij}) \Leftrightarrow \min \Sigma_{e_i \in Path} - log(1 - R_{ij}) \Leftrightarrow \min \Sigma_{e_i \in Path} log(1 - R_{ij})^{-1}$, which is exactly the proposed c_{ij}. ■

What makes a UCAV different from a UAV is the fact that it has offensive capabilities, i.e. it can fire anti radar missiles such as the HARM.To include the HARMs we extend the statespace of the planning, i.e. the Voronoi graph, to reflect not only position of the aircraft but also the number of HARMS available. In other words, we stack a number of graphs on top of each other, each layer representing different number of HARMs. Firing a HARM thus corresponds to moving to the layer below the current one.

III. SIMULATIONS

We run the algorithm in a setting with the majority of SAM sites placed in a u-shape surrounding the destination. This setup makes to tactical sense, but was chosen for illustrative reasons. The WCSPP was solved using an approximation method based on the Dijkstra algorithm [6], and bisection search.

In Figures 1 and 2 the SAM radar ranges are indicated by circles, with the engaged ones crossed out. The distances are all in kilometers with SAM ranges of around 65km. In Figure 1 we ran the algorithm with one HARM

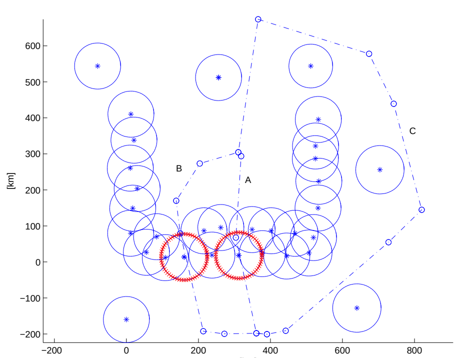


Fig. 1. Note the crossed out circles denoting the engaged SAMs.

available and upper range bounds of 600km, 1000km and 2000km respectively. The resulting paths were of (A) 541km with a success probability of 0.2, (B), 782km with a success probability of 0.8 and (C), 1714km with a success probability of 1.0. Note that all these probabilities are as accurate, or inaccurate as the individual segment risk estimates. In Figure 2 two HARMs are available and the upper bound is 600km. The resulting path length is 552km with an estimated success probability of 0.69. One might argue that the middle path segment should be somewhat more to the right, but the path can only be as good as the graph discretization.These simulation results concludes this extended abstract.

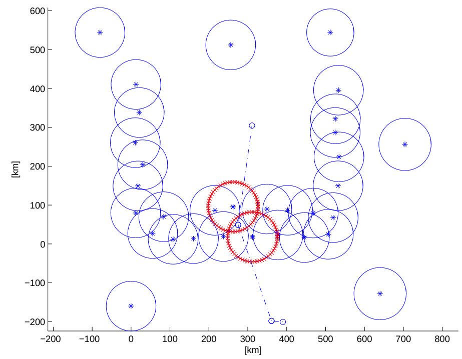


Fig. 2. Two HARMs and a range constraint of 600km.

IV. REFERENCES

[1] Irina Dumitrescu and Natashia Boland *Algorithms for the Weight Constrained Shortest Path Problem* International Transactions in Operational Research Volume 8 Issue 1 Page 15 - January 2001

[2] R.W. Beard, T.W. McLain, M.A. Goodrich, E.P Anderson, *Coordinated target assignment and intercept for unmanned air vehicles* IEEE Transactions on Robotics and Automation Volume: 18 Issue: 6 , Dec. 2002 Page(s): 911 -922

[3] Zabarankin, M., Uryasev, S., and P. Pardalos. *Optimal Risk Path Algorithms* R. Murphey and P. Pardalos (Eds.) Cooperative Control and Optimization. Kluwer Academic Publishers, 2002, 273-303

[4] Darpa Joint Unmanned Combat Air Systems (J-UCAS) program, (WWW page, cited 10 january 2005) http://www.darpa.mil/j-ucas/.

[5] M. Winstrand. *Mission Planning and Control of Multiple UAVs*, FOI report, FOI-R–1382–SE, October 2004.

[6] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization Algorithms and Complexity*. Dover Publications, Inc, 1998.

On-line trajectory planning using adaptive temporal discretization

David A. Anisi†
Optimization and Systems Theory
Royal Institute of Technology
100 44 Stockholm, Sweden
anisi@math.kth.se

***Abstract*— Direct methods for trajectory optimization are traditionally based on *a priori* temporal discretization and collocation methods. In this work, the problem of node distribution is formulated as an optimization problem, which is to be included in the underlying NLP. The benefits of utilizing such adaptive temporal discretization method for trajectory optimization, are illustrated by a missile guidance example.**

I. INTRODUCTION

Consider the following optimal control problem (OCP):

$$\begin{array}{llcl} \min & J & = & \int_0^T \mathcal{L}(x,u)\mathrm{d}t + \Psi(x(T)) \\ \text{s.t.} & \dot{x} & = & f(x,u) \\ & g(x,u) & \leq & 0 \\ & \Psi_i(x(0)) & \in & S_i \subseteq \mathbb{R}^n \\ & \Psi_f(x(T)) & \in & S_f \subseteq \mathbb{R}^n, \end{array}$$

where the state $x \in \mathbb{R}^n$, the control $u \in \mathbb{R}^m$, and the constraints $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$. Assuming the problem originates from a complex, real-world application, the existence of analytical solutions is disregarded, thus seeking fast computational algorithms for solving the OCP. To this end, the infinite-dimensional problem of choosing a control function, u, in a given space, have to be *transcribed* into a finite dimensional non-linear mathematical programming problem (NLP). This work focuses on *direct* transcription methods based on *temporal discretization*. In most direct methods (see *e.g.* [1]), the transcription is achieved by *a priori* partition of the time interval into a prescribed number of subintervals whose endpoints are called *nodes*. Generally, trajectory optimization run-times are critically depending on the number of variables in the NLP. These in turn, are proportional to the number of nodes or collocation points[1], N, in the temporal discretization. Therefore, it is extremely important to keep N as low as possible when aiming at constructing computationally efficient methods for trajectory optimization. In most cases however, there exist a trade-off between accuracy and efficiency in terms of computational effort. Here-within, the focus is on improving accuracy for a given efficiency requirement. More precisely, once the number of collocation points has been decided, the question of optimal node distribution is raised.

Although adaptive grid methods have been an active field for the last couple of decades, to the best of our knowledge, utilizing them for trajectory optimization has not been considered elsewhere.

In what follows, we advocate that in any computational efficient method, node distribution should be a part of the optimization process and show that the receding horizon (RH) approach is merely an outcome of such a paradigm. In Section III, the benefits of utilizing the suggested method are confirmed by a missile guidance example.

II. ADAPTIVE TEMPORAL DISCRETIZATION

It is a well-established fact in numerical analysis, that a proper distribution of grid points is crucial for the accuracy of the approximating solution. By concentrating the nodes and hence computational effort in those parts of the grid that require most attention (*e.g.* areas with sharp non-linearities and large solution variations), it becomes possible to gain accuracy whilst retaining computational efficiency.

In general, there exist three types of grid adaption techniques [2]. However, as trajectory optimization run-times are critically depending on N, it is the idea of maintaining a fixed number of nodes, but relocating them strategically over the interval that suits the on-line trajectory optimization problem best (this is referred to as r-refinement). To this end, let $p = [t_1, \dots, t_N] \in \mathbb{R}^N$ denote a partition of $[0, T]$,

$$0 = t_1 < t_2 < \cdots < t_{N-1} < t_N \leq T.$$

Adaptive grid methods are then based on either *equidistribution* of a *monitor function*, or *functional minimization* (FM) [2], [3]. The equidistribution principle (EP) requires a chosen positive definite monitor function (or weight), w, to be equidistributed over all subintervals. Mathematically, the EP can be expressed in various equivalent forms, *e.g.*:

$$m_i(p) = \int_{t_i}^{t_{i+1}} w\mathrm{d}t - \frac{\int_0^T w\mathrm{d}t}{N-1} = 0,\ i = 1, \dots, N-1, \text{or}$$
$$m_i(p) = \int_{t_{i-1}}^{t_i} w\mathrm{d}t - \int_{t_i}^{t_{i+1}} w\mathrm{d}t = 0,\ i = 2, \dots, N-1.$$

Commonly employed monitor functions include the "arclength monitor function", $w = \sqrt{\varepsilon + \dot{x}^2}$, and "curvature monitor function", $w = (\varepsilon + \ddot{x}^2)^{\frac{1}{4}}$. Regardless the choice of w, we remark that node allocation by the EP, can be determined by imposing a number of *grid constraints*, $m(p) \leq 0$. The functional framework to grid generation (FM), is based on the principle of specifying a measure of the grid quality. Traditionally, principles as smoothness, orthogonality and clustering properties of the grid are included in the functional, $I(p)$ [3]. Minimizing $I(p)$, will produce an optimal partition with respect to the chosen grid quality measure.

† Support by the Department of Autonomous Systems at the Swedish Defence Research Agency (FOI), is gratefully acknowledged.

[1] Since the nodes and collocation points have the same cardinal number, they are here-whithin considered to be *conceptually* equivalent.

Based on the two existing frameworks for adaptive grid generation (EP and FM), we now outline a generalized approach. The idea is to formulate the problem of collocation point distribution as a constrained optimization problem:

$$\begin{array}{ll} \min & I(p) \\ \text{s.t.} & m(p) \le 0, \end{array} \tag{1}$$

which is to be augmented with the underlying NLP. From (1) it is plainly seen that EP and FM are merely special cases of the suggested approach. We conclude this section by giving examples of the usage of this approach.

Example 1: Setting $d_i = t_{i+1} - t_i, i = 1\ldots, N-1$, the solution to the following optimization problem:

$$\begin{array}{lll} \min & I(d) & = \sum_{i=1}^{N-1} d_i - \varepsilon \ln d_i \\ \text{s.t.} & m(d) & = \sum_{i=1}^{N-1} d_i - T \le 0 \quad (d_i \ge 0), \end{array}$$

is the equidistant RH discretization scheme with ε deciding the step length (and hence planning horizon). This follows since if $(N-1)\varepsilon \le T$, then

$$\nabla_i I(d) = 1 - \frac{\varepsilon}{d_i} = 0 \Longrightarrow d_i = \varepsilon.$$

Example 2: The linear constraint

$$m(d) = \sum_{i=1}^{\varepsilon_1(N-1)} d_i - \varepsilon_2 T \le 0,$$

reflects the objective of distributing ε_1 parts of the nodes in the first ε_2 parts of the time interval.

III. Design Study: Missile Guidance

The 3DoF equations of motion in the pitch plane consider the rotation of a body-fixed coordinate frame, (X_b, Z_b) about an Earth-fixed inertial frame, (X_e, Z_e). The governing dynamic equations are

$$\begin{aligned} \dot{u} &= \frac{F_x}{m} - qw - g\sin\theta \\ \dot{w} &= \frac{F_z}{m} + qu + g\cos\theta \\ \dot{q} &= \frac{M}{I_y} \\ \dot{\theta} &= q \\ \dot{x}_e &= u\cos\theta + w\sin\theta \\ \dot{z}_e &= -u\sin\theta + w\cos\theta, \end{aligned}$$

where u and w are the X_b and Z_b components of the velocity vector, x_e and z_e denote the position of the missile in the inertial frame (X_e, Z_e), q is the pitch angular rate, θ denotes the pitch angle, m is the missile mass, g is the gravitational force, while I_y denotes the pitching moment of inertia. The system inputs are the applied pitch moment, M, together with the aerodynamic forces, F_x, F_z, acting along the X_b and Z_b axis respectively. During the simulations, we set $m = 204.02\ kg$, $g = 9.8\ m/s^2$ and $I_y = 247.437\ kg\, m^2$.

Referring to Fig. 1 and 2, the first simulation shows the terminal guidance part of a missile trajectory optimization problem. The missile starts off horizontally from $(0, 10)$ aiming at a target in $(700, 0)$ with terminal aspect angle $-\frac{\pi}{2}$. Fig. 1 depicts the reference trajectories with the missile velocities (in the inertial frame) indicated by small line segments.

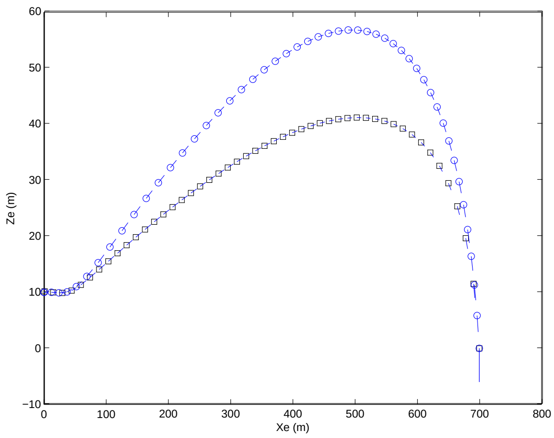


Fig. 1. Reference trajectories: static (○) and adaptive (□).

In the adaptive case, an EP based on the arclength monitor function together with a linear $I(p)$ is used. Seeing beyond the unequal axis scales, the nodes have been distributed more evenly over different path segments. In fact, there are 7 nodes/100 m path segment in the adaptive case, while the same figure varies between $5-13$ in the static case.

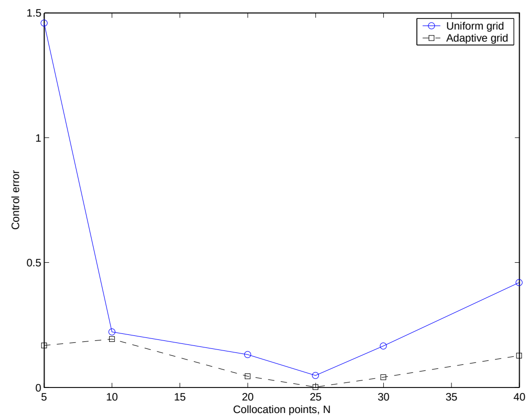


Fig. 2. The accuracy of the approximating control.

Fig. 2 shows the optimal control approximation error as a function of N. It can be noted that, for a given N, the extra degree of freedom provided by distributing the nodes is used constructively to improve accuracy. This illustrates the soundness of the proposed approach.

References

[1] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, Mars–April 1998.

[2] K. Blake, "Moving mesh methods for non-linear parabolic partial differential equations," Ph.D. dissertation, Department of Mathematics, the University of Reading, U.K., 2001.

[3] S. Kennon and G. Dulikravich, "Comparison of grid generation techniques," in *IEEE Fall Joint Computer Conference*, Dallas, TX, USA, 1986, pp. 568–575.

Creating Quality Imagery and Video from Inexpensive UAV Technology - through determining Camera Motion and 3D Structure

Harald Klomp Jakob Sandström

harald@cb.uu.se jakob@cb.uu.se

Abstract

Due to the small size and limited payload capability of mini-UAV systems mechanical stabilization of sensors is often not a feasible option. Turbulence is often significant at the typical operating height of most mini-UAV systems and cost requirements usually dictate the use of non-metric cameras. By consequence the raw imagery acquired is often characterized by unstable orientation, motion blur, high noise levels and significant optical distortions. Hence, extensive processing has to be applied in order to meet the user requirement of crisp stabilized video and high resolution image mosaics with map compatible geometry. The focus of this thesis is to explore methods and techniques used in Computer Vision and Photogrammetry in order to retrieve geometric information and camera motion for ortorectification and mosaicking. We propose a simple linear stabilization of the video, compensate for the optical distortion introduced by the cameras used and propose SIFT feature detection, RANSAC feature matching and global bundle adjustment as the solution to the problem of extracting geometric information from an unordered sequence of aerial imagery.

Summary in Swedish

Inledning

Vi har inom ramen för detta examensarbete testat metoder för att utifrån bilder tagna med ett flygbaserat kamerasystem generera bildmosaiker och stabila video bilder. Bilderna till detta examensarbete är tagna från ett litet obemannat flygplan kallat SmartOne. SmartOne är utvecklat för IFLYG projektet för att ta bilder på i första hand skog och jordbruksområden och är utvecklat av företaget SmartPlanes i samarbete med Sveriges Lantbruksuniversitet i Umeå. Flygplanets position kan bestämmas på ett ungefär med GPS men dess orientering i förhållande till marken är okänd. Detta examensarbete ingår som en del av forskningsprojektet IFLYG på SLU.
IFLYG består av delprojekten flygplansteknik, fotogrammetri, stödsystem, säkerhet, sensorer. Inom delprojektet flygplansteknik har obemannade flygplanet SmartOne tagits fram. SmartOne väger ca ett kg och har ett 120 cm vingspann. SmartOne flyger på mindre än 300 meters höjd vilket med en fyra megapixel kamera ger ca fem centimeter markupplösning. En GPS baserad autopilot skall också testas. Vårt examensarbete hamnar inom fotogrammetridelprojektet där planets tredimensionella rörelse tas fram. Kamerarörelsen används sedan för att kunna generera stabiliserad video och bildmosaiker. Stödsystem tas fram för att bland annat hantera bilder, flygloggar. Säkerhet är viktigt för att kunna erhålla flygcertifikat för flygplanet SmartOne. Olika sensorer skall prövas både optiska och multispektrala bildsensorer.

Teori
Eftersom orienteringen på planet inte är känt kan varje bild matcha ett antal andra bilder i serien. Bilder matchas mot varandra i en flerstegsprocess. I första steget matchas bilder mot varandra i tidsföljd. Varje bild antas överlappa föregående och nästa bild. Bilden analyseras och områden med bra mönster väljs ut. Förflyttningen av dessa bildpunkter från en bild till nästa kan sedan tas fram. Det är viktigt att mönstret har sådan egenskap att riktningen på rörelsen kan fås fram. Om mönstret är en linje eller en yta med jämn färg kan rörelsen inte bestämmas, detta kallas för bländarproblemet. När bildpunkternas rörelse är känt kan två efterföljande bilder passas ihop. När fler än fem bildpunkter har matchas ihop kan kamerans förflyttning och rotation beräknas. För att täcka ett större område än kameran kan se på en gång flygs flygplanet fram och tillbaka i stråk. I vårt fall är dessa stråk inte nödvändigtvis inte utan flygoperatören kan flyga fram och tillbaka i ett godtyckligt mönster. När flygplanet ha vänt fotograferar den delvis samma punkter på marken som tidigare. Att bestämma hur två flygstråk passar ihop är viktigt för att kunna skapa en sammanhängande bildmosaik. För att kunna göra detta använder vi en metod där områden med bra mönster enligt tidigare kriterier tilldelas en vektor av tal som beskriver mönstret. Vektorn kan sedan användas för att matcha två mönster mellan två bilder. När ett tillräckligt antal mönsterpunkter i två bilder matchar varandra görs ett försök att passa ihop bilderna. Om bilderna även passar ihop geometriskt antas de matcha varandra och den relativa kamera förflyttning och kamera rotation beräknas. Efter att ha passat ihop bilderna parvis så måste en globalt optimal lösning för alla bilder hittas. Varje kamera förflyttning ger sex frihetsgrader, tre för rotation av kameran och tre för den tre dimensionella förflyttningen. Kvadraten på avståndet mellan alla matchande mönsterpunkter beräknas. Förflyttningen och rotationen av kamerorna anpassas sedan så att summan av kvadratavståndet mellan alla matchande bildpunkter minimeras. Minimering av denna funktion görs stegvis genom att hela tiden stega sig i funktions gradientriktning. Denna metod kallas för block triangulering och är en standard metod för att passa ihop ett stort antal bilder samtidigt.

Resultat och diskussion
Vi har hittat och testat metoder för att matcha bilder i tidsföljd och mellan flygstråk som beskrivits i teoridelen. Bilder kan matchas ihop helt automatiskt och utifrån detta kan kamerans tredimensionella rörelse med avseende på förflyttning och rotation tas fram. Eftersom flygplanet var helt nytt hade vi bara tillgång till en begränsad uppsättning flygbilder. Därför återstår tester med fler bilder på bland annat jordbruksmark och annan tänkbar vegetation. En fullständig integration av alla metoder behöver också göras för att få en enhetlig helautomatisk produkt. Slutsteget att gå från kamera rörelsen för varje bild till att skapa ortofoto återstår också att utveckla men här finns standardmetoder att använda. Tester har också utförts på video bilder och här kan beräkning av bildrörelsen användas för att stabilisera skakningar i videon. Utifrån de bilder och flygmönster vi fått är vår slutsats att automatisk bildmatchning är möjligt och att kamerans rörelse kan beräknas även om enbart en ungefärlig GPS position är känd. Kommersiella system som SLU har prövat kräver att bildpunkter väljs manuellt och att närmevärden på både kamerans position och rotation skall vara känd innan. Jämfört med detta ger de metoder som vi hittat mycket snabbare resultat då manuell hantering kan minimeras.

Information-Theoretic Approach for Concurrent Path and Sensor Planning for a UAV with EO/IR sensors

Per Skoglar, Jonas Nygårds, Morgan Ulvklo, Petter Ögren
E-mail: {persko, jonny, morgan, petter.ogren}@foi.se
Swedish Defence Research Agency (FOI)

2005

In this paper, we propose a framework for autonomous path and sensor planning of a UAV, performing a surveillance and exploration mission. Concurrent path and sensor planning is a very challenging problem, since realistic models of platforms, sensors and environment are very complex and inherently non-linear. Our framework is one attempt to address these issues and formulate an optimization problem using tools from information theory.

Concurrent Path and Sensor Planning

At FOI, in a project called *Autonomous information driven control*, an information-theoretic approach to concurrent path and sensor planning for a UAV with gimballed EO/IR sensors is developed. The goal is autonomuos UAV surveillance, i.e., search for objects along a road or in a certain area and localization of discovered targets. The work is inspired by research in optimal observer trajectory computations for bearings-only tracking.

The optimization parameters are waypoints of the UAV flight path and the sensor gazing directions. An information matrix is constructed from the states representing targets and area points. A utility function is defined based on the information matrix and the planning process determines the optimization parameters that maximize the utility at the planning horizon. The method is applied to single target localization, n target localization, and area exploration. The method can handle limited field-of-view sensors and occlusion.

Simulation Example

The scenario is based on a real world environment, Figure 1 (left). The task is to survey a road segment and a building, and to detect and localize an object on the road. A number of area points along the road and around the building are set out. The number of area points is a compromise between long computational time and fair area coverage. An occlusion model is associated with each area point, and points on the edge of a forest and around the building are partly occluded.

Figure 1 (middle) shows, besides the vehicle path, the gaze direction of the sensor at some points. The vehicle is flying round the building as the sensor is pointing at the building. This is reasonable; according to the occlusion definitions of the area points, the sensor can not ”see” all area points from just one direction. Figures 1 (right) shows a snapshot when the sensor is gazing at the building.

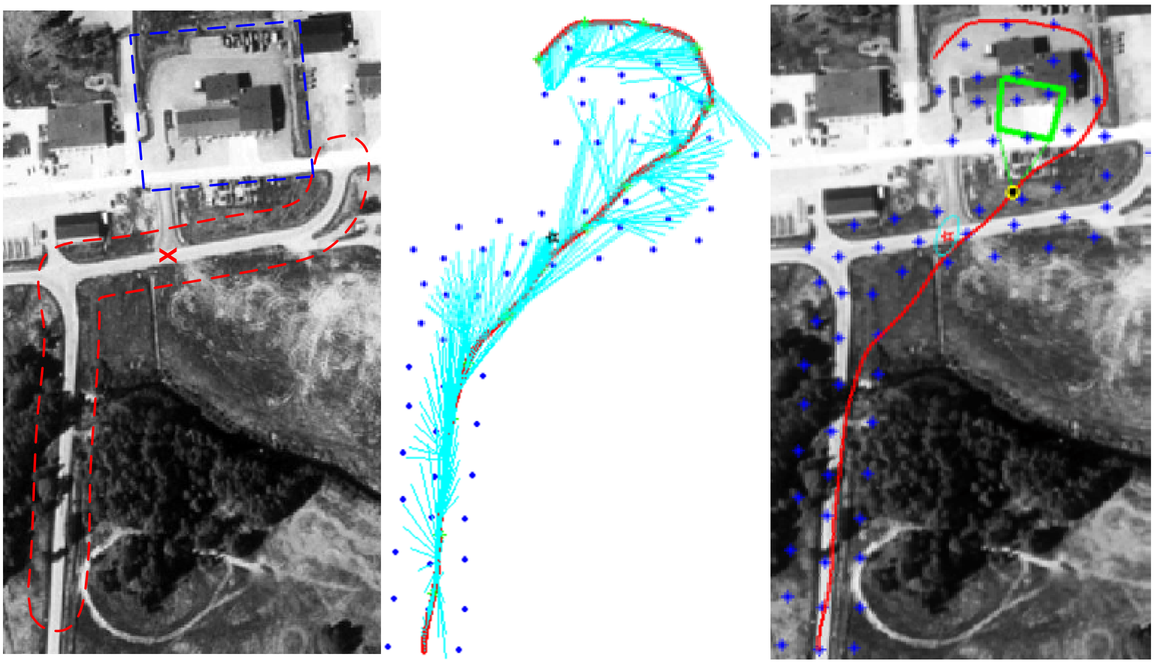


Figure 1: *Left:* Road surveillance scenario. A road segment (red lines) and a building (blue rectangle). In addition, there is an object on the road (cross). *Middle:* The UAV path and the sensor gazing directions. *Right:* The sensor is gazing at the building, the green square represent the sensor footprint.

Long Term Planning

A monolithic planner that can handle arbitrarily scaled scenarios is probably impossible to develop; instead a hierarchical decomposition is necessary. The issue then, is how to decompose the problem into sub-problems that guarantee that the overall objective is achieved. Development of a long term path planning method is in progress. Figure 2 shows an example of a graph approach called *Weight Constrained Shortest Path Problem* (WCSPP).

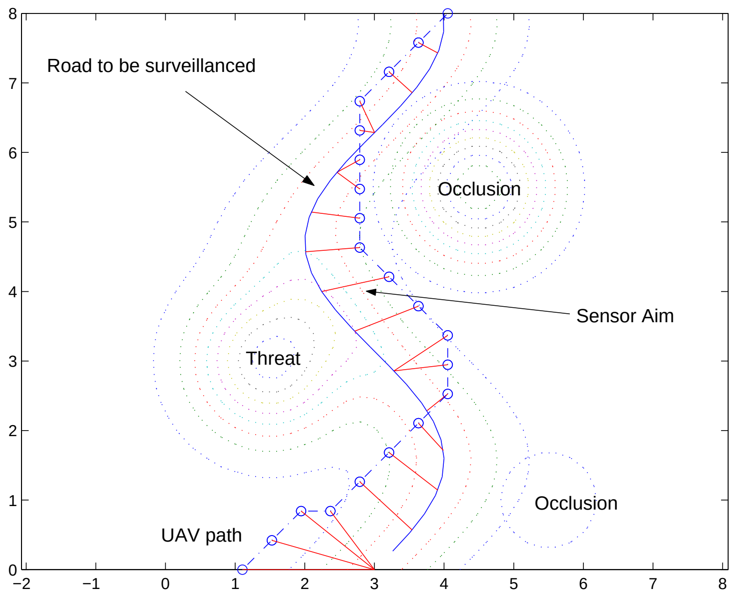


Figure 2: Output of the high level path/sensor planner. Note the effect of the threat zones and the low visibility areas.

Reconfigurable Path Planning for an Autonomous Unmanned Aerial Vehicle

Mariusz Wzorek and Patrick Doherty
Department of Computer and Information Science
Linköping University, SE-58183 Linköping, Sweden
{marwz,patdo}@ida.liu.se

Abstract

In this paper, we present a motion planning framework for a fully deployed autonomous unmanned aerial vehicle which integrates two sample-based motion planning techniques, Probabilistic Roadmaps and Rapidly Exploring Random Trees. Additionally, we incorporate dynamic reconfigurability into the framework by integrating the motion planners with the control kernel of the UAV in a novel manner with little modification to the original algorithms. The framework has been verified through simulation and in actual flight. Empirical results show that these techniques used with such a framework offer a surprisingly efficient method for dynamically reconfiguring a motion plan based on unforeseen contingencies which may arise during the execution of a plan.

1 Introduction

The use of Unmanned Aerial Vehicles (UAVs) which can operate autonomously in dynamic and complex operational environments is becoming increasingly more common. While the application domains in which they are currently used are still predominantly military in nature, in the future we can expect widespread usage in the civil and commercial sectors. In order to insert such vehicles into commercial airspace, it is inherently important that these vehicles can generate collision-free motion plans and also be able to modify such plans during their execution in order to deal with contingencies which arise during the course of operation. Motion planners capable of dynamic reconfiguration will be an essential functionality in any high-level autonomous UAV system. The motion planning problem, that of generating a collision-free path from an initial to goal waypoint, is inherently intractable for vehicles with many degrees of freedom. Recently, a number of sample-based motion planning techniques [2, 4] have been proposed which tradeoff completeness in the planning algorithm for tractability and efficiency in most cases. The purpose of this paper is to show how one can incorporate dynamic reconfigurability in such motion planners on a deployed and fully operational UAV by integrating the motion planner with the control kernel of the UAV in a novel manner with little modification of the original algorithms. Integrating both high- and low-end functionality seamlessly in autonomous architectures is currently one of the major open problems in robotics research. UAV platforms offer an especially difficult challenge in comparison with ground robotic systems due to the often tight time constraints present in the plan generation, execution and reconfiguration stages in many complex mission scenarios. It is the intent of this paper to show how one can leverage sample-based motion planning techniques in this respect, first by describing how such integration would be done and then empirically testing the results in a fully deployed system. The techniques and solutions described are generic in nature and suitable for platforms other than the one used in this experimentation. An important point to note is that to our knowledge we are the first to use these sample-based motion planning techniques with fully deployed UAVs.

2 The Path Planning algorithms

The problem of finding optimal paths between two configurations in a high-dimensional configuration space such as a helicopter is intractible in general. Sample-based approaches often make the path planning problem solvable in practice by sacrificing completeness and optimality. Two implementations of such algorithms are used in the WITAS [1]system: probabilistic roadmaps (PRM) and rapidly exploring random trees (RRT).
The PRM planner is an extended version of the standard algorithm proposed in [2]. It creates a roadmap in the offline stage based on the 3D model of the environment which is used later during the query phase. Our extentions to the original algorithm deal with problems of nonholonomic constraints and delayed constraints handling.
The use of rapidly exploring random trees (RRT) provides an efficient motion planning algorithm [4] that constructs a roadmap online rather than offline (PRM). After the roadmap is created, the remaining steps in the algorithm are the same as with PRMs.
The mean planning time in the current implementation for both planners is below 1000 ms and the use of runtime constraints do not noticeably influence the mean [5].

[1]WITAS is an acronym for the Wallenberg Information Technology and Autonomous Systems Lab which hosted a long term UAV research project (1997-2004).

3 Path execution mechanism

The standard path execution scheme in our architecture [3] for static operational environments is depicted in Fig. 1. A UAV mission is specified via a task procedure

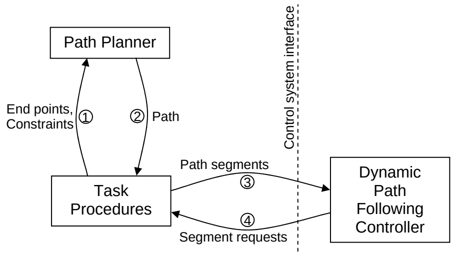


Figure 1: Plan execution scheme

(TP) in the reactive layer of our architecture, (perhaps after calling a task-based planner). A TP is a high-level procedural execution component which provides a computational mechanism for achieving different robotic behaviors.

For the case of flying to a waypoint, an instance of a navigation TP is created. First it calls the path planner service (step 1) with the following parameters: initial position, goal position, desired velocity and additional constraints.

If successful, the path planner (step 2) generates a segmented cubic polynomial curve. Each segment is defined by start and end points, start and end directions, target velocity and end velocity. The TP sends the first segment (step 3) of the trajectory via the control system interface and waits for the *Request Segment* event that is generated by the controller. At the control level, the path is executed using a dynamic path following controller [1] which is a reference controller that can follow cubic splines. When a *Request Segment* event arrives (step 4) the TP sends the next segment. This procedure is repeated (step 3-4) until the last segment is sent. However, because the high-level system is not implemented in hard real-time it may happen that the next segment does not arrive to the control kernel on time. In this case, the controller has a timeout limit after which it goes into safety braking mode in order to stop and hover at the end of the current segment. The timeout is determined by a velocity profile, current position and current velocity. In practice the time between receiving *Request segment* event and the controller timeout is large enough to reconfigure the path using the standard path planner. The updated segments are then sent to the DFP controller transparently.

Reconfiguration is triggered by the event created when new forbidden regions are added or deleted. There are several different policies that can be used during the reconfiguration step (Fig. 2):

Policy 1

Reconfiguration is done from the next waypoint (start point of the next segment) to the end point. This implies longer planning times and eventual replacement of collision-free segments.

Policy 2

Segments up to the colliding one are left intact and reconfiguration is done from the last collision-free waypoint to the end point.

Policy 3

Replanning is done only for colliding segments. The helicopter will stay as close to the initial path as possible. Note that each of these policies progressively re-uses more

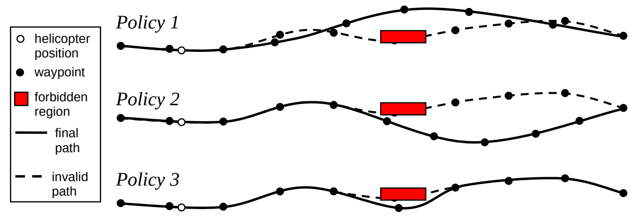


Figure 2: Replanning policies.

of the plan originally generated, thus cutting down on planning times. We are currently experimenting with all three policies.

4 Experimental results

In our experiments we have used both the PRM and the RRT planner with a TP that implements the first policy. Forbidden regions were randomly added by the ground operator during the flight. Typical helicopter velocity was up to $7m/s$ and the total path length up to $500m$. The results of the experiments show that the time window between sending two successive segments is generally greater than four times the amount of time required to generate full plans using either the PRM or RRT planners.

References

[1] G. Conte, S. Duranti, and T. Merz. Dynamic 3D Path Following for an Autonomous Helicopter. In *Proc. of the IFAC Symp. on Intelligent Autonomous Vehicles*, 2004.

[2] L. Kavraki *et al.* Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces. *Proc. of the IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[3] P. Doherty *et al.* A Distributed Architecture for Autonomous Unmanned Aerial Vehicle Experimentation. In *Proc. of the Int. Symp. on Distributed Autonomous Robotic Systems*, pages 221–230, 2004.

[4] J. J. Kuffner and S. M. LaValle. RRT-connect: An Efficient Approach to Single-Query Path Planning. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 995–1001, 2000.

[5] P-O Pettersson. Using Randomized Algorithms for Helicopter Path Planning. *Lic. Thesis Linköping University.* To appear, 2005.

List of Participants

Achim Lilienthal	Örebro U.	Leo Koikeroinen	BAE Systems Bofors AB
Albert Speranson	KTH	Leonid Freidovich	Umeå U.
Anders Engman	FOI	Magnus Lindhe	KTH
Anders Lennartsson	FOI	Maja Karasalo	KTH
Anders Robertsson	LTH	Marcus Berglund	LiTH
Anders Sandberg	Umeå U.	Mariusz Wzorek	LiTH
Andrew Wallace	Frontios	Martin Hagström	FOI
Anton Shiraev	Umeå U.	Martin Nilsson	SICS
Ather Gattami	LTH	Martin Norsell	FHS
Bengt Boberg	FOI	Martin Persson	Örebro U.
Björn Löfgren	Skogforsk	Michael Rotkowitz	KTH
Boris Duran	Halmstad U.	Morgan Ulvklo	FOI
Boyko Iliev	Örebro U.	Nils Axel Andersen	DTU
Carl Lundberg	MHS	Nils Hulth	Rotundus
Charlotte Averman	FOI	Ola Bengtsson	Halmstad U.
Christian Carling	FOI	Ola Ringdahl	Umeå U.
Christian Mårtensson	FOI	Olle Hagner	SLU
Christoffer Wahlgren	Örebro U.	Patric Jensfelt	KTH
Daniel Aarno	KTH	Patrick Doherty	LiTH
Daniel Skoog	FOI	Patrik Lif	FOI
David Anisi	KTH	Paul Sundvall	KTH
Elin Anna Top	KTH	Pelle Carlborn	Saab Bofors Dynamics
Erik Skarman	Saab AB	Per Bodin	Swedish Space Corporation
Fredrik Gustafsson	LiTH	Per Ljunggren	Intelligent Machines
Fredrik Heinz	LiTH	Per Skoglar	FOI
Gunnar Hovmark	FOI	Peter Alvå	FOI
Hans Jander	FOI	Peter Strömbäck	FOI
Harald Klomp	UU	Petter Ögren	FOI
Jakob Sandström	UU	Piotr Rudol, y	LiTH
Johan Hamberg	FOI	Pontus Svensson	FOI
Johan Larsson	Örebro U.	Roland Lenain	Cemagref
Johan Larsson	Örebro U.	Rolf Johansson	LTH
Johan Tegin	KTH	Thomas Hellström	Umeå U.
Jonathan Borgvall	FOI	Thomas Svantesson	Saab Bofors Dynamics
Kalle Johansson	KTH	Tom Duckett	Örebro U.
Kalle Prorok	Umeå U.	Torbjörn Martinsson	Volvo Wheel Loaders AB
Kristian Lundberg	Saab Bofors Dynamics	Torsten Merz	LiTH
Lars Andersson	LiTH	Viktor Kaznov	Rotundus
Lars Asplund	Mälardalen U.	Åke Wernersson	LTU

FOI is an assignment-based authority under the Ministry of Defence. The core activities are research, method and technology development, as well as studies for the use of defence and security. The organization employs around 1350 people of whom around 950 are researchers. This makes FOI the largest research institute in Sweden. FOI provides its customers with leading expertise in a large number of fields such as security-policy studies and analyses in defence and security, assessment of different types of threats, systems for control and management of crises, protection against and management of hazardous substances, IT-security and the potential of new sensors.

FOI
Defence Research Agency
Systems Technology
SE-164 90 Stockholm

Phone: +46 8 555 030 00
Fax: +46 8 555 031 00

www.foi.se