

Smooth Task Switching through Behaviour Competition

Philipp Althaus and Henrik I. Christensen

*Centre for Autonomous Systems, Numerical Analysis and Computer Science,
Royal Institute of Technology (KTH), S-10044 Stockholm, Sweden*

Abstract

Navigation in large-scale environments is composed of different local tasks. To achieve smooth switching between these tasks and thus a continuous control signal, usually a precise map of the environment and an exact pose estimate of the robot are needed. Both are hard to fulfill for experiments in real-world settings. We present a system that shows how one can relax the need for accurate metric models of the environment while at the same time achieving smooth task switching. To facilitate this scheme the dynamical systems approach is used, which incorporates behaviour coordination through competition in a dynamic framework. Feature detectors use sonar data to provide means for local navigation. This ability combined with a simple topological map constitutes a complete navigation system for large-scale office environments. Experiments showed that a Scout robot using this scheme is able to successfully navigate through our whole institute. Through the use of the dynamic behaviour coordination, switching between the navigational tasks occurs in a smooth manner leading to continuous control of the platform.

Key words: behaviour coordination, mobile robots, robot navigation, dynamical systems, hybrid deliberative systems

1 Introduction

A complete navigation system of an autonomous robot is traditionally of a hybrid nature. It consists of a reactive part, which allows to deal with unforeseen local events. Then, there is a deliberative part, which takes care of global issues like path planning. See [3] for an overview of different architectures.

Email addresses: philipp@nada.kth.se (Philipp Althaus), hic@nada.kth.se (Henrik I. Christensen).

The deliberative part is making use of a map to allow navigation in large-scale structures. For the implementation of this map and its use there are two conceptually different approaches, using either an exact geometric map or a qualitative map (e.g. topological map).

An exact geometric map (for example, a grid map [11] or a feature map [10]) allows to determine an appropriate control action at any robot position, which leads to smooth control and trajectories. However, this precise map first has to be obtained and the exact robot position has to be known at all times. These conditions are hard to fulfill and might be computationally very expensive. In contrast, the use of a topological map requires only an approximate guess of the robot pose; also the small-scale geometry does not have to be known precisely. Through sensory information from the robot it is determined when the robot has reached a node in the map. This information is enough to switch between the necessary behaviours to deal with the navigation task at hand. The Polly system [5], for example, shows how successful navigation can be achieved using this simple type of qualitative map. However, through discrete switching between tasks the control signals are not continuous. Furthermore, also the trajectory of the robot is not smooth anymore.

In this paper we propose a unified control scheme combining the advantages of each of the two approaches above. We present a navigation system for office environments using feature detectors which extract representations from sensory data. This geometric place recognition facilitates safe local navigation. To capture the large-scale properties of the environment a simple topological map is used. Nevertheless, through smooth task switching in a dynamical system framework, continuity in the control signal can be achieved. The dynamical systems approach to behaviour based robotics introduced by Schöner and Dose [15] provides this necessary framework. It utilizes a competition scheme among individual behaviours. So far, this competition has only been exploited in simulation work [15,16,9] using simplified small-scale settings. The system proposed in this paper, however, acts in a large-scale real-world environment using a topological map containing the global information necessary for navigation. This information about the whereabouts of the robot is still of a discrete nature; nevertheless, it is only used as parameters in the dynamic coordination among the behaviours. This dynamic system then ensures smooth switching between tasks. In this way the discrete nature of a qualitative map is combined with the need for continuous navigation in a unified control scheme. In essence, the system presented shows that one can relax the need for continuous pose estimation while maintaining a possibility for smooth behaviour switching.

Initially, the topological map facilitating task switching is introduced in Section 2. Then, in Section 3 the geometric representations used for local navigation are presented. The dynamical systems approach is outlined in Section

4, followed by the design of our coordination system. Finally, an example experiment is presented in Section 5, while a summary and avenues for future research are outlined in Section 6.

2 The Topological Map and its Use

The topological map is used by the system to switch between different tasks, in essence to activate an appropriate set of behaviours. Hence, it contains qualitative information about the large-scale structure of the environment. This information is reflected in nodes and edges that connect these nodes. The nodes stand for important places in the environment. There has to be one in front of each door, at each corridor crossing and at other places of interest (e.g. goal locations and charging station). Each node has a location in a fixed coordinate system. The edges that connect these nodes can be of three different types: room, corridor, door. Due to the limited amount and simplicity of information in the topological map, it is a matter of minutes to construct a new one for a previously unknown office environment. Figure 1 shows the topological map of our institute. To plan a path between two arbitrary nodes, the system conducts a breath first search through the graph. This is a feasible solution for environments of the size of an office area.

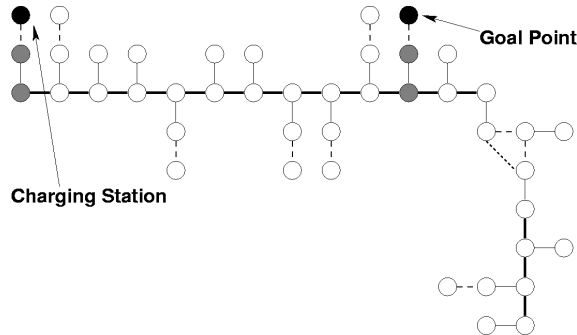


Fig. 1. The topological map of our institute. The circles in the map depict nodes, which have an exact location in a coordinate system. Edges are of three different types: corridor (thick line), room (dashed line), and door (thin line). Additional nodes for goal points and starting positions can be added arbitrarily. The nodes denoted with “Charging Station” and “Goal Point” correspond to the initial and final location of the experiment described in Section 5. The nodes in gray are the ones used to execute this plan.

At plan execution, it is assumed that the initial position and orientation of the robot are known (e.g. charging station). From there odometry is used to determine the robots location. This introduces errors in the estimation of the exact position of the robot, but is totally sufficient to determine if the system is in the vicinity of a node. This information is enough to evoke the

appropriate behaviours, which in turn are capable of dealing with the small-scale structures of the environment (see Section 4.2). Note that the error in the position estimate would grow bigger than desired on long trials over a great distance. To avoid this deficiency, the pose estimate is corrected based on detected features (see Section 3).

3 Extracting Geometric Representations from Raw Sensor Data

In our experiments we used the Scout robot from Nomadic Technologies (Figure 2). The platform has a cylindrical shape with a diameter of 38 cm and moves at a speed of up to 1 m/s. The robot is equipped with a ring of 16 evenly spaced ultrasonic sensors. The perception and geometric reconstruction of obstacles, walls and doors is based solely on the information provided by these sensors. Each sonar has a beam width of 25° and a detection range of 6 to 255 inches. The robot possesses a two wheel differential drive located at the geometric center which allows omni-directional steering at zero turning radius. The wheels are equipped with encoders to obtain odometric data.

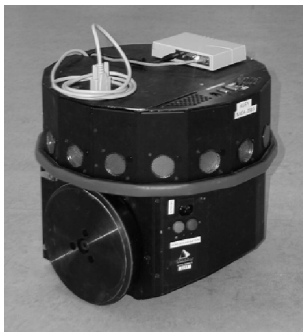


Fig. 2. The Scout robot used in the experiments.

The control system consists of five different behaviours: CORRIDOR FOLLOWING, WALL AVOIDANCE, DOOR PASSING, OBSTACLE AVOIDANCE, and GO TO. These behaviours ensure safe navigation given an individual task. Hence, they all need some geometric representation of the local properties of the environment.

To navigate along a corridor two behaviours have been designed: CORRIDOR FOLLOWING and WALL AVOIDANCE. They are based on the orientation of the corridor and the distance to its walls. To obtain this information the 200 most recent sonar readings are kept in a FIFO buffer. A Hough transform [4] is invoked on the sonar data every few seconds in order to extract the pair of parallel lines (one on either side of the robot) that coincide with the largest number of sonar echos. No assumptions on the width or direction of the corridor are made.

The behaviour `DOOR PASSING` guides the robot safely through an arbitrary, large enough opening in a wall. In order to find a door, when the robot finds itself in a corridor, the direction to the detected corridor wall is used. The 25 most recent sonar readings, that lie in the direction of this wall and not more than 50 cm behind it, are kept in a FIFO buffer. The largest angular segment (from the robots point of view) that does not contain any sonar reading is determined. If this segment is greater than 15° we consider a door to be detected and its direction is defined as the centre of the free segment. This process is invoked at every control cycle of the robot. Note that this door detector is very crude, due to the simplicity of the sensors used. Especially half blocked doors, with passages that are too small to pass, will sometimes still be detected as doors. However, situations like this are resolved by the coordination among a door passing and an obstacle avoidance behaviour (see the experimental results in Section 5). Further, if the robot is in a room the same strategy to detect a door is applied. However, first the wall at which the door is located has to be extracted. In order to do this a Hough transform is invoked on the 100 most recent sonar echos.

Due to the limited angular resolution of sonar sensors, the geometric representation of obstacles (used by the behaviour `OBSTACLE AVOIDANCE`) is rather simple and closely linked to the actual perception of the robot. Out of the 50 most recent sonar readings that do not belong to detected walls, the ones in the frontal half plane of the current robot heading are considered. Obstacles are reconstructed from these detected echos in ascending order of their distance to the robot. The echo closest to the robot defines the first obstacle whose orientation in the robot frame is given by the axis of the sensor that received the echo. A new obstacle is recorded for every subsequent echo whose orientation differs by an angle of at least 22.5° from any previously identified obstacle. New obstacles are added in an incremental fashion until the sonar buffer contains no further echos. Obstacle reconstruction is invoked at every control cycle of the robot. Notice, that our representation only considers the distance to an obstacle but ignores its shape or size. Despite its simplicity, the chosen representation is powerful enough to successfully navigate in cluttered areas [2,1].

Each of the above detectors keeps a certain number of the most recent sonar readings in a FIFO buffer. While collecting these readings the robot is driving a short distance. Odometry is used, to calculate the location of sonar readings taken at different robot positions, which introduces further uncertainty in the sonar data. These errors, however, are comparatively small and hardly influence the performance of the behaviours.

The behaviour `GO TO` aligns the robots heading with the direction of a goal point. We do not use any detector for this goal point yet; its location is only defined by a node in the topological map (Section 2).

The map not only facilitates switching between different tasks, it is also used to determine which of the above detectors should be invoked. The information about the exact location of its nodes and the robots position estimate (from odometry) determine if the robot finds itself in a corridor or in a room and/or close to a door. The information from the detected features, in turn, is used to update the position estimate. Otherwise, the error in this estimate may grow bigger than desired using only odometry. Hence, in a corridor the robots computed orientation and position are adjusted relative to the corridor walls each time the Hough transform is invoked (i.e. every few seconds). In addition, every time a door is passed orientation and position relative to the door posts can be updated accordingly.

4 Behaviour Coordination

The dynamical systems approach has been used to design the individual behaviours and their interaction. The conceptual framework of this approach is based on the theory of nonlinear dynamical systems [13]. In Section 4.1 we only provide a brief outline of this framework and refer the interested reader to [16] for a more detailed description. Section 4.2 describes the design of our system, especially the dynamic coordination of the individual behaviours. Note that both, the behaviours and their coordination, are implemented in a unified framework to provide a continuous control signal at all times. The parameters of this framework are anchored in the topological map (Section 2) and in the features extracted from the sensory data (Section 3).

4.1 Dynamical Systems Approach

A behaviour b emerges from the time evolution of the *behavioural variables* described by the vector \vec{x} . In a navigation task for example the robot heading and velocity constitute the set of behavioural variables. In the dynamical system described by

$$\dot{\vec{x}} = \vec{f}_b(\vec{x}) \quad (1)$$

the function \vec{f}_b can be interpreted as a *force* acting on the behavioural variables. This force is designed such that the desired values of \vec{x} (e.g. direction of a target) form an attractor and undesired values (e.g. direction of an obstacle) form a repeller in the dynamics of the behavioural variables. The function \vec{f}_b depends on the relative pose between the robot and its environment. However, the dynamics of \vec{x} takes place on a much faster time scale than the gradual changes that emerge in \vec{f}_b as a result of the robot's motion. This property assures that the dynamic variables remain close to the attractor state at all

times. Multiple behaviours are aggregated by weighted addition of the individual contributions \vec{f}_b :

$$\dot{\vec{x}} = \sum_b |w_b| \vec{f}_b(\vec{x}) + \text{noise} \quad (2)$$

The weights $w_b \in [-1, 1]$ define the strength of each behaviour and are computed based on the perceived context of operation. The noise has a small amplitude and merely ensures that the dynamics escapes unstable fix-points (repellers). Coordination among behaviours is modelled by means of an additional competitive dynamics that controls the weights w_b , which evolve in the following fashion:

$$\tau_b \dot{w}_b = \alpha_b (w_b - w_b^3) - \sum_{b' \neq b} \gamma_{b',b} w_{b'}^2 w_b + \text{noise} \quad (3)$$

The first term constitutes a pitchfork bifurcation, i.e. the dynamics possesses stable fix-points at

$$w_b = \begin{cases} \pm 1 & \text{if } \alpha_b > 0 \\ 0 & \text{if } \alpha_b < 0 \end{cases} \quad (4)$$

The factors $\alpha_b \in [-1, 1]$ are called *competitive advantages*. They determine the degree to which a behaviour is appropriate and desirable in the present context. The second term in equation 3 captures the competitive dynamics in that an active behaviour b' of higher priority suppresses the activation of another conflicting behaviour b . Hence, the factors $\gamma_{b',b} \in [0, 1]$ are called *competitive interactions*. For $|w_{b'}| \sim 1$ and $\gamma_{b',b} > \alpha_b$, the point $w_b = 0$ becomes the new stable fix-point of behaviour b , despite a positive competitive advantage $\alpha_b > 0$. A detailed analysis of how the stability of fix-points varies across different values of competitive advantages and interactions is given in [9]. The time constant τ_b determines the rate at which the behaviours are switched on and off. Similar to the behavioural dynamics, the noise term helps the system to escape unstable fix-points in terms of behaviour coordination.

4.2 System Design

We chose the robot heading ϕ as the behavioural variable of the dynamical system as it offers the advantage that the behaviours can be naturally expressed in this variable. Furthermore, the commanded turn rate $\dot{\phi}$ can be directly applied as a control action to the robot. The translational velocity is regulated by an external control loop, which reduces the robot speed based on two values: 1) the proximity of nearby obstacles, for safety reasons, 2) a high turn rate $\dot{\phi}$, to assure that the robots heading remains close to an attractor state at all times (see Section 4.1).

As mentioned in Section 3, a total of five behaviours have been designed: CORRIDOR FOLLOWING, WALL AVOIDANCE, DOOR PASSING, OBSTACLE AVOIDANCE, and GO TO. The design of these behaviours and their functional form $f_b(\phi)$ (equation 1) are motivated and discussed in our previous work [2,1]. These behaviours or combinations of them are able to deal successfully with the environment on the small scale. In [2], for example, we showed analytically that the system can reliably distinguish between those passages that are too narrow to pass and gaps that are wide enough to traverse safely.

The overall dynamics of the system is obtained from the weighted summation of individual behaviours based on equation 2:

$$\dot{\phi} = \sum_b |w_b| f_b(\phi) + \text{noise} \quad (5)$$

with $b \in \{goto, obst, corr, wall, door\}$. For the coordination of the behaviours the competitive advantages α_b , the competitive interactions $\gamma_{b',b}$, and the time constants τ_b in equation 3 have to be chosen appropriately.

The competitive advantages reflect the relevance and applicability of a behaviour in a particular context. Obviously, GO TO should be activated whenever the agent finds itself in a room and is supposed to approach a goal; otherwise, it is turned off. If the robot is in a certain room, is determined using the estimate of the robot's position and the topological map (Section 2). For $\alpha_{goto} \in (0, 1]$ the behaviour GO TO is switched on. To have the possibility for any competitive interaction $\gamma_{b, goto} \in [0, 1]$ to be greater or smaller than α_{goto} , a value of 0.5 is chosen for the competitive advantage. Hence:

$$\alpha_{goto} = \begin{cases} 0.5 & \text{if in a room} \\ -0.5 & \text{otherwise} \end{cases} \quad (6)$$

Equivalently, CORRIDOR FOLLOWING and WALL AVOIDANCE are relevant if the robot is in a corridor.

$$\alpha_{corr} = \alpha_{wall} = \begin{cases} 0.5 & \text{if in corridor} \\ -0.5 & \text{otherwise} \end{cases} \quad (7)$$

The competitive advantage of DOOR PASSING is tightly coupled to the perception. It is set to a positive value as soon as the door we want to pass is detected (section 3).

$$\alpha_{door} = \begin{cases} 0.5 & \text{if door detected} \\ -0.5 & \text{otherwise} \end{cases} \quad (8)$$

The relevance of OBSTACLE AVOIDANCE depends on the number and proximity of the obstacles currently surrounding the robot. As a measure reflecting this

circumstances, we define an obstacle density by

$$\rho = \sum_i e^{-d_i} \quad (9)$$

The sum goes over all obstacles i and the d_i stand for the distance to them as a multiple of the robot's radius. The competitive advantage of OBSTACLE AVOIDANCE is, then, computed according to

$$\alpha_{obst} = \tanh(\rho - \rho_0) \quad (10)$$

The constant ρ_0 determines the density above which obstacle avoidance becomes relevant (i.e. $\alpha_{obst} > 0$). The tangent hyperbolic ensures that the magnitude of α_{obst} is limited to the interval $[-1, 1]$.

The competitive interaction $\gamma_{b',b}$ reflects the degree to which an active behaviour b' suppresses another behaviour b . In fact, there are situations where behaviours would interfere with each other in an undesirable, counterproductive manner. A door that is half-blocked by an obstacle might still be detected as a door, although the gap to pass is actually too narrow. Hence, we want OBSTACLE AVOIDANCE to suppress DOOR PASSING in the presence of a high obstacle density. Furthermore, if two obstacles lie close to each other, the dynamics of ϕ generates a weak repeller in the middle of them (this has been shown in [2]). This repeller, however, could be dominated by an attractor of another behaviour, which would inevitably lead to collision. Consequently, OBSTACLE AVOIDANCE ought to suppress GO TO and CORRIDOR FOLLOWING as well, if the obstacle density (equation 9) exceeds a critical threshold ρ_c . This prioritization is achieved by appropriately choosing the competitive interactions:

$$\gamma_{obst, goto} = \gamma_{obst, corr} = \gamma_{obst, door} = \frac{1}{2}(1 + \tanh(\rho - \rho_c)) \quad (11)$$

The constant ρ_c determines the density at which obstacle avoidance suppresses the other behaviours ($\gamma_{obst,b} > 0.5$). The functional form of the term is chosen such that $\gamma_{obst,b} \in [0, 1]$. Since there exist no potential conflicts among any other pair of behaviours, all other competitive interactions $\gamma_{b',b}$ are set to zero.

The time constants τ_b determine the time scale at which the behaviours are switched on and off respectively. τ_{obst} is chosen very small, such that the robot reacts almost immediately if a new obstacle is perceived. The same holds for τ_{wall} . As soon as a door is detected, the robot should turn towards it before driving out of detection range again. Consequently, τ_{door} is also chosen to be small. The dynamics of w_{goto} and w_{corr} evolve at a slower rate $\tau_{goto} = \tau_{corr} \gg \tau_{obst}$. Once OBSTACLE AVOIDANCE becomes less relevant, e.g. after the robot circumnavigates an obstacle, the other behaviours switch on gradually rather than causing jitter among themselves and OBSTACLE AVOIDANCE.

5 Results

In the following we discuss a path followed by the robot. The result is quite representative for results obtained from other experiments. Figure 3 shows the trajectory of the robot during a typical task: driving from the charging station in the living room to a goal point in the manipulator lab. During this task, the robot covered a distance of about 50 meters. In the middle of the corridor the way was blocked by people. Therefore, the robot was circling for about a minute before the passage was cleared and it was able to proceed (see [2] for details on behaviour switching in blocked corridors). Through this circling the error in the position estimate, obtained from odometry only, would be too large. However, by correcting this estimate on the basis of the detected corridor (Section 3) the error was kept small and the robot continued successfully to the goal point.

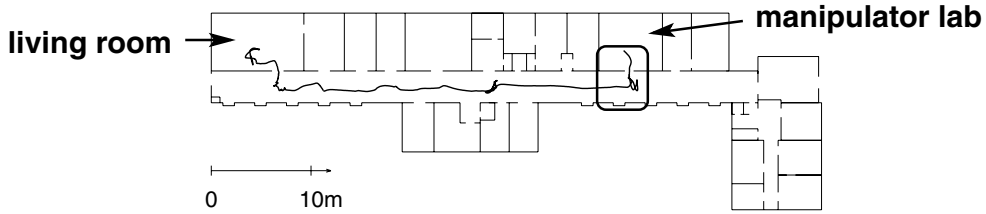


Fig. 3. The trajectory from a trial in our institute. The robot started at the charging station in the living room and proceeded through the corridor to a goal point in the manipulator lab. The bold rectangle can be seen enlarged in Figure 4.

Figure 4 visualizes more details on the trajectory from following a corridor to passing a door and reaching a goal point. In addition, the absolute values of the different weights $|w_b|$ (equation 5) are plotted over time. The labelled ticks on the time axis correspond to the situations A-F that are described below. Notice, that the time difference between two successive ticks is not proportional to the path length between the corresponding events, as the robot does not move at a constant speed.

A) The vicinity of the next node in the topological map was reached and the door was detected. DOOR PASSING was gradually switched on and guided the robot towards the door. CORRIDOR FOLLOWING was turned off on a slower time scale than WALL AVOIDANCE. **B)** The door was blocked by a person leaving the room. However, OBSTACLE AVOIDANCE competed with DOOR PASSING and finally deactivated the latter due to a high obstacle density (equation 11). Consequently, the robot turned away from the door. **C)** The door was detected again and obstacle density has decreased. Therefore DOOR PASSING was gradually activated and the robot turned towards the door again. **D)** The robot passed the door. Due to the high obstacle density DOOR PASSING was switched off again and OBSTACLE AVOIDANCE guided the robot through the opening. **E)** The vicinity of the next node was reached. Thus, GO TO was

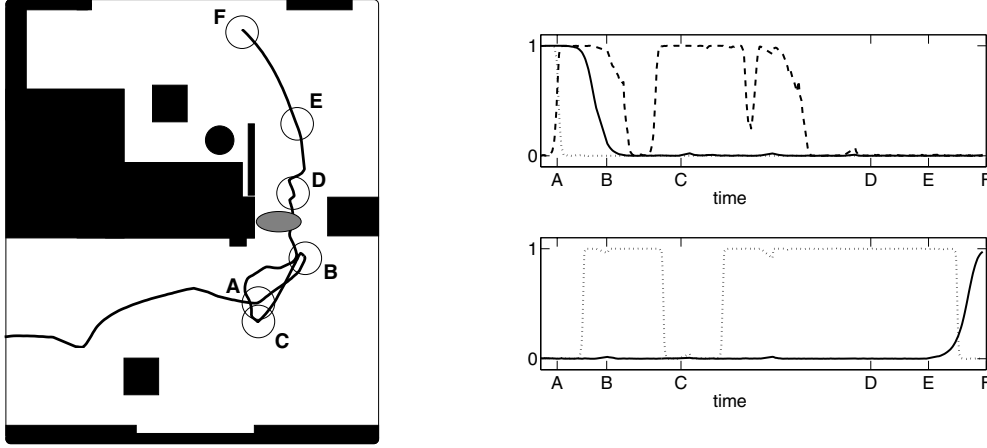


Fig. 4. Details on the task switching from following a corridor to reaching a goal point in a room. The left image shows an outline of the trajectory of the robot. The gray ellipse denotes a person that was leaving the room, when the robot was at location B. On the right hand side, the absolute values of the weights (see equation 5) are plotted over time: $|w_{corr}|$ (upper plot, solid curve), $|w_{wall}|$ (upper plot, dotted curve), $|w_{door}|$ (upper plot, dashed curve), $|w_{obst}|$ (lower plot, dotted curve), and $|w_{goto}|$ (lower plot, solid curve). The situations A-F in the left image correspond to the time instances labeled in the plots. These situations are explained in more detail in the text.

gradually turned on leading to a smooth switch in tasks again. **F)** Eventually, the goal point was reached and the trial was completed.

Note that in Figure 4 not the actual trajectory of the robot is plotted, but the position estimate obtained from the odometry. The sharp bend right after passing the door (D) is therefore not a turn that the robot actually made. It is simply an artifact of updating the position estimate after having passed a door (see Section 3). Further, if the door was closed or the doorway had not been cleared after a certain amount of time, the planning module would detect this. It would search for an alternative path through the topological map. If there is no other way to reach the goal, the robot repeats steps C and D until it eventually will be able to pass.

In section 4.2 the parameter values of the behaviour coordination scheme are motivated and explained. However, an important question is the sensitivity of the system to different parameter values. Our experience shows that the precise values of the competitive advantages, α_b , (equations 6, 7, 8, 10) are not that important as long as the signs are correct. However, the performance of the system is quite sensitive to the competitive interactions, $\gamma_{obst,b}$, (equation 11). This means that the critical obstacle density, ρ_c , has to be chosen carefully, such that OBSTACLE AVOIDANCE suppresses the other behaviours at the right moment.

6 Discussion

Feature detectors were used to extract environmental representations from the sensory data obtained by the sonars of the Scout robot; namely corridors, doors, walls, and obstacles. This geometric information facilitates the single behaviours to perform individual tasks: navigating safely along corridors, through doors and around obstacles. To determine which one of these tasks is appropriate in a given context, a simple topological map and odometric data from the robot were used. To combine the continuous control of the different behaviours with the discrete information from the qualitative map we deployed the dynamical systems approach to behaviour based robotics. In this way smooth switching between the navigational tasks could be achieved. In essence, this comprises a unified control scheme that relaxes the need for accurate metric models of large-scale environments, through embedded place recognition in the behaviours and behavioural competition.

This framework has been successfully tested in the premises of our institute (70×20 meters). An example of an experiment has been presented, which shows that the system is capable of conducting long trials through a large-scale real-world office environment. It can also be seen that the competition among the behaviours is able to deal with more complex situations like half-blocked doors.

There are other approaches to navigate in large-scale environments. Many of them (Xavier [7], for example) need more detailed models of the environment and sophisticated algorithms (e.g. Markov decision process models) to determine an appropriate control action for the robot at all times. An other approach [14] uses, as we do, the superposition of different local behaviours (motor schemas). However, discrete context changes lead to discrete changes in control. The same holds for a variety of systems using topological maps (Dervish [12], for example). In our system this information from the topological map can be merged with the continuous nature of the individual behaviours using the dynamical system approach. This leads to a unified control scheme.

At first sight, it seems that the coordination framework would not scale well to a large number of behaviours, since there is a parameter, $\gamma_{b',b}$, to be determined for each pair of behaviours. However, in our experience, this should not become a problem. As in our system, there are usually few behaviours conflicting with each other. This means that most competitive interactions are set to zero.

The use of sonars as the only sensors restricts our system in different ways. The representations of the environment are rather simple, which can lead to problems (e.g. if two doors are right next to each other). Future research in this project will be directed towards integration of more accurate sensors

(e.g. laser) to get a more reliable representation of the environment. With these more sophisticated sensing capabilities, we will also try to solve the problem of global localization (neglected in this paper) using just a simple topological map. In addition, advanced sensing capabilities would also provide a basis for learning the topological map by the system itself. This could be achieved using more sophisticated modules for place recognition as shown in [8]. Further, the generality of the behaviour coordination scheme has to be exploited. Up to now, it has only been used for navigation behaviours in fetch-and-carry type tasks. It is planned to apply the framework to a different platform [6], where the emphasis lies on human-robot interaction.

Acknowledgment

This research has been sponsored by the Swedish Foundation for Strategic Research. The support is gratefully acknowledged.

References

- [1] P. Althaus and H. I. Christensen. Behaviour coordination for navigation in office environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2298–2304, 2002.
- [2] P. Althaus, H. I. Christensen, and F. Hoffmann. Using the dynamical system approach to navigate in realistic real-world environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1023–1029, 2001.
- [3] R. C. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.
- [4] J. Forsberg, U. Larsson, and Å. Wernersson. Mobile robot navigation using the range-weighted hough transform. *IEEE Robotics & Automation Magazine*, 2(1):18–26, 1995.
- [5] I. Horswill. The Polly system. In D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots: Case studies of successful robot systems*, chapter 5, pages 125–139. MIT Press, Cambridge, MA, 1998.
- [6] T. Kanda, H. Ishiguro, M. Imai, T. Ono, and K. Mase. A constructive approach for developing interactive humanoid robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1265–1270, 2002.

- [7] S. Koenig and R. G. Simmons. Xavier: A robot navigation architecture based on partially observable markov decision process models. In D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots: Case studies of successful robot systems*, chapter 4, pages 91–122. MIT Press, Cambridge, MA, 1998.
- [8] B. Kuipers, R. Froom, W.-Y. Lee, and D. Pierce. The semantic hierarchy in robot learning. In J. Connell and S. Mahadevan, editors, *Robot Learning*, pages 141–170. Kluwer Academic Publishers, Boston, MA, 1993.
- [9] E. W. Large, H. I. Christensen, and R. Bajcsy. Scaling the dynamic approach to path planning and control: Competition among behavioral constraints. *The International Journal of Robotics Research*, 18(1):37–58, 1999.
- [10] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.
- [11] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988.
- [12] I. Nourbakhsh. Dervish: An office-navigating robot. In D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots: Case studies of successful robot systems*, chapter 3, pages 73–90. MIT Press, Cambridge, MA, 1998.
- [13] L. Perko. *Differential Equations and Dynamical Systems*. Springer, New York, 1991.
- [14] A. Saffiotti, K. Konolige, and E. Ruspini. A multi-valued logic approach to integrating planning and control. *Artificial Intelligence*, 76(1–2):481–526, 1995.
- [15] G. Schöner and M. Dose. A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems*, 10:253–267, 1992.
- [16] G. Schöner, M. Dose, and C. Engels. Dynamics of behavior: theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16(2–4):213–245, 1995.

Authors



Philipp Althaus is a PhD student at the Centre for Autonomous Systems at the Royal Institute of Technology (KTH) in Stockholm, Sweden. He graduated 1998 in physics from the Federal Institute of Technology (ETH) in Zurich, Switzerland. Until 1999, he worked as a research assistant at the Institute of Neuroinformatics at the University of Zurich. His research interests include behaviour based robotics, navigation, and mobile robots in general.



Henrik I. Christensen is Professor of Computer Science at the Royal Institute of Technology and director of the Centre for Autonomous Systems. He was awarded M.Sc. and Ph.D. degrees in 1987 and 1989, respectively, from Aalborg University, Denmark. His research is primarily in dynamic vision, robot navigation and systems integration. He has published/edited 7 books, 6 special issues and 140 papers. Dr. Christensen is the coordinator of the European Robotics Network, EURON, and a research coordinator for the EU network on Cognitive Vision. In addition he is on the editorial board of the journals IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal on Robotics Research, International Journal of Pattern Recognition and Artificial Intelligence, Machine Vision and Applications, and Robotics and Autonomous Systems. He maintains active collaborations with research groups on three continents.