Behaviour Coordination for Navigation in Office Environments

Philipp Althaus and Henrik I. Christensen

Centre for Autonomous Systems, Numerical Analysis and Computer Science
Royal Institute of Technology (KTH), S-10044 Stockholm, Sweden
[philipp, hic]@nada.kth.se

Abstract

Behaviour coordination is a notorious problem in mobile robotics. Behaviours are either in competition or collaborating to achieve the goals of a system, which leads to requirements for arbitration and/or fusion of control signals. In most systems the arbitration is specified in terms of "events" that denote positions or sensory input. The detection of these events allows discrete switching between groups of behaviours. In contrast, the fusion of behaviours is often achieved using potential fields, fuzzy rules, or superposition. In most cases, the underlying theoretical foundation is rather weak and the behaviour switching results in discrete changes in the overall system dynamics. In this paper, we present a scheme for behaviour coordination that is grounded in the dynamical systems approach. The methodology provides a solid theoretical basis for analysis and design of a behaviour coordination framework. This framework is demonstrated in the context of a domestic robot for fetch-and-carry type tasks. It is here shown that behaviour coordination can be analyzed as an integral part of the design to facilitate smooth transition and fusion between behaviours.

1 Introduction

Most recent mobile robotics systems exploit a hybrid deliberative system architecture (see [2] for an overview). The deliberative part of such a system is responsible for generating a list of tasks to be accomplished in order to achieve goals. The actual execution of tasks is monitored by a supervisor, while the tasks themselves are implemented as a composition of behaviours that provide control output on the basis of sensory input. The task switching and behaviour coordination involves, typically, a combination of arbitration and fusion across behaviours. There are many different approaches to the coordination as outlined in [2]. The arbitration schemes are often based on the use of discrete logic that can be described as discrete event systems [5]. Popular arbitration schemes include the subsumption system by Brooks [3], and the task language used in the TCA system [15]. In contrast to behaviour arbitration, which typically signifies a task switch, behaviour fusion is used for integration of output from multiple behaviours into a single control signal for the platform. By far the most popular method has been the use of potential fields [6]. In addition, methods such as voting [11] and fuzzy rules [12] have been exploited. A notorious problem in many of these systems is the lack of a solid theoretical foundation for weight selection of different behaviours for integration. Especially for task switches, the existing solutions are rather ad hoc.

An alternative to these methods is the dynamical systems approach introduced by Schöner and Dose [13]. Here a non-linear dynamics approach is adopted to capture both continuous and discrete integration of behaviours into a unified theoretical framework. The approach has so far only been used in relatively simple settings and with a small number of behaviours. This paper outlines how the methodology can be used to implement larger-scale, real-world systems and how adoption of such a framework provides the basis for theoretical design of the behaviour coordination system.

Initially, the dynamical systems approach is introduced (section 2). Another benefit of this approach is that it can be designed to provide robust control solutions in the presence of noise through adaptation of qualitative representations. As such a representation for a navigation task we used a topological map. This map is introduced in section 4. The combination of the dynamical systems approach and qualitative maps allows construction of robot systems which have smooth control in the presence of behaviour coordination/task switching while still encompassing facilities for operation in realistic large-scale environments. The overall system design is presented in section 3, while example results are presented in section 5. Finally, a summary and avenues for future research are outlined in section 6.

2 Dynamical Systems Approach

The conceptual framework of this approach is based on the theory of nonlinear dynamical systems [10]. In the following, we only provide a brief outline of this framework and refer the interested reader to [14] for a more detailed description.

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

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

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 repellor 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 $\vec{f}_b$:

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

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 (repellors). Coordination among behaviours is modelled by means of an additional competitive dynamics that controls the weights w_b, which evolve in the following fashion:

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

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

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

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 [8]. 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.

3 System Design

We have chosen 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 (section 4). 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 ensure that the robot's heading remains close to an attractor state at all times (see section 2). In the remainder of this section, all values denoting distances are expressed as a multiple of the robot radius. This keeps the formulas simpler and the constants dimensionless.

3.1 Design of the Individual Behaviours

To provide the functionality of fetch-and-carry in a domestic setting the following behaviours were designed: GO TO, OBSTACLE AVOIDANCE, CORRIDOR FOLLOWING, WALL AVOIDANCE, and DOOR PASSING.

The behaviour GO TO is expected to align the robot's heading with the direction ψ_{goal} of a goal point in a room (e.g. charging station or a spot in front of a door to be passed). Hence the behavioural dynamics possesses an attractor at ψ_{goal}. To guarantee the continuity of the dynamics over the entire range of heading direction, the function f_{goto} is designed with a periodicity of 2π. The simplest form that meets these criteria is given by (Figure 1):

$$\dot{\phi} = f_{goto}(\phi) = -\lambda_{goto} \sin(\phi - \psi_{goal}) \tag{5}$$

The strength of the attractor is defined by λ_{goto}.

To circumnavigate obstacles, the behaviour OBSTACLE AVOIDANCE has been defined. Furthermore, the combination of CORRIDOR FOLLOWING and WALL AVOIDANCE leads the robot safely along a corridor. The design of these behaviours is motivated and discussed in our previous work [1]. For the sake of completeness their mathematical forms, f_{obst}, f_{corr} and f_{wall}, are stated below.

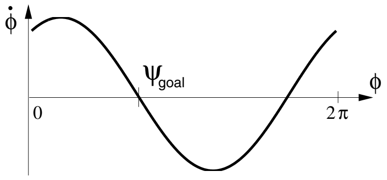

Figure 1: *The dynamics of* GO TO. *At the direction* ψ_{goal}*, in which the goal point lies, an attractor is generated.*

$$f_{obst}(\phi) = \lambda_{obst} \sum_i \left[(\phi - \psi_i) \cdot e^{-c_{obst} d_i} \cdot e^{-\frac{(\phi-\psi_i)^2}{2\sigma_i^2}} \right] \tag{6}$$

The sum goes over the set of all obstacles $\{i\}$, where ψ_i is the direction and d_i the distance to the obstacle.

$$f_{corr}(\phi) = -\lambda_{corr} \sin(\phi - \psi_{corr}) \tag{7}$$

ψ_{corr} denotes the direction along which the corridor has to be traversed.

$$f_{wall}(\phi) = \lambda_{wall} \sum_{l=1}^{2} \left[\sin(\phi - \psi_{w_l}) \cdot e^{-c_{wall} d_{w_l}} \right] \tag{8}$$

The distances and angles to the two corridor walls are denoted by d_{w_l} and ψ_{w_l}. By choosing appropriate values for the gains c_{obst} and c_{wall}, and the angular range σ_i, the robot will only drive through passages between obstacles and walls that are broader than any desired safety width (shown analytically in [1]).

The behaviour DOOR PASSING is supposed to lead the robot through a door. This is in principle the same as moving towards a goal in the direction of the door. Therefore, the same functional form as for GO TO (equation 5) has been chosen:

$$\dot{\phi} = f_{door}(\phi) = -\lambda_{door} \sin(\phi - \psi_{door}) \tag{9}$$

ψ_{door} denotes the direction of the detected door and λ_{door} defines the strength of the attractor.

Since all $f_b(\phi)$ do not explicitly depend on time and the temporal changes of its parameters (e.g. ψ_{goal}) happen on a much slower timescale than the dynamics of ϕ, it is straightforward to show stability for each individual behaviour.

3.2 Design of the Behaviour Coordination

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

$$\dot{\phi} = \sum_b |w_b| f_b(\phi) + \text{noise} \tag{10}$$

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. 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:

$$\alpha_{goto} = \begin{cases} 0.5 & \text{if in a room} \\ -0.5 & \text{otherwise} \end{cases} \tag{11}$$

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

$$\alpha_{corr} = \alpha_{wall} = \begin{cases} 0.5 & \text{if in corridor} \\ -0.5 & \text{otherwise} \end{cases} \tag{12}$$

The competitive advantage of DOOR PASSING is set to a positive value as soon as the door we want to pass is detected (section 4.2).

$$\alpha_{door} = \begin{cases} 0.5 & \text{if door detected} \\ -0.5 & \text{otherwise} \end{cases} \tag{13}$$

The relevance of OBSTACLE AVOIDANCE depends on the number and proximity of the obstacles currently surrounding the robot. The competitive advantage of OBSTACLE AVOIDANCE is related to the obstacle density, $\rho = \sum_i e^{-d_i}$, and is computed according to

$$\alpha_{obst} = \tanh(\rho - \rho_0) \tag{14}$$

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 repellor in the middle of them (shown in [1]). This repellor, 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 exceeds a critical threshold ρ_c. This prioritization is achieved by appropriately choosing the competitive interactions:

$$\gamma_{obst,goto} = \gamma_{obst,corr} = \gamma_{obst,door} = \frac{1}{2}(1+\tanh(\rho - \rho_c)) \tag{15}$$

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.

4 Implementation of the System

To verify the design outlined above, a system has been designed around a 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. Other robots in the laboratory have more comprehensive sensing capabilities, but for the application at hand sonars are adequate to demonstrate the basic principles. The robot possesses a two-wheel differential drive located at the geometric centre which allows omni-directional steering at zero turning radius. For basic navigation in an indoor environment, our institute (70 × 20 metres) in this case, a topological map is used.

4.1 The Topological Map and its Use

The topological map allows both basic task decomposition for selection of a route and provides an identification of places used for the behaviour coordination (section 3.2). This map consists of nodes and edges that connect these nodes. 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. Figure 3 shows the topological map of our institute.

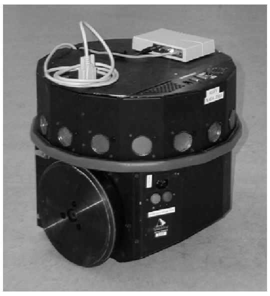

Figure 2: *The Scout robot used in the experiments.*

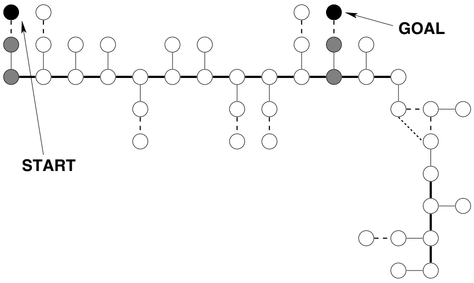

Figure 3: *The topological map of our institute: The circles depict nodes. Edges are of three different types: corridor (thick line), room (dashed line), and door (thin line). Additional nodes for goal points and starting positions can be added arbitrarily. The nodes denoted with "start" and "goal" correspond to the initial (charging station) and final location of the trial described in the results (section 5). The nodes in grey are the ones used to execute this plan.*

It is assumed that the initial position and orientation of the robot is known (e.g. charging station). From there odometry is used to determine the robot's 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. However, on long trials over a great distance the error would grow bigger than desired. To avoid this the odometry values are corrected based on detected features (section 4.2). In a corridor, the robot's orientation and its position relative to the corridor walls are adjusted. Every time a door is passed orientation and position relative to the door posts can be updated correctly.

4.2 Extracting Geometric Representations from Raw Sensor Data

For navigation in an indoor environment using the behaviours described in section 3.1, it is necessary to equip the robot with facilities for wall detection, obstacle extraction, and recognition of doorways.

The walls of a corridor are extracted from the sonar readings using a Hough transform [4], without making any assumptions on the direction or width of the corridor. For obstacles, a very simple representation is used, only considering the distance to obstacles but ignoring their size and shape. For more details on how corridors and obstacles are detected see [1].

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 the wall and not more than 50 cm behind the wall are kept in a FIFO buffer. The largest angular segment (from the robot's 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 ψ_{door} (equation 9) 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 still be detected as doors. However, situations like this are resolved by the coordination between a door passing and an obstacle avoidance behaviour (see the design in section 3.2). 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 detected. In order to do this, a Hough transform is invoked on the 100 most recent sonar echos.

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 relative 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.

To determine which detectors should be invoked, the topological map is used. The information about the exact location of its nodes and the odometry values determine if the robot finds itself in a corridor or in a room and/or close to a door. To detect a goal point, no sensors are used yet. Its location is defined by a node in the topological map. In combination with odometry this information provides an estimate of $\phi - \psi_{goal}$ (equation 5), the direction of the goal relative to the robot's orientation.

5 Results

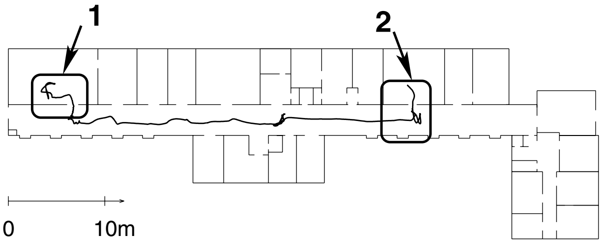

Figure 4: *The trajectory of the robot in a typical task driving through our institute (from left to right). The rectangle denoted by 1 is shown enlarged in Figure 5; the one denoted by 2 in Figure 7.*

Figure 4 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. The rectangles denoted by 1 and 2 are shown enlarged in Figures 5 and 7. In these figures different situations are denoted by the symbols A-O, which are described in the text below. Figure 6 and Figure 8 depict the evolution of the weights of the behaviours. The labelled tics on the time axis refer to the corresponding locations of the robot. For a detailed description of the part of the trajectory in the corridor we refer to our earlier work [1]. During this trial the robot covered a distance of about 50 metres. The corresponding track through the topological map can be seen in Figure 3.

A) The robot at its starting position: Immediately after driving off, OBSTACLE AVOIDANCE was switched on. It stayed on at all times, while moving around in the room, since the obstacle density was always above ρ_0 (equation 14). **B)** GO TO, which evolves on a slower time scale than OBSTACLE AVOIDANCE was gradually switched on: The robot started turning towards the position of the node in front of the door. **C)** The way towards the door was blocked: The obstacle density exceeded the critical value ρ_c and GO TO was turned off (equation 15). The robot turned around to avoid the obstacles. **D)** GO TO was turned on again: The obstacle density has dropped, and $|w_{goto}|$ increased on a slow time scale. The robot's heading was directed towards the location of the node in front of the door. **E)** OBSTACLE AVOIDANCE controlled the robot: The gap was big enough for the robot to pass, hence it stayed in the middle, between the two obstacles (see [1] for details). GO TO was off, due to a high obstacle density. **F)** The vicinity of the next node was reached: The direction of the door was extracted from the sonar data (see section 4.2). DOOR PASSING was turned on almost immediately and the robot turned towards the

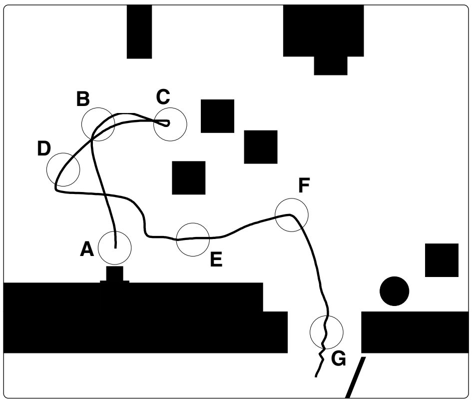

Figure 5: *The trajectory of the robot starting at the charging station (A) and leaving the room towards the corridor (G). The black obstacles denote chairs, two shelves, a table and a waste bin. The situations labelled by the symbols A-G are explained in the text. The circles at these points depict the size of the robot.*

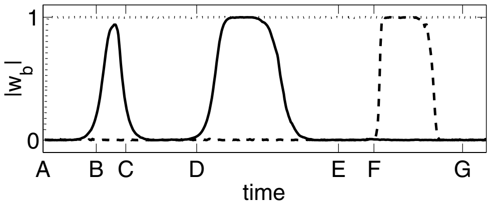

Figure 6: *Time plot of the absolute values of the weights: $|w_{obst}|$ (dotted curve), $|w_{goto}|$ (solid curve), and $|w_{door}|$ (dashed curve) (see equation 10). The time instances labeled by the symbols A-G correspond to the situations in Figure 5.*

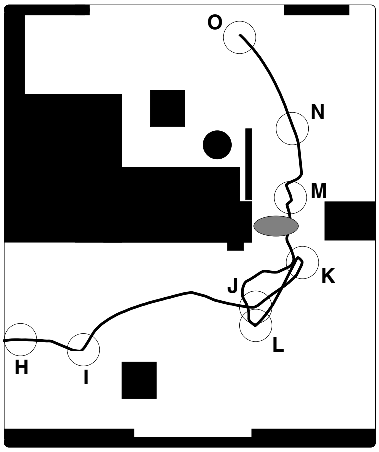

Figure 7: *The trajectory of the robot from the corridor (H) to a goal point in a room (O). The grey ellipse denotes a person that was leaving the room, when the robot was at location K. The situations labelled by the symbols H-O are explained in the text. The circles at these points depict the size of the robot.*

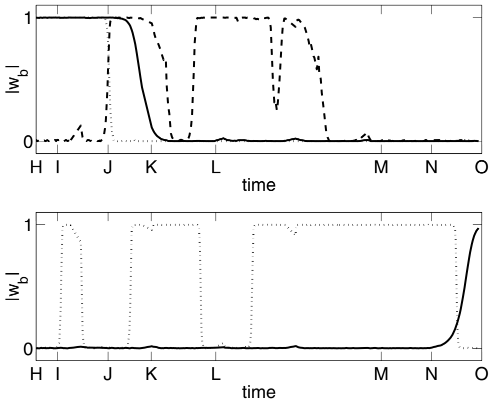

Figure 8: *Time plot of the absolute values of the weights: $|w_{corr}|$ (upper plot, solid curve), $|w_{wall}|$ (upper plot, dotted curve), $|w_{door}|$ (upper plot, dashed curve), $|w_{obst}|$ (lower plot, dotted curve) and $|w_{goto}|$ (lower plot, solid curve) (see equation 10). The time instances labeled by the symbols H-O correspond to the situations in Figure 7.*

door. **G)** The robot passed the door: Due to a high obstacle density, DOOR PASSING was actually turned off. Nevertheless, OBSTACLE AVOIDANCE guided the robot out of the room. After following the corridor, the robot reached point **H)** The robot was still in the corridor: CORRIDOR FOLLOWING and WALL AVOIDANCE were switched on; the other behaviours were switched off. **I)** An obstacle appeared: OBSTACLE AVOIDANCE was turned on for a short time. **J)** The vicinity of the next node was reached and the door detected: DOOR PASSING was switched on and guided the robot towards the door. CORRIDOR FOLLOWING was turned off on a slower time scale than WALL AVOIDANCE. **K)** The door was blocked by a person leaving the room: The robot still detected the small opening and considered it to be a door. However, the obstacle density was above ρ_c and DOOR PASSING was switched off (equation 15). The robot turned away from the door. **L)** The door was detected again: The person had left the door passage, and DOOR PASSING was switched on again. **M)** 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 door. **N)** The vicinity of the next node was reached: GO TO was gradually turned on and the robot was heading for the goal point. **O)** The goal point was reached: The robot arrived at the node of the goal point and the task was completed.

6 Discussion

We presented a control scheme which successfully navigates a mobile robot through a cluttered large-scale real-world office environment. The dynamical system approach provided a suitable means for the design of robotic behaviours and their coordination. The behaviours rely on an approximate, simple geometric representation of the environment that directly anchors on the information provided from low level sensors. The activation dynamics to coordinate the behaviours also makes use of these representations and a simple topological map combined with odometry. The continuous nature of signals controlling the robot (behaviours) and the discrete nature of task switching (coordination) have been expressed in a unified framework. This framework comprises a mathematically sound basis, where behaviours are gradually turned on and off on different time scales.

There are other successful indoor navigation systems using a topological map described in the literature: for example Xavier [7] and Dervish [9], to name just two. These approaches use assumptions on probabilities of detecting features and progressing to the next node (state). In our approach the robot only roughly knows its position, and also the detection of features is not entirely reliable. However, the dynamic coordination scheme allows the robot to navigate safely and cope with unforeseen or complex situations, such as blocked passages and partially closed or miss-detected doors, in a flexible manner.

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 obtain a more reliable representation of the environment. Also, the problem of global localization (neglected in this paper) using just a simple topological map, can only be solved with more sophisticated sensing capabilities.

Once the tasks are more complex (e.g. longer missions with multiple goals) or the environment poses unexpected constraints (e.g. closed doors or permanently blocked corridors), observation of the current plan execution status becomes a necessity. We are considering integrating this into a coherent framework that also allows the robot to explore alternative strategies to achieve a particular task.

Acknowledgments

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

References

[1] 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.

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

[3] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[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] Y. C. Ho. *Introduction to Discrete Event Systems.* IEEE Press, 1991.

[6] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[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] 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.

[9] 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.

[10] L. Perko. *Differential Equations and Dynamical Systems.* Springer, New York, 1991.

[11] P. Pirjanian, H. I. Christensen, and J. A. Fayman. Application of voting to fusion of purposive modules: An experimental investigation. *Robotics and Autonomous Systems*, 23(4):253–266, 1998.

[12] A. Saffiotti, K. Konolige, and E. Ruspini. A multivalued logic approach to integrating planning and control. *Artificial Intelligence*, 76(1–2):481–526, 1995.

[13] 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.

[14] 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.

[15] R. G. Simmons. Structured control for autonomous robots. *IEEE Transactions on Robotics and Automation*, 10(1):34–43, 1994.