Kungl
Tekniska
Högskolan

Indoor Navigation for Mobile Robots: Control and Representations

Philipp Althaus

Stockholm 2003

Doctoral Dissertation
Royal Institute of Technology
Numerical Analysis and Computer Science
Computational Vision and Active Perception Laboratory

Akademisk avhandling som med tillstånd av Kungl Tekniska Högskolan framlägges till offentlig granskning för avläggande av teknisk doktorsexamen fredagen den 21 november 2003 kl 13.00 i Kollegiesalen, Administrationsbyggnaden, Kungl Tekniska Högskolan, Valhallavägen 79, Stockholm.

ISBN 91-7283-605-9

TRITA-NA-0322
ISSN 0348-2952
ISRN KTH/NA/R-03/22
CVAP 281

Universitetsservice US-AB, Stockholm 2003

Abstract

This thesis deals with various aspects of indoor navigation for mobile robots. For a system that moves around in a household or office environment, two major problems must be tackled. First, an appropriate control scheme has to be designed in order to navigate the platform. Second, the form of representations of the environment must be chosen.

Behaviour based approaches have become the dominant methodologies for designing control schemes for robot navigation. One of them is the dynamical systems approach, which is based on the mathematical theory of nonlinear dynamics. It provides a sound theoretical framework for both behaviour design and behaviour coordination. In the work presented in this thesis, the approach has been used for the first time to construct a navigation system for realistic tasks in large-scale real-world environments. In particular, the coordination scheme was exploited in order to combine continuous sensory signals and discrete events for decision making processes. In addition, this coordination framework assures a continuous control signal at all times and permits the robot to deal with unexpected events.

In order to act in the real world, the control system makes use of representations of the environment. On the one hand, local geometrical representations parameterise the behaviours. On the other hand, context information and a predefined world model enable the coordination scheme to switch between subtasks. These representations constitute symbols, on the basis of which the system makes decisions. These symbols must be anchored in the real world, requiring the capability of relating to sensory data. A general framework for these anchoring processes in hybrid deliberative architectures is proposed. A distinction of anchoring on two different levels of abstraction reduces the complexity of the problem significantly.

A topological map was chosen as a world model. Through the advanced behaviour coordination system and a proper choice of representations, the complexity of this map can be kept at a minimum. This allows the development of simple algorithms for automatic map acquisition. When the robot is guided through the environment, it creates such a map of the area online. The resulting map is precise enough for subsequent use in navigation.

In addition, initial studies on navigation in human-robot interaction tasks are presented. These kinds of tasks pose different constraints on a robotic system than, for example, delivery missions. It is shown that the methods developed in this thesis can easily be applied to interactive navigation. Results show a personal robot maintaining formations with a group of persons during social interaction.

Keywords: mobile robots, robot navigation, indoor navigation, behaviour based robotics, hybrid deliberative systems, dynamical systems approach, topological maps, symbol anchoring, autonomous mapping, human-robot interaction

ISBN 91-7283-605-9 • TRITA-NA-0322 • ISSN 0348-2952 • ISRN KTH/NA/R-03/22 • CVAP 281

Acknowledgments

Many people are responsible for the quality of this thesis. Over the last four years, they contributed either directly to my scientific work or to my quality of life in general. There are too many persons involved to name them all. Here, I would like to express my sincere thanks to the most important ones:

Henrik for directing my research throughout my studies. Thank you for your competent scientific guidance as my supervisor, and for providing the great research environment as the director of the Centre for Autonomous Systems.

Guido, Frank, and **Lars** for all the enjoyable lunch breaks filled with discussions about soccer and other important aspects of life.

Patric for all your valuable help with virtually any kind of hardware or software problem. Just knowing that you are around improved the advancement of my work.

All members of **CAS** and **CVAP** for contributing to the pleasant atmosphere every single day at work.

Hiroshi Ishiguro for guiding my research during the four months in Japan.

All members of **IRC** at ATR in Japan for providing a great research environment.

Göran, Patrick, Juan, Peter, and **Örjan** for being my friends and making me feel home in Stockholm almost immediately.

The whole **tidsfördrif gang** for all the merry Tuesday evenings.

Den **Vollidioten**, den **Badenerstrassgspänli**, den alten **Handballkollegen**, und meinen Geschwistern **Annette** und **Christian** für die vielen emails aus der Heimat, den tollen Besuchen in Stockholm und all den vergnüglichen Stunden während meiner Schweizaufenthalte.

Meinem lieben **Mami** für all die Unterstützung während meiner Doktorarbeit; in moralischer und finanzieller Hinsicht.

Min älskling **Jeanette** för allt stöd och all kärlek. Du är bäst!

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

Contents

List of Figures

Chapter 1

Introduction

Today, various robotic systems are used in industry. They are applied to tasks like packing, welding, and painting. These static manipulators operate in predefined safe workspaces with emphasis on speed and precision. Very different types of systems are *autonomous robots* built on mobile platforms, where a new degree of control flexibility is needed. As opposed to industrial robots, they move around in their environment, which is often highly unstructured and unpredictable. Slowly, various markets are emerging for this type of robotic systems. Entertainment applications and different types of household or office assistances are the primary targets in this area of development. The Aibo dog from Sony and the Trilobite vacuum cleaner from Electrolux are early examples of this future industry.

All these existing and potential applications of autonomous systems have one problem in common: *navigation*. The robots must move about in their surroundings in a flexible and robust manner. Since environments in real-world applications are highly unpredictable, the system gathers sensory information from which it must extract representations of its surroundings. These representations are used by the control system in order to fulfill the navigational tasks. The main focus of this thesis is precisely these two topics, *control* of the platform and *anchoring* sensory information to objects in the real world. It must be emphasised that these two problems are in no way independent of each other. The design of a controller is heavily dependent on the nature of the available knowledge of the robot's immediate surroundings as well as the large-scale structure of the environment. The richer and more accurate this knowledge is, the easier it becomes to find a suitable control algorithm. However, acquiring this knowledge, which means extracting representations from sensory data, is a delicate problem in itself. Usually, the more complex these representations are, the more computational power is needed. Hence, the complexity of the anchoring processes must be kept within certain bounds to allow the control system to guide the robot at a reasonable speed.

The problems of controlling the platform and acquiring knowledge of its surroundings can be highly simplified by engineering the environment. For example, the whole setting can be designed to account for the specifics of the robotic platform as in factory spaces. Alternatively, easy identifiable artificial landmarks can be placed to simplify self-localisation of the platform. Buried wires to guide autonomous lawn mowers are a typical example. However, for large markets targeting all households and office spaces additional investments for engineering environments are believed to be too high. Hence, this thesis focuses on *minimalistic models* in terms of environment modelling. The navigation system developed here is applicable to any indoor environment consisting of rooms, corridors, and doorways, which are accessible by a wheeled platform.

The remainder of this chapter is organised as follows. First, the two main topics, control (section 1.1) and representations (section 1.2), are introduced. In section 1.3, the scientific contributions are addressed. Finally, an outline of this thesis is given in section 1.4.

1.1 Control

The first so-called intelligent systems were dominated by approaches from *classical artificial intelligence*. Complete knowledge of the environment and a deterministic outcome of actions was presumed. Then, a symbol based planner calculated an action sequence to be executed in order to accomplish the given goal. The applicability of this methodology to mobile robots was rather limited due to various deficiencies. Complete knowledge of the world is usually not available and the outcome of control actions is subject to noise introduced by imperfect actuators and other outside influences from the surrounding. Hence, the robot must make use of sensors to update its belief of the world. The first systems for autonomous robots were designed based on these sensory data and a reasoning system inspired from classical AI (see (Moravec, 1983) for a review). They carried out extensive planning and replanning due to the noise in sensory data and outcome of motor actions. Thus, enormous computation time was used, which made the robots very slow. Furthermore, environments are often dynamic in an unpredictable way, which means that even in theory it is impossible to capture all of their properties.

To overcome the flaws of classical AI approaches, a new paradigm, *behaviour based robotics*, has established itself over the last decade. Arkin (1998) provides an excellent overview of existing systems. In a behaviour based approach the control of the platform is distributed to several so called behaviours. Each of these behaviours is tightly coupled to sensory data and controls the robot in a reactive way to accomplish some subproblem of the navigational task. Typical examples of such subtasks are “obstacle avoidance”, “wall following”, or “approach goal”. Intelligence arises through the combination of these perception-action loops rather than through symbolic reasoning. Despite their success, these initial purely reactive approaches were rather limited in solving complex tasks, for example to reach a distant goal in a large-scale environment. Hence, elements of the classical approach were incorporated again to enable planning and reasoning on a symbolic level using some predefined model of the world. This led to *hybrid deliberative systems*[1] functioning on two levels of control. On the one hand, a *reactive* part deals with local navigation based on immediate sensor readings. On the other hand, a *deliberative* part takes care of long term mission planning to achieve certain higher level goals.

There are many different methodologies to design these hybrid deliberative architectures, each with its own advantages and drawbacks. Unfortunately, most of them lack a sound theoretical framework to express task constraints in a mathematical way. However, the *dynamical systems approach* by Schöner and Dose (1992) models behaviours and their coordination based on nonlinear dynamical systems. A unified framework incorporates both levels of control of a hybrid deliberative architecture. Through mathematical design, task constraints can easily be expressed in the system. Furthermore, it makes it possible to combine the continuous nature of geometrical properties and control signals with the discrete nature of decision making and task switching. One great advantage of this property is that a continuous control signal, which is applied to the robot's actuators, can be assured at all times.

The dynamical systems approach was the control framework of choice in the studies covered by this thesis. Earlier, it was deployed successfully in simulation projects and on robots in simplified real-world settings. Here, it was applied to a robot, whose primary task was to navigate from any starting position to an arbitrary goal point in a large-scale indoor environment.

1.2 Representations

A control system acting in the real world needs some kind of knowledge about its environment. In robotics, this knowledge is usually referred to as *representations*, which provide parameters to the controller. These representations are mostly geometrical, capturing the properties of the environment. Furthermore, they can also be in the form of labelled entities, which are typically used for task specification. In AI, this knowledge is often referred to as *symbols* allowing decision making and reasoning about the world. In behaviour based systems two different types of representations or symbols are considered. On the one hand, symbols with geometrical properties reflect the immediate surrounding of the robot in order to allow safe navigation fulfilling local tasks. In indoor navigation these types of symbols include walls, doorways, or obstacles such as tables, chairs, and people. On the other hand, a model of the large-scale structure of the area is required to enable planning of routes to fulfill an entire mission. These representations are usually in the form of maps, which have geometrical and topological properties.

Geometric properties of objects are extracted from sensory information to allow safe navigation. Naturally, the type and quality of these properties are heavily

[1] In the literature, they are often also referred to as "hybrid systems". Throughout this thesis the term "hybrid deliberative system" is used to make a distinction to the definition of hybrid systems in control theory.

dependent on the sensors used by the system. The most common ones are range sensors, such as sonars or lasers. Another, more advanced approach is the use of cameras and computer vision techniques to extract information from pictures, but these tend to be computationally expensive. The studies covered in this thesis focus on the use of ultrasonic sensors. However, there is nothing in the methodology that prevents employing more advanced sensing capabilities. Furthermore, odometric information from wheel encoders can be used to combine several measurements taken at different time instances.

Maps are used to reflect the large-scale structure of environments. In general, they are predefined and constitute the system's a priori knowledge about the world. Many different types of maps are used in the literature. This choice is also dependent on the sensing capabilities of the robot, since the system needs modules for place recognition. Only this capability allows self-localisation and subsequent use of the map to achieve the navigational task. The quality of the odometric information available often poses the major constraints on the complexity of the map. A topological map containing low complexity information was applied in the studies of this thesis, since the focus lies on minimalistic models that are computationally inexpensive and can be applied to basically every indoor environment. Apart from using a map for navigation, the problem of acquiring this map autonomously is a major issue in research. Naturally, the methodologies and algorithms to acquire such a map are very much dependent its actual type and complexity.

Both types of representations, geometrical properties of nearby objects and maps of large-scale structures, must be related to sensory data. Establishing this connection and maintaining it over time is called *symbol anchoring*. This problem is usually solved on a system by system basis due to a lack of a general methodology. However, such a general framework would be very useful as a design principle and to enable comparison of different solutions. In this thesis an attempt of a framework for symbol anchoring in hybrid deliberative systems is presented.

1.3 Contributions

This section summarises the scientific contributions of this thesis. Furthermore, publications which cover most of the material are listed. These papers can be downloaded at `http://www.nada.kth.se/~philipp/Publications`.

The main contribution concerns navigation in indoor environments (section 1.3.1). In addition, a map acquisition scheme was developed (section 1.3.2), and studies on navigation in human-robot interaction were performed (section 1.3.3).

1.3.1 Indoor Navigation

This thesis covers many different aspects of indoor navigation for autonomous robots: behaviour design, behaviour coordination, map representations, and symbol anchoring.

The dynamical systems approach is applied to realistic tasks in the real world for the first time. In contrast to previous implementations, the speed of the platform is incorporated in the framework. Then, the behaviour coordination scheme is exploited thoroughly. Continuous sensory signals and discrete information from the topological map are integrated into the same framework. Furthermore, these data are used for decision making, which is of a discrete nature. However, the integration of dynamical systems leads to smooth switching between different navigational tasks which ensures a continuous control signal at all times.

The system uses symbols on both levels of control. On the reactive level, geometrical representations of the environment determine the output of individual behaviours. On the deliberative level, context information enables coordination of these behaviours. A framework is proposed, where the anchoring of these symbols in the real world clearly distinguishes between the two levels of abstraction.

In essence, a unified navigation scheme for large-scale indoor environments is presented. It copes with unforeseen situations in a flexible manner and, through its low complexity processes, uses very limited CPU time.

- Althaus P., Christensen H. I., and Hoffmann F. 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.
- Althaus P. and Christensen H. I. Smooth task switching through behaviour competition. In *Intelligent Autonomous Systems 7*, pages 9-17. IOS Press, Amsterdam, NL, 2002.
- Althaus P. and Christensen H. I. Behaviour coordination for navigation in office environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2298-2304, 2002.
- Althaus P. and Christensen H. I. Smooth task switching through behaviour competition. *Robotics and Autonomous Systems*, 44(3-4):241–249, 2003.
- Althaus P. and Christensen H. I. Behavior coordination in structured environments. *Advanced Robotics*, 17(7):657–674, 2003.
- Althaus P. and Christensen H. I. A framework for anchoring in hybrid deliberative systems. Submitted to *Autonomous Robots*, 2003.

1.3.2 Automatic Map Acquisition

The navigation system was enhanced with a behaviour for person following. This enables the robot to follow a guide through a previously unknown environment. Via a basic interface, it receives input about the existence of corridors and the names of rooms. The system is able to build its own topological map of the area with simple algorithms and very low computational complexity. This map can in turn be used by the robot to navigate autonomously throughout the whole environment.

- Althaus P. and Christensen H. I. Automatic map acquisition for navigation in domestic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1551–1556, 2003.

1.3.3 Human-Robot Interaction

During a 4 months stay at the Intelligent Robotics and Communication Laboratories at the Advanced Telecommunications Research Institute (ATR) in Kyoto (Japan), navigation in the context of human-robot interaction was studied. The framework used for office navigation was extended by adding new behaviours. The topological map was connected to a state diagram reflecting the events in an interaction task. On its way through the offices the robot was able to join a group of people engaged in a conversation, and subsequently resuming its old plan. This project was a first attempt to study navigation specifically for human-robot interaction.

- Althaus P., Ishiguro H., Kanda T., Miyashita T. and Christensen H. I. Navigation for human-robot interaction tasks. Submitted to *IEEE International Conference on Robotics and Automation*, 2004.

1.4 Outline

The main topic of this thesis, indoor navigation, is covered in the next three chapters. The control system is studied in chapter 2, the issues on representations and symbol anchoring in chapter 3, and the results in chapter 4. Chapter 5 covers the map acquisition procedure with its results, while chapter 6 presents the studies on navigation in human-robot interaction tasks. Finally, conclusions are drawn in chapter 7.

Chapter 2: Control System

An overview of different behaviour based approaches is given with emphasis on behaviour coordination. The dynamical systems approach is introduced in detail. Then, the motivation and the design of the navigation behaviours and their coordination is presented.

Chapter 3: Representations

Issues on local geometrical representations and indoor maps are presented. Symbol anchoring in the real world are discussed and the general framework for anchoring in hybrid deliberative architectures is introduced. Then, the choices of symbols in these studies are motivated and all the anchoring processes are discussed in detail.

Chapter 4: Evaluation

First, the actual indoor environment and the robot used in the experiments and its sensing capabilities are introduced. Navigation results in the office environment of our institute are presented. Emphasis is placed on decision making on both levels of control: the reactive (attractor dynamics) and the deliberative (behaviour coordination). In addition, some issues related to symbol anchoring are discussed.

Chapter 5: Autonomous Map Acquisition

Initially, an overview of map learning in indoor environments is given. Then, the scenario of semi-autonomous map acquisition is introduced. The acquisition mechanisms and results are presented. The chapter concludes with a discussion.

Chapter 6: Human-Robot Interaction

The special constraints on navigation in human-robot interaction tasks are discussed. Then, the behaviour design and coordination for a test scenario are presented. Furthermore, the extension of the topological map, the state diagram, is introduced. Finally, some initial results are displayed, followed by a discussion.

Chapter 7: Discussion

Finally, the thesis is summarised with particular focus on the office navigation system as a whole. Moreover, the major open issues and avenues of future research are discussed.

Chapter 2

Control System

This chapter covers all issues related to the control of the indoor navigation system. Behaviour based control is the dominant paradigm in mobile robotics. Its basic principles and some examples are discussed in section 2.1. The dynamical systems approach became the control framework of choice in the work of this thesis and is introduced in section 2.2. In section 2.3 the actual design of the indoor navigation system is presented. Finally, the chapter concludes with a summary and a discussion (section 2.4).

2.1 Behaviour Based Robotics

First, a bit of history and an overview on the behaviour based approach is given (section 2.1.1). Only the basic concepts and the terminology are introduced here. For a broad review on behaviour based systems see (Arkin, 1998). Yet one focus of this thesis is on the question of how to combine different behaviours. This problem and existing solutions are discussed in more detail in section 2.1.2.

2.1.1 Overview

As outlined in the introduction (section 1.1), the classical AI approaches to robotics (Moravec, 1983) were limited by their computational complexity. The actions of the robot were determined through reasoning in a central world model. Hence, these systems scaled poorly to real-world complexity and could hardly react to unforeseen events in real-time. Two new, at that time revolutionary, ideas paved the way for behaviour based robotics: *situatedness* and *modularity.*

Situatedness describes the idea that intelligence has to be studied in terms of an agent interacting with its environment (Brooks, 1990). Which means that the sensors which are deployed to perceive the environment and the motors used to move about play a central role. It was understood that a tight coupling between sensors and motors can lead to surprisingly advanced behaviour without using any form of abstract reasoning. Braitenberg (1984) gives a brilliant insight into this new paradigm.

By studying humans performing various tasks, Arbib et al. (1987) discovered that complex movements can be described by different, rather independent modules. They called these modules motor schemas. Each of these schemas takes care of one subtask of the whole action. By combining these simpler units of motor control, rather complex moving patterns could be explained.

In robotics, these two insights led to the concept of *behaviours*. A behaviour is a module that couples motor commands very tightly to sensory readings. Behaviours have a minimal internal state and solve tasks of limited complexity in a reactive manner. Nevertheless, intelligence arises through the combination of several behaviours and the interaction of the robot with the world. The subsumption architecture of Brooks (1986) was a first example of these new ideas implemented on a real robot wandering around in an office environment. A nice review on this shift in paradigm is given in (Brooks, 1991b). The initial success of such architectures led to a design philosophy which could be called *strict behaviourism*, where any kind of representations and reasoning in the sense of traditional AI were rejected (Brooks, 1991a). However, this purely reactive approach failed to design systems for more complex missions and it was argued that it does not scale to human-like tasks (Tsotsos, 1995). Consequently, the behaviour based systems were enhanced by a reasoning module which takes care of mission planning. Usually, this module is steering the coordination of the different behaviours. As a result, a *hybrid deliberative architecture* as depicted in Figure 2.1 became the most common type of behaviour based systems.

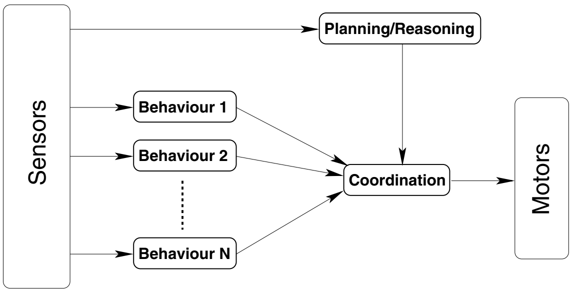

Figure 2.1. The hybrid deliberative architecture. An arbitrary amount of behaviours solve subtasks in a reactive manner. A coordination scheme is used to determine the appropriate motor command. This scheme is influenced by a reasoning system creating plans for complex missions.

The design of the individual behaviours is strongly dependent on the platform used. Its shape, actuators, and sensors influence the nature of these reactive modules. Nevertheless, some types of behaviours have become the standard choice for navigational tasks. Every system moving around in the world needs two basic abilities: approaching some kind of target and avoiding obstacles on its way. Moreover, additional behaviours like, for example, "corridor following", "door passing", "wall following", and "avoid past" are very common. The exact choice of behaviours is highly dependent on the environment and the tasks to be achieved. Furthermore, the behaviour based approach is not restricted to navigation. It is also successfully used for learning and group behaviour (Mataric, 1997), and on humanoid robots (Brooks, 1997).

There were many different strategies developed to combine the individual behaviours. The same holds for the way the planning module is integrated into the architecture. We will investigate this issue in more detail in the following section.

2.1.2 Behaviour Coordination

The behaviour based approach to robotics is a conceptual framework; and as such it does not specify a particular formalism for the design of the system. This great freedom makes it also hard to compare different implementations. As opposed to classical AI approaches, this paradigm is not based on provable facts such as plan correctness, optimality, and so forth. However, the planning module of a hybrid deliberative architecture (see Figure 2.1) can usually be built using these theoretical tools. The meeting point of these two design techniques is at the level of behaviour coordination. Here, it must be decided at what time which set of behaviours should be active. Furthermore, behaviours have their own, possibly incompatible, objectives. Hence, from several behaviour outputs, a coordination scheme has to determine a single control signal which can be applied to the actuators of the robot. In general, existing coordination approaches can be divided into two classes: *behaviour arbitration* and *behaviour fusion*. Different systems of these two classes are introduced below. For a detailed comparison see (Pirjanian, 1998).

Behaviour Arbitration

Arbitration mechanisms select at all times the most appropriate behaviour. This behaviour is, then, taking control of the robot and determining the action to be taken. Arbitration strategies can roughly be divided into *priority based* and *state based*.

In priority based arbitration, behaviours are ordered in a hierarchical way. The subsumption architecture (Brooks, 1986) is an example using this type of arbitration. Each behaviour has its own level of competence. A behaviour can suppress input and inhibit output of modules on lower levels dependent on sensory data or a timing policy. This mechanism of choosing the most appropriate behaviours is preprogrammed. Hence, this is a purely reactive approach without any central representations or reasoning system present. Nevertheless, Mataric (1992) could extend this architecture to a goal-driven robot by integrating representations on the level of individual behaviours.

In state based arbitration, the robot's state in a plan execution determines the appropriate behaviour. This plan execution is typically defined in the framework of discrete event systems (Ho, 1991). Kosecká and Bajcsy (1994), for example, designed such a system for robot navigation. Each state corresponds to one behaviour. At detection of certain events, a state change occurs; thus, activation of a new behaviour. A more abstract formulation of this approach can be found in (Kosecká et al., 1997). Another navigation example using the same concepts was presented by Arkin and MacKenzie (1994). Their terminology is based on a formal definition of motor schemas (Arkin, 1990) and finite state acceptor diagrams. The latter framework is also very common for systems performing human-robot interaction tasks (Kanda, Ishiguro, Imai, Ono and Mase, 2002).

Behaviour Fusion

Fusion mechanisms allow several behaviours, at the same time, to contribute to the action taken by the robot. These behaviours have typically different, and possibly incompatible, objectives. The strategies to determine a single motor command from these multiple objectives can roughly be classified into *voting*, *fuzzy approaches*, and *superposition*.

In voting based fusion (Payton et al., 1990), each behaviour casts votes for the various possible actions. Then, the action receiving the most votes is applied to the actuators of the platform. This approach was successfully used in the navigation system DAMN (Rosenblatt, 1997), and also for other tasks like steering a camera head (Pirjanian et al., 1998).

Approaches using fuzzy logic (Saffiotti et al., 1995) are, in principle, similar to voting based fusion. Instead of voting for different actions, each behaviour defines a membership function over the set of possible actions, dependent on the situation the robot faces. These functions are combined using standard fuzzy reasoning. The resulting membership function is defuzzyfied to compute a single motor action. Yen and Pfluger (1995) made the similarity to voting based fusion explicit by formulating a fuzzy version of the DAMN architecture.

The most popular approach to superposition based fusion is the potential field method (see (Latombe, 1991) for an introduction). Initially, it was developed for path planning in a manipulation task by Khatib (1986). Potential functions are defined around obstacles and a goal state. The solution of the path planning problem is to follow the gradient of the superposition of these potentials, which eventually leads to the target configuration. In (Rimon and Koditschek, 1992) a wide variety of potential functions are introduced in the context of navigation. Arkin (1990) adapted this method to behaviour based robotics using the terminology of motor schemas (Arbib et al., 1987). Here, each behaviour defines a potential function based on the sensory context. The gradient of each function is determined at the position of the robot, which gives the desired direction of movement. By a weighted addition of the gradients from different behaviours, the motor action of the platform is determined. This method is widely used and particularly easy to implement using range sensors (Veelaert and Bogaerts, 1999). Another approach to superposition based fusion is the dynamical systems approach (Schöner and Dose, 1992; Schöner et al., 1995). This is the framework of choice for the work presented in this thesis and will be introduced in detail in section 2.2.

Planning and Reasoning

The behaviour based approach was novel in the aspect of distributing sensory processing and control to individual modules. However, central representations of physical objects and plans are needed to achieve higher level missions (Lyons, 1993; Chatila, 1995). In a hybrid deliberative architecture, a planner or any kind of reasoning system is using these representations to coordinate the activation of the individual behaviours. Note that this planning problem is easier than the one of classical AI approaches. Now, the planner operates in the "behaviour space" as opposed to the "action space". In other words, it has to choose between a, usually, rather small set of behaviours, instead of the set of all possible actions of the robot. A general formulation of the problem of combining reactive perception-action systems with an abstract reasoning level is presented in (Bajscy and Large, 1999).

Most of the behaviour coordination schemes introduced above were implemented in combination with a reasoning module as depicted in Figure 2.1. The integration of discrete event systems (Ho, 1991; Kosecká and Bajcsy, 1994) allows theoretical analysis of plan execution. A framework of combining a planner with robot schemas is presented in (Lyons and Hendriks, 1995). Also in approaches deploying behaviour fusion, the contributions of the individual behaviours can be governed by a planner. Rosenblatt (2000), for example, integrates this with a voting scheme by maximising a utility function. In the work of this thesis, the dynamical systems approach has been combined with a planner that finds a path through a topological map (section 3.3.2).

The reasoning parts of hybrid deliberative architectures are of many different types with various complexity. The planner in the navigation system presented of this thesis performs a simple search through a graph. Depending on the goals to be achieved, it can be more complex as the TCA system (Simmons, 1994) incorporating perception and improvement of efficiency. In (Peterson and Cook, 2003) the concept of uncertainty was integrated for a task of playing miniature golf. Often, general reasoning systems for robot control are not explicitly hybrid. They are designed in a hierarchical manner (Albus, 1991), where the lowest level of this hierarchy is equivalent to the reactive part of a hybrid deliberative architecture.

2.2 Dynamical Systems Approach

The dynamical systems approach unifies both the design of behaviours and their coordination in one framework. This approach is based on the theory of *nonlinear dynamical systems*. Here, we consider mainly the qualitative properties of the mathematical framework. Relevant concepts like attractor, repellor, bifurcation, and stability are introduced later in this chapter. For a thorough theoretical background on dynamical systems there are a number of excellent textbooks, for example (Perko, 1991) and (Khalil, 1995). An introduction into the topic in the context of autonomous agents can be found in (Beer, 1995).

The approach was originally motivated from biology. It has been found that the behavioural information in the brain is organised in a way that underlies some dynamics (Schöner, 1991). Furthermore, also the nervous system follows dynamical laws (Schöner and Kelso, 1988). This led to the idea of defining behaviours as dynamical systems in a space spanned by so-called behavioural variables (see section 2.2.1). Others have used these mathematical tools as well. For example, Beer (1995) uses it for analysis of a six legged agent. Here, dynamical systems are, however, used as a design principle, rather than an analysis tool.

The dynamical systems approach was first introduced by Schöner and Dose (1992). The methodology for behaviour design was applied to different navigation tasks consisting of some form of target acquisition and obstacle avoidance. Experiments were mainly performed in simulation (Schöner and Dose, 1992; Schöner et al., 1995; Steinhage and Schöner, 1997; Large et al., 1999) and some simplified real-world settings (Schöner et al., 1995; Neven and Schöner, 1996; Bicho and Schöner, 1997; Bicho et al., 2000; Bicho, 2000a). Moreover, the behaviour coordination scheme has been applied to simulation experiments only (Schöner and Dose, 1992; Schöner et al., 1995; Large et al., 1999). In the work presented here, the methodology has been applied for the first time to the design of a system acting in realistic real-world environments. In particular, the behaviour coordination scheme was exploited and explicitly connected to some planning module in order to allow mission planning. This is done for fetch-and-carry type tasks in an indoor environment. Furthermore, the dynamical systems approach was also applicable to navigation for two robots cooperating (Large et al., 1999) or moving in formation (Monteiro and Bicho, 2002). In addition, the same methodology was applied to navigation in human-robot interaction as a part of this thesis, which will be presented later in chapter 6.

The basic concepts of the dynamical systems approach are introduced below. These concepts will be further clarified in section 2.3 by means of the design of the navigation system. Other introductions into the topic and in-depth discussions of the motivation of the methods used can be found in (Schöner and Dose, 1992; Schöner et al., 1995; Steinhage, 1998; Bicho, 2000b).

2.2.1 Behavioral Variables

Modelling takes place in the space of *behavioural variables*, here, described by the vector $\vec{x}$. These variables define the behavioural dimensions; a continua along which behaviour can change. At any time, the state of the system is defined by a specific point in this space. The first design choice in the dynamics systems approach is to define the set of behavioural variables, such that the following conditions are satisfied:

- Task constraints of a behaviour b must be expressible as a point or set of points $\vec{x}_b$ in the space of these variables.

- The constraints $\vec{x}_b$ must be independent of the current state $\vec{x}$ of the system.

- It must be possible to specify the points defining a task constraint by collected sensory information or some internal world model.

- It must be possible to impose the time derivative $\dot{\vec{x}}$ of the behavioural variables on the actuator system of the robot.

In a navigation task, for example, the robot's heading direction ϕ, measured against a world-fixed reference direction, is usually one of those variables. The constraint for a target approaching task can be defined by the point ψ_{target}, the direction of the target from the robot's position. This value is independent of the robot's actual heading direction. The direction of this target, of course, must be known to the system. Furthermore, the time derivative $\dot{\phi}$ is the turnrate, which on most platforms can be directly controlled.

2.2.2 Behavioural Dynamics

The next design step is to generate *dynamical systems* of the behavioural variables. For each behaviour b the time evolution of these variables is described by

$$\dot{\vec{x}} = \vec{f}_b(\vec{x}, P) \tag{2.1}$$

The function $\vec{f}_b$ can be interpreted as a force acting on the behavioural variables dependent on the current state $\vec{x}$ and a set of additional parameters P, which depend on the relative pose between the robot and its environment. This force is designed in such a way that the point or the set of points $\vec{x}_b$ defining a task constraint are *fixpoints* of the dynamical system.

$$\left.\frac{\overrightarrow{dx}}{dt}\right|_{\vec{x}=\vec{x}_b} = \vec{f}_b(\vec{x}_b, P) = 0 \tag{2.2}$$

If the task constraint $\vec{x}_b$ is a desired value of the behaviour than the fixpoint must be an *attractor*. In the example above, this is the case for ψ_{target}. On the other hand, if the value of $\vec{x}$ defining the task constraint is an undesired one (for example the direction of an obstacle), the fixpoint has to be a *repellor*. This concept will become clear in section 2.3, where the design of the actual behaviours is presented.

Each attractor $\vec{x}_b$ can be characterised by its *strength*. This can be expressed by the slope of the dynamics at the fixpoint for each dimension i of the space spanned by the behavioural variables.

$$\lambda_{b,i} = -\frac{\partial f_{b,i}(\vec{x}, P)}{\partial x_i}\bigg|_{\vec{x}=\vec{x}_b} \tag{2.3}$$

After a perturbation in dimension i, the dynamics of behaviour b needs a certain time to approach the attractor state again. This *relaxation time* can be characterised by $\lambda_{b,i}^{-1}$; the inverse of the strength of the attractor. While the robot is moving through its environment, the parameters of the set P are gradually changing their values. This can bring along a shift of the attractors in the space of the behavioural variables. To ensure stability, which means that the system is in or close to an attractor state at all times, this shift must occur on a slower timescale than the attractor's relaxation time. This condition can be assured by choosing the robot's velocity and the strength of the attractors appropriately (see the design in section 2.3.2).

Furthermore, changes in a parameter of the set P can not only shift an attractor gradually, but also cause a *bifurcation* in the dynamical system. This means that an attractor can become a repellor or vice versa. In general, the set of fixpoints is altered, which brings along a complete change in strategy of the behaving system. This type of decision making will be illustrated, for example, with the design of an obstacle avoidance behaviour in section 2.3.1 (Figure 2.7).

To obtain the overall dynamics of the behavioural variables, multiple behaviours of a set B are aggregated by weighted addition of the individual contributions $\vec{f}_b$.

$$\dot{\vec{x}} = \sum_{b \in B} |w_b| \vec{f}_b(\vec{x}, P) + \text{noise} \tag{2.4}$$

The *weights* $w_b \in [-1, 1]$ define the level of contribution of each behaviour and are computed based on the perceived context of operation (see section 2.2.3). The noise has a small amplitude and merely ensures that the dynamical system escapes unstable fix-points (repellors). Note that summing is only a method of combining behaviours. This linear combination is not a constraint on the overall behaviour, since all the contributions are, in general, nonlinear.

By combining different behaviours, the *range* of each individual dynamical system in the space of the behavioural variables becomes important. This property determines, if two contributions interact with each other or if they are completely independent. Also this issue will be illustrated in section 2.3.1 by designing the actual behaviour for avoiding obstacles.

2.2.3 Competitive Dynamics

Coordination among behaviours is modelled by means of an additional *competitive dynamics* that controls the weights w_b for each behaviour b, which evolve in the following fashion:

$$\tau_b \dot{w}_b = \alpha_b (w_b - w_b^3) - \sum_{b' \neq b} \gamma_{b',b} w_{b'}^2 w_b + \text{noise} \tag{2.5}$$

The first term constitutes a pitchfork bifurcation, which means that the dynamics possesses stable fix-points at

$$w_b = \begin{cases} \pm 1 & \text{if } \alpha_b > 0 \\ 0 & \text{if } \alpha_b < 0 \end{cases} \tag{2.6}$$

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. In other words, a behaviour b is switched on for a positive competitive advantage and switched off if α_b is negative.

The need of activating a behaviour can often not only be determined by the environmental context. A certain behaviour can also conflict with another one being currently active. The second term in equation 2.5 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 $\alpha_b > 0$.

While the robot is moving, the environmental context is changing, which influences the parameters α_b and $\gamma_{b',b}$. This in turn modifies the amount and type (attractor/repellor) of fixpoints in the competitive dynamics, which can lead to a change in the navigation strategy. Hence, also on this level, decision making is modelled by bifurcations in a nonlinear dynamical system. A detailed analysis of how the stability of fixpoints varies across different values of competitive advantages and interactions is given in (Large et al., 1999). Similar to the behavioural dynamics, the noise term helps the system to escape unstable fixpoints (repellors).

Finally, instead of instantaneously switching behaviours on and off, they can be activated and deactivated gradually in this framework. The rate at which these transitions take place is determined by the *time constant* τ_b.

2.3 System Design

As outlined in the introduction, the system developed here provides the navigation functionality of fetch-and-carry type tasks in large-scale indoor environments. A typical mission of the robot starts at the charging station located in a room. Then, the platform has to drive to one or several goal points in the area, and eventually get back to recharge its batteries. It must be able to pass doors, drive along corridors and approach the targets, while avoiding collisions on its way. The following five behaviours with their associated functionality were designed:

- GO TO: moving towards a given goal point in an unstructured room in the absence of obstacles. This goal can also be some intermediate point of a longer mission (for example, the place in front of a door before leaving a room).
- OBSTACLE AVOIDANCE: avoiding collision with any kind of obstruction, while at the same time moving[1].
- CORRIDOR FOLLOWING: guiding the robot along corridors in the absence of obstacles.
- WALL AVOIDANCE: driving towards the middle of long corridors through avoiding its walls.
- DOOR PASSING: passing a narrow gap to traverse between two rooms, from a corridor to a room, or vice versa.

Following the methodology of the dynamical systems approach outlined in section 2.2, the behavioural variables $\vec{x}$ have to be defined first. The robot's heading ϕ relative to a world-fixed reference direction and the speed v of the platform were chosen.

$$\vec{x} = \begin{pmatrix} \phi \\ v \end{pmatrix} \tag{2.7}$$

Navigation behaviours can be expressed naturally in these terms, because constraints are usually defined by directions of objects (target or obstacles) and by restrictions on speed. Further, most mobile platforms accept control commands specifying the turnrate and the translational velocity. These values are provided by the behavioural dynamics. The turnrate $\dot{\phi}$ directly, and the velocity can be calculated by integrating the acceleration $\dot{v}$ for each timestep.

In order to keep the mathematical analysis of the dynamical systems as simple as possible, the dynamics of ϕ and v will not depend on each other for all behaviours (compare to equation 2.1).

$$\begin{pmatrix} \dot{\phi} \\ \dot{v} \end{pmatrix} = \vec{f}_b(\phi, v, P) = \begin{pmatrix} f_{b,\phi}(\phi, P) \\ f_{b,v}(v, P) \end{pmatrix} \tag{2.8}$$

Since, we now deal with two one-dimensional systems, the analysis of fixpoints and their stability properties becomes much easier. Hence, for each of the basic behaviours introduced above, two dynamical systems are designed. One for the heading direction (section 2.3.1) and one for the speed (section 2.3.2). It has been attempted to define these systems in the simplest mathematical form possible, such that the desired functionality of each behaviour is achieved. The design of the coordination among these basic behaviours is presented in section 2.3.3.

The explicit inclusion of the set of parameters P will be omitted in the remainder of this section. Moreover, all values denoting distances are expressed as a multiple of the robot's radius. This keeps the formulas simpler and the constants dimensionless.

[1] Standing still would be a rather good strategy for avoiding collisions; however, not very useful for achieving a navigational task.

2.3.1 Dynamics of Heading Direction

All dynamical systems defined in this section are in some way dependent on the angle under which an object (target, obstacle, wall, corridor, or door) is seen from the robot's actual position. These angles are measured from a world-fixed reference direction. These parameters change gradually as the platform moves around in the environment, which must be taken in consideration in order to assure that the heading direction ϕ stays close to an attractor state at all times. This can be guaranteed by choosing an appropriate balance between the relaxation time of the dynamical system (equation 2.3) and the robot's speed of motion. This poses constraints on the dynamics of v. Hence, the discussion of these stability considerations is postponed until section 2.3.2.

Go To

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

$$\dot{\phi} = f_{goto,\phi}(\phi) = -\lambda_{goto,\phi} \sin(\phi - \psi_{goal}) \tag{2.9}$$

The strength of the attractor (equation 2.3) is defined by the constant $\lambda_{goto,\phi} > 0$. A plot of the dynamical system in phase space can be seen in Figure 2.2. The intersection with the ϕ-axis defines the fixpoint ($\dot{\phi} = 0$). Since the slope of $f_{goto,\phi}$ at this intersection is negative, the fixpoint is an attractor.

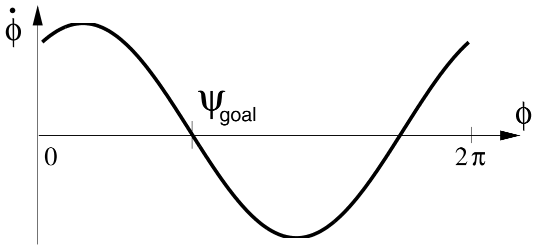

Figure 2.2. The dynamics of heading direction ϕ for GO TO. An attractor is generated at the direction ψ_{goal}, in which the goal point lies.

Obstacle Avoidance

The behaviour OBSTACLE AVOIDANCE is expected to turn the robot away from the direction of nearby obstacles. In case of a single obstacle i, the dynamics should create a repellor along the obstacle direction ψ_i. Since remote obstacles are less important than the ones nearby, the magnitude of the repellor should decrease with increasing distance to the obstacle. Moreover, an angular decay term captures the observation that obstacles along the current direction of motion pose a bigger threat than the ones on the side. All these criteria are met by the following dynamics:

$$f_{i,\phi}(\phi) = \lambda_{obst,\phi}(\phi - \psi_i)\, e^{-c_{obst} d_i}\, e^{-\frac{(\phi-\psi_i)^2}{2\sigma_i^2}} \tag{2.10}$$

The distance to the obstacle is denoted by d_i. The angular range of the repellor is defined by $\sigma_i > 0$. The parameter $c_{obst} > 0$ defines the decay of the strength of the repellor ($\lambda_{obst,\phi}$) with increasing distance to the obstacle. A plot in phase space of this system can be seen in Figure 2.3. The positive slope of $f_{obst,\phi}$ at the intersection with the ϕ-axis characterises a repellor.

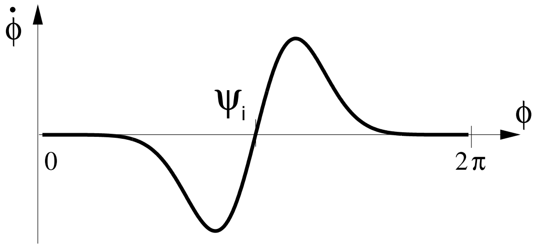

Figure 2.3. The dynamics of heading direction ϕ for OBSTACLE AVOIDANCE for a single obstacle. A repellor is generated at the direction ψ_i of the obstacle.

In case of multiple obstacles, the resulting force $f_{obst,\phi}(\phi)$ is computed by adding the contributions of individual obstacles.

$$\dot{\phi} = f_{obst,\phi}(\phi) = \sum_i f_{i,\phi}(\phi) \tag{2.11}$$

The platform is supposed to pass between two obstructions, if it is able to maintain a certain safety distance D_s to the obstacles located on either side of the robot. In other words, if the obstacles are too close to each other the dynamics should create a repellor along the direction of the gap (Figure 2.4). On the other hand, if the gap is sufficiently wide the dynamics should instead generate an attractor (Figure 2.5).

This form of decision making can be achieved by choosing the angular range σ_i (equation 2.10) appropriately. In order to do this, we have to examine the slope of f_{obst} at its fixpoint. If it is negative, the fixpoint is an attractor; if it is positive, a repellor. Let's consider the situation of two obstacles at equal distance (d_{obst}) from the robot, which is heading towards the middle of the gap ($\phi - \psi_i = \psi_j - \phi$) as

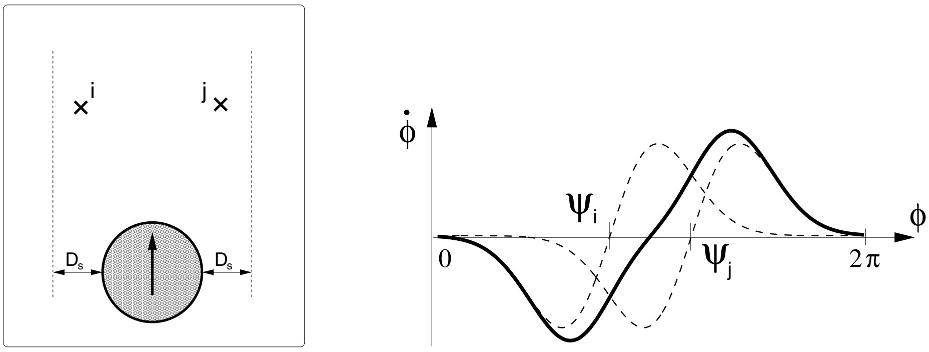

Figure 2.4. The dynamics of heading direction ϕ in the case of two obstacles i and j (dashed curves). If the gap between the two does not allow to stay a safety distance D_s away from them, a repellor is created.

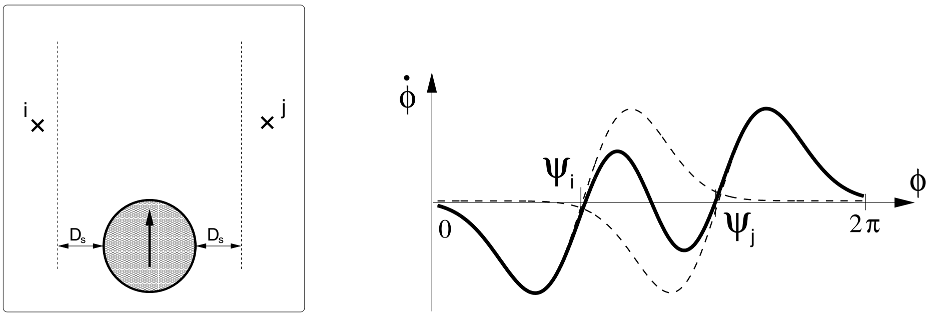

Figure 2.5. The dynamics of heading direction ϕ in the case of two obstacles i and j (dashed curves). If the gap between the two allows to stay a safety distance D_s away from them, an attractor in the middle of the two is created.

depicted in the two Figures 2.4 and 2.5. From equations 2.10 and 2.11 the slope of $f_{obst,\phi}$ for this situation can be calculated.

$$\left.\frac{d(f_{i,\phi}+f_{j,\phi})(\phi)}{d\phi}\right|_{\phi=\frac{\psi_i+\psi_j}{2}} = \lambda_{obst,\phi}\cdot e^{-c_{obst}d_{obst}-\frac{(\psi_i-\psi_j)^2}{8\sigma_i^2}}\cdot\left[2-\frac{(\psi_i-\psi_j)^2}{2\sigma^2}\right] \quad (2.12)$$

From this we can see that

$$\left.\frac{df_{obst,\phi}(\phi)}{d\phi}\right|_{\phi=\frac{\psi_i+\psi_j}{2}} \begin{cases} <0 \text{ (attractor)} & \text{if } \sigma < \frac{|\psi_i-\psi_j|}{2} \\ >0 \text{ (repellor)} & \text{if } \sigma > \frac{|\psi_i-\psi_j|}{2} \end{cases} \quad (2.13)$$

Let's now define the angular range σ_i in the following way:

$$\sigma_i = \arcsin\left(\frac{1+D_s}{1+d_i}\right) \tag{2.14}$$

In Figure 2.6 it can be seen that the dynamics erects a repellor ($\sigma > \frac{|\psi_i - \psi_j|}{2}$), if the gap is too narrow for the robot to pass ensuring a safety distance D_s. On the other hand, if the gap is wide enough, an attractor is created ($\sigma < \frac{|\psi_i - \psi_j|}{2}$).

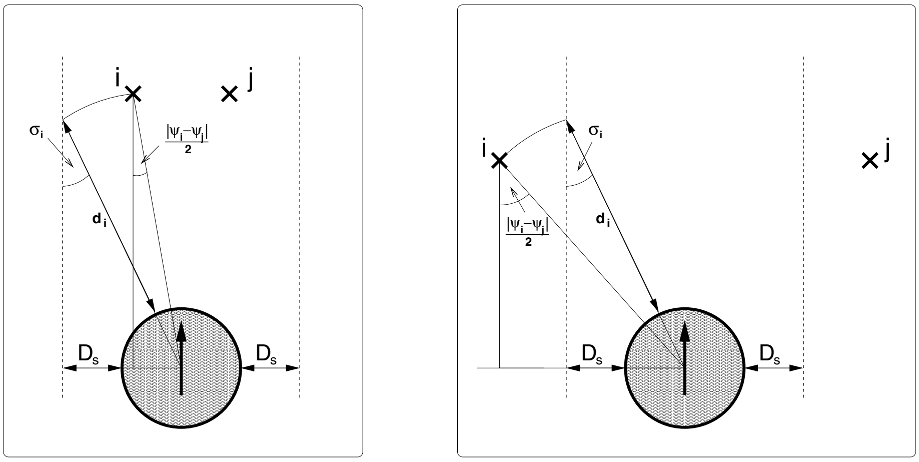

Figure 2.6. Illustration of the choice of σ. If the gap between the two obstacles i and j is too narrow (left image) , σ is greater than $\frac{|\psi_i - \psi_j|}{2}$. Thus, a repellor is created by the ϕ-dynamics (equation 2.13). If the gap is wide enough (right image), $\sigma < \frac{|\psi_i - \psi_j|}{2}$ holds, and an attractor occurs.

In this example, it can clearly be seen how decision making processes are modelled as bifurcations in the dynamical systems. As the distance between two obstacles changes, the amount and nature of the fixpoints is modified. A single repellor, which triggers turning away from the obstacles, becomes an attractor and two repellors as the distance between the obstacles grows. Hence, the latter situation makes the robot passing the gap. Figure 2.7 shows a plot of the fixpoints dependent on the distance D_{obst} between the obstacles. The safety distance was set to one robot radius. Thus the bifurcation occurs at $D_{obst} = 4$.

Corridor Following and Wall Avoidance

The behaviours CORRIDOR FOLLOWING and WALL AVOIDANCE navigate the robot along an empty corridor. CORRIDOR FOLLOWING is expected to align the robots

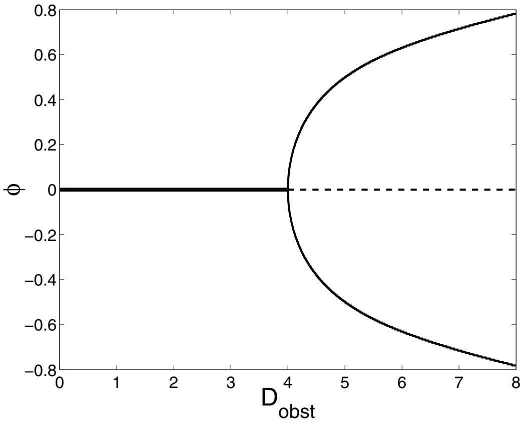

Figure 2.7. Repellors (solid line) and attractors (dashed line) of the ϕ-dynamics dependent on the distance between two obstacles D_{obst} (see Figures 2.4 and 2.5). Units on the x-axis are multiples of the robot's radius. The safety distance was set equal to 1. Thus, the bifurcation occurs at 4.

heading with the corridor direction ψ_{corr}, which the robot is supposed to follow. Hence, the behavioural dynamics has the same form as for GO TO (equation 2.9).

$$\dot{\phi} = f_{corr,\phi}(\phi) = -\lambda_{corr,\phi} \sin(\phi - \psi_{corr}) \tag{2.15}$$

WALL AVOIDANCE is supposed to guide the robot towards the center of the corridor. Thus, for each wall the dynamics contains a repellor located at the direction of these walls, ψ_{wall-1} and ψ_{wall-2} respectively; and an attractor along the opposite direction. The magnitude of the repellor should decrease with increasing distance to the wall at a rate determined by a gain $c_{wall} > 0$. The stronger contribution of the closer wall dominates the repulsive force of the remote wall in a way that results in a repellor generated along the direction of the former. Again, we require the function f_{wall} to be 2π-periodic. These criteria are met by a dynamics of the following form (Figure 2.8):

$$\dot{\phi} = f_{wall,\phi}(\phi) = \lambda_{wall,\phi} \sum_{l=1}^{2} \left[\sin(\phi - \psi_{wall-l}) \cdot e^{-c_{wall} d_{wall-l}}\right] \tag{2.16}$$

d_{wall-1} and d_{wall-2} denote the distances between the robot and each of the two walls.

Door Passing

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, ψ_{door}.

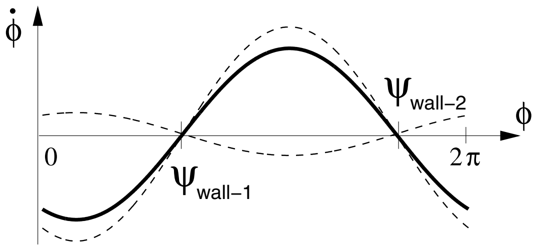

Figure 2.8. The dynamics of heading direction ϕ for WALL AVOIDANCE. Each wall provides a contribution (dashed curves) with an attractor repellor pair. The sum of the two contributions forms a repellor in the direction of the closer wall (ψ_{wall-1} in this case). An attractor occurs at the opposite direction ψ_{wall-2}.

Therefore, the same functional form as for GO TO (equation 2.9) was chosen:

$$\dot{\phi} = f_{door,\phi}(\phi) = -\lambda_{door,\phi} \sin(\phi - \psi_{door}) \tag{2.17}$$

2.3.2 Dynamics of Speed

Before the definitions of the dynamical systems for the robot's speed v are presented, an upper limit on this speed is established due to stability considerations. All the above introduced dynamical systems $f_{b,\phi}$ depend on an angle under which a certain object (goal, obstacle, corridor, wall, or door) is perceived. Usually, this angle constitutes a fixpoint in the dynamics of ϕ. However, it is changing its value, while the robot is moving through its environment. Hence, the fixpoints gradually move along the ϕ-axis. The rate at which these points are shifting must be slow enough for the dynamical system to stay close to the attractor at all times. This constraint can be fulfilled by choosing the maximum speed of the robot appropriately. How this is done is described in the following, taking the behaviour GO TO as an example.

While the robot is moving an infinitesimal distance $v\,dt$ the angle ψ_{goal}, at which the goal point lies changes by $d\psi_{goal}$, which can be calculated in the following way (see Figure 2.9):

$$|d\psi_{goal}| \approx |\sin(d\psi_{goal})| = |\sin(\phi - \psi_{goal})| \cdot \frac{v\,dt}{d_{goal}} \tag{2.18}$$

where d_{goal} is the distance from the robot to the goal point. During the same time, the robot's heading ϕ changes by a value $d\phi$ due to its own dynamics defined in equation 2.9.

$$|d\phi| = \lambda_{goto,\phi} |\sin(\phi - \psi_{goal})|\, dt \tag{2.19}$$

In order to ensure that the behavioural dynamics stays close to an attractor state at all times, the dynamics of ϕ has to take place on a faster timescale than the gradual shift of the angle ψ_{goto}; which means $|d\phi| \gg |d\psi_{goal}|$. This can be

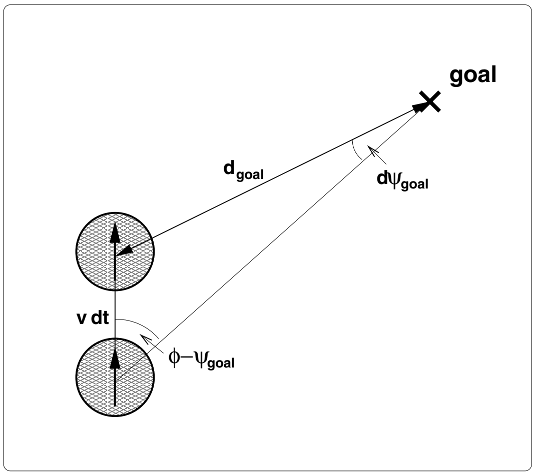

Figure 2.9. While the robot is moving a distance $v\,dt$ through its environment, the angle to the goal point changes by $d\psi_{goal}$ according to equation 2.18.

achieved by choosing the maximum speed dependent on the distance to the goal point.

$$v \leq k_{goto} d_{goal} \qquad \text{where} \quad k_{goto} \ll \lambda_{goto,\phi} \tag{2.20}$$

Substituting this value for the speed in equation 2.18 shows that $|d\phi| \gg |d\psi_{goal}|$ holds. Hence, stability of the ϕ-dynamics can be achieved by choosing appropriate values for the constants $\lambda_{goto,\phi}$ and k_{goto}. In other words, this is done by raising the strength of the attractor (increasing $\lambda_{goto,\phi}$) or by slowing down the platform (decreasing k_{goto}). These considerations must be taken into account in the design of the dynamics of speed for all behaviours, which are presented below.

Go To

The desired speed for the GO TO behaviour is in general quite high in order to reach the goal as quickly as possible. However, it must be restricted to some maximum value $v_{goto,max}$, because of limitations of the motor system and security reasons. Furthermore, the platform has to be slowed down as the distance to the goal point d_{goal} decreases due to the stability considerations outlined above. Hence, the desired speed for the GO TO behaviour becomes

$$v_{goto} = \min(k_{goto} d_{goal}, v_{goto,max}) \tag{2.21}$$

The simplest dynamics enforcing this speed is of a linear form (Figure 2.10):

$$\dot{v} = f_{goto,v}(v) = -\lambda_{goto,v}(v - v_{goto}) \tag{2.22}$$

Also here, the strength of the attractor (equation 2.3) is defined by a constant $\lambda_{goto,v} > 0$. This might look as the robot would be slowing down constantly as it is approaching the target. However in reality, k_{goto} can be chosen high enough, such that the robot mostly drives at the speed $v_{goto,max}$. Only when coming very close to the goal point, the platform is decelerating.

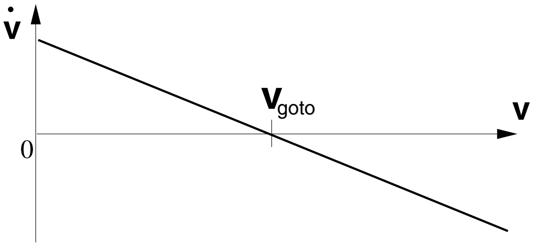

Figure 2.10. The dynamics of speed v for GO TO. At the desired velocity v_{goto}, an attractor is generated.

Also for this dynamics we have to make some considerations of the parameters to ensure that the dynamics stays close to an attractor state at all times. The attractor v_{goto} is shifting due to a change in d_{goal} when the robot is getting close to the goal point. If the robot is heading straight towards it, this change is at its maximum. In this case, the attractor moves by dv_{goto} during the time dt. From equation 2.21 we get

$$|dv_{goto}| = k_{goto}\,|dd_{goal}| = k_{goto}|v|\,dt \tag{2.23}$$

During the same time the speed v is changing due the behavioural dynamics (equation 2.22) by dv.

$$|dv| = \lambda_{goto,v}\,|v - v_{goto}|\,dt \tag{2.24}$$

The behavioural dynamics will converge to a point, where the attractor v_{goto} is running away at the same pace as v is approaching it. At this point $|dv_{goto}| = |dv|$ holds. To guarantee that this point is close to the attractor state v_{goto} we must to assure the following:

$$\frac{k_{goto}}{\lambda_{goto,v}} = \frac{|v - v_{goto}|}{|v|} \ll 1 \qquad \text{hence:} \qquad \lambda_{goto,v} \gg k_{goto} \tag{2.25}$$

In other words, choosing the strength $\lambda_{goto,v}$ of the attractor big enough, ensures that the system stays close to the attractor state v_{goto}.

Obstacle Avoidance

For the behaviour OBSTACLE AVOIDANCE there are basically two constraints posed on the speed of the robot. On the one hand, it has to be above some slow constant minimum speed v_{min} in order to keep moving and getting out of impasse situations. On the other hand, it must be below some maximum, which is decreasing as the robot gets closer to obstacles due to security reasons. For this maximum speed a linear dependency is chosen in the same way as in equation 2.20. These two constraints define a set of fixpoints as opposed to an isolated fixpoint, which was the case in all dynamical systems defined so far. Thus, the behavioural dynamics of a single obstacle i is defined as follows (Figure 2.11):

$$f_{i,v}(v) = \begin{cases} -\lambda_{obst,v}(v - v_{min}) & \text{for } v < v_{min} \\ 0 & \text{for } v_{min} \leq v \leq k_{obst} d_i \\ -\lambda_{obst,v}(v - k_{obst} d_i) & \text{for } v > k_{obst} d_i \end{cases} \tag{2.26}$$

d_i is the distance from the robot to obstacle i and the constant $\lambda_{obst,v} > 0$ defines the strength of the attractive region. To assure that both the dynamics of ϕ and the dynamics of v stay close to their attractor states, the same conditions as for the GO TO behaviour (equations 2.20 and 2.25) must be satisfied.

$$\lambda_{obst,\phi} \gg k_{obst} \qquad \text{and} \qquad \lambda_{obst,v} \gg k_{obst} \tag{2.27}$$

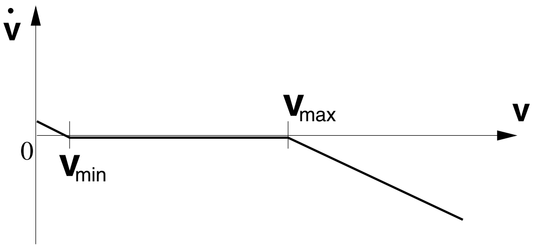

Figure 2.11. The dynamics of speed v for OBSTACLE AVOIDANCE for a single obstacle. The region between the two speed constraints v_{min} and v_{obst} is attractive.

As in the case of the dynamics of heading direction (equation 2.11) individual obstacle contributions are added to define the dynamics of speed for OBSTACLE AVOIDANCE.

$$\dot{v} = f_{obst,v}(v) = \sum_i f_{i,v}(v) \tag{2.28}$$

If we now combine two obstacles i and j at distances d_i and d_j from the robot; with let's say $d_i < d_j$. The upper bound of the attractive region is $k_{obst} d_i$. Hence the maximum speed is determined in a conservative fashion by the distance to the closer obstacle as illustrated in Figure 2.12.

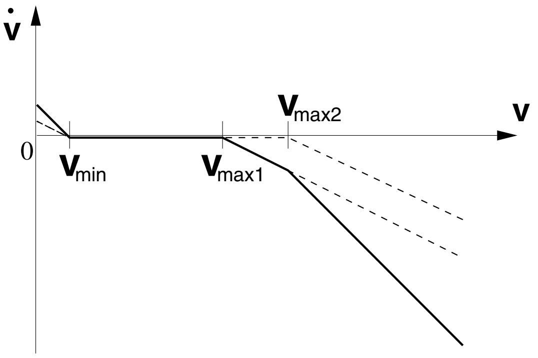

Figure 2.12. The dynamics of speed v for OBSTACLE AVOIDANCE for two obstacles at distances d_1 and d_2 from the robot (dashed lines). The overall dynamics (solid line) has an upper bound of the attractive region, which is determined by the closer obstacle (obstacle 1 in this case).

Wall Avoidance

For the behaviour WALL AVOIDANCE the same arguments as for OBSTACLE AVOIDANCE hold. Hence, each wall l contributes to the behavioural dynamics as follows:

$$f_{l,v}(v) = \begin{cases} -\lambda_{wall,v}(v - v_{min}) & \text{for } v < v_{min} \\ 0 & \text{for } v_{min} \leq v \leq k_{wall} d_i \\ -\lambda_{wall,v}(v - k_{obst} d_l) & \text{for } v > k_{wall} d_l \end{cases} \tag{2.29}$$

The overall dynamics of the speed v for WALL AVOIDANCE is obtained by summing the two wall contribution as in the case of the ϕ-dynamics in equation 2.16.

$$\dot{\phi} = f_{wall,v}(v) = \sum_{l=1}^{2} f_{l,v}(v) \tag{2.30}$$

Also here the conditions on the strengths of the attractors must hold.

$$\lambda_{wall,\phi} \gg k_{wall} \qquad \text{and} \qquad \lambda_{wall,v} \gg k_{wall} \tag{2.31}$$

Corridor Following and Door Passing

As for the ϕ-dynamics, the dynamics of the robot's speed v for the behaviours CORRIDOR FOLLOWING and DOOR PASSING have the same mathematical form as GOTO (equation 2.22).

$$\dot{v} = f_{corr,v}(v) = -\lambda_{corr,v}(v - v_{corr}) \tag{2.32}$$

$$\dot{v} = f_{door,v}(v) = -\lambda_{door,v}(v - v_{door}) \tag{2.33}$$

The stability consideration presented at the beginning of this section do not apply for CORRIDOR FOLLOWING, since the direction of the corridor ψ_{corr} (equation 2.15) does not change as the robot moves. Hence, v_{corr} is constant. However, for DOOR PASSING the distance to the door has to be taken into consideration analogously to the distance to a goal point (equation 2.21).

$$v_{door} = \min(k_{door} d_{door}, v_{door,max}) \tag{2.34}$$

$v_{door,max}$ is chosen rather small, since doorways are usually narrow. Finally, the same conditions on the parameters as for GO TO (equations 2.20 and 2.25) must be satisfied, of course.

$$\lambda_{door,\phi} \gg k_{door} \qquad \text{and} \qquad \lambda_{door,v} \gg k_{door} \tag{2.35}$$

2.3.3 Behaviour Coordination

The overall dynamics of the system is obtained from the weighted summation of individual behaviours based on equation 2.4:

$$\begin{pmatrix} \dot{\phi} \\ \dot{v} \end{pmatrix} = \sum_{b \in B} |w_b| \begin{pmatrix} f_{b,\phi}(\phi) \\ f_{b,v}(v) \end{pmatrix} + \text{noise} \tag{2.36}$$

with $B = \{goto, obst, corr, wall, door\}$. For the coordination of the behaviours, the competitive advantages α_b, the competitive interactions $\gamma_{b',b}$, and the time constants τ_b in the competitive dynamics of the weights w_b (equation 2.5) must to be chosen appropriately.

Competitive Advantages

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 (equation 2.6). 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.

$$\alpha_{goto} = \begin{cases} 0.5 & \text{if in a room} \\ -0.5 & \text{otherwise} \end{cases} \tag{2.37}$$

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

$$\alpha_{corr} = \alpha_{wall} = \begin{cases} 0.5 & \text{if in corridor} \\ -0.5 & \text{otherwise} \end{cases} \tag{2.38}$$

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

$$\alpha_{door} = \begin{cases} 0.5 & \text{if door detected} \\ -0.5 & \text{otherwise} \end{cases} \tag{2.39}$$

The relevance of OBSTACLE AVOIDANCE depends on the number and proximity of the obstacles currently surrounding the robot. Hence, the competitive advantage of OBSTACLE AVOIDANCE is related to the obstacle density ρ defined in the following way:

$$\rho = \sum_i e^{-d_i} \tag{2.40}$$

and is computed according to

$$\alpha_{obst} = \tanh\left(\frac{\rho - \rho_0}{\rho_0}\right) \tag{2.41}$$

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 $[-1, 1]$.

Choosing the parameters in this way, depending on sensory and topological context, incorporates both behaviour arbitration and fusion. Arbitration is achieved between GO TO and CORRIDOR FOLLOWING, for example. Only one of the two is activated depending on the location of the robot. Fusion occurs between GO TO and OBSTACLE AVOIDANCE, for example, by adding the two contributions (equation 2.36). In case an obstacle is blocking the way towards the goal point, the ϕ-dynamics guides the robot around the obstruction and the v-dynamics slows the robot down (Figure 2.13).

Another example of behaviour fusion is the combination of OBSTACLE AVOIDANCE and WALL AVOIDANCE. When the robot is travelling along a corridor containing obstacles the safety distance D_s (introduced in section 2.3.1) has to be kept to all obstructions. Thus, if a gap between an obstacle and a wall is wide enough the dynamics should create an attractor in the direction of this gap. So, similar considerations as in the case of two obstacles have to be made (see Figures 2.4 and 2.5). This poses some constraints on the decay factors c_{obst} and c_{wall}, and the strength of the slopes $\lambda_{obst,\phi}$ and $\lambda_{wall,\phi}$. However, solving the mathematical equations is somewhat more complex than in the case of two obstacles (equations 2.12 and 2.13). Hence, the relation of the parameters was determined numerically in order to achieve the desired behaviour.

Competitive Interactions

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.

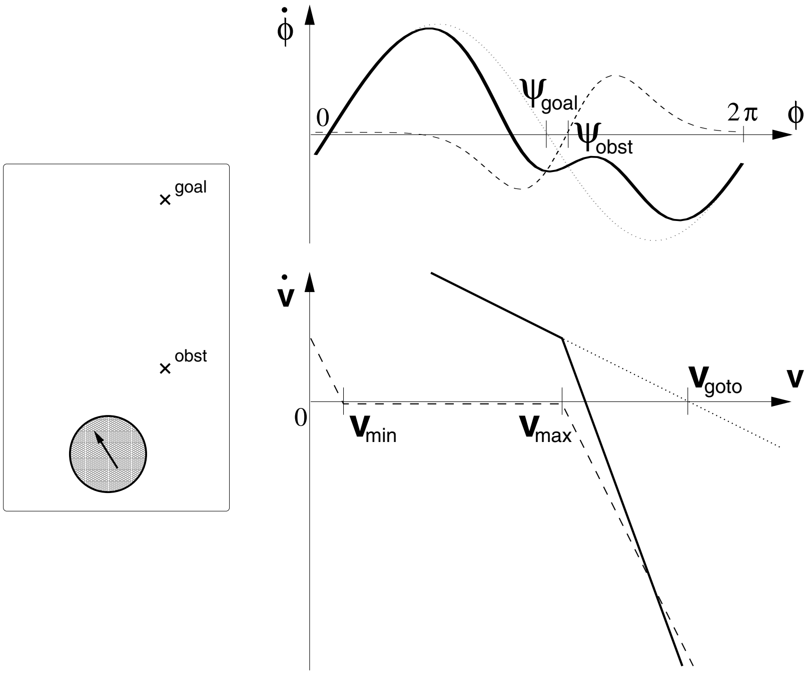

Figure 2.13. Fusion of GO TO (dotted curve) and OBSTACLE AVOIDANCE (dashed curve). Adding those two contributions gives the overall dynamics (solid curve). For the heading direction ϕ an attractor is created, which lies on the left of the goal direction ψ_{goal}. For the speed v the attractor lies close to v_{max} of OBSTACLE AVOIDANCE in order to slow down the robot.

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. This repellor, however, could be dominated by an attractor of another behaviour, which would inevitably lead to collision (Figure 2.14). Consequently, a behaviour arbitration mechanism is needed, where OBSTACLE AVOIDANCE ought to suppress GO TO and CORRIDOR FOLLOWING as well, if the obstacle density (equation 2.40) exceeds a critical threshold ρ_c. This prioritisation is achieved by appropriately choosing the competitive interactions:

$$\gamma_{obst,goto} = \gamma_{obst,corr} = \gamma_{obst,door} = \frac{1}{2}\left(1 + \tanh\left(\frac{\rho - \rho_c}{\sigma_\rho}\right)\right) \tag{2.42}$$

The constant ρ_c determines the density at which OBSTACLE AVOIDANCE suppresses the other behaviours ($\gamma_{obst,b} > 0.5$). The width σ_ρ defines the size of the region around ρ_c, where OBSTACLE AVOIDANCE influences the degree of activity of the

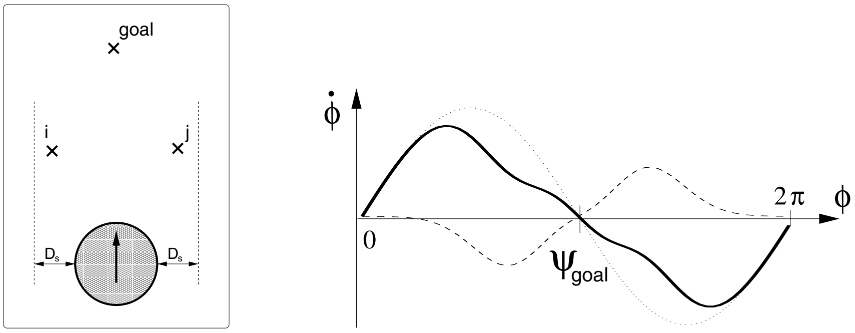

Figure 2.14. The contribution of two close obstacles i and j (dashed line) has a repellor in the middle of them (compare to Figure 2.4). However, this rather weak repellor is dominated by the attractor of GO TO (dotted line, see Figure 2.2). Hence the sum of the two (solid line) would ignore the obstructions. An arbitration scheme is needed to prioritise OBSTACLE AVOIDANCE.

other behaviours (without switching them off yet). 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.

Through this competition, another example of decision making by bifurcations is illustrated. This time in the dynamics of the weights w_b. Consider the situation depicted in Figure 2.14, for example. When the obstacle density is low, the weight w_{goto} possesses two attractors at ±1 and a repellor at 0. Thus, the behaviour GO TO is switched on. Then, as the robot approaches the obstacles, the obstacle density ρ increases, and so does the competitive interaction $\gamma_{obst,goto}$. When it exceeds the value of α_{goto} (0.5), a single attractor for w_{goto} is created at 0. Hence, at the bifurcation a decision is made by the system, and the navigational strategy changes completely. Figure 2.15 shows a plot of the fixpoints of w_{goto} dependent on the obstacle density ρ. The critical density ρ_c was set to 0.5. Thus, this is the value where the bifurcation occurs.

Time Constants

The time constants τ_b determine the time scale at which the behaviours are switched on and off. 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 (for example, when the robot is about to clear an impasse situation) the other behaviours switch on gradually. This often prohibits the robot

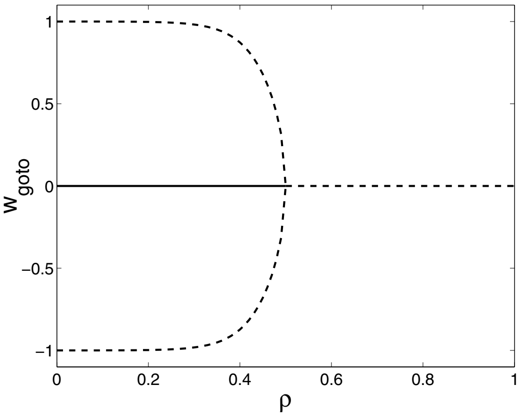

Figure 2.15. Repellors (solid line) and attractors (dashed line) of the w_{goto}-dynamics dependent on the obstacle density ρ (equation 2.40). The critical density ρ_c was set to 0.5. Thus, this is the value where the bifurcation occurs.

from driving back towards the same obstructions (see the results in section 4.2). Moreover, the slow time scale of these behaviours avoids jitter among themselves and OBSTACLE AVOIDANCE.

2.4 Discussion

The dynamical systems approach has been chosen as the framework for the behaviour based control system. Five behaviours were defined motivated by the general structure of an indoor environment: GO TO, OBSTACLE AVOIDANCE, CORRIDOR FOLLOWING, WALL AVOIDANCE, and DOOR PASSING. The individual task constraints could be mathematically formulated and were expressed as fixpoints in the dynamics of the heading direction ϕ and the speed v. Through theoretical analysis of the fixpoints, additional constraints could be incorporated, which determine the occurance of bifurcations. These bifurcations express decision making on the reactive level as seen in the case of two obstacles. Furthermore, the behaviour coordination scheme has been exploited. The robot's current context influences the parameters in the competitive dynamics, which activates and deactivates the behaviours on different timescales. It has also been shown how both arbitration and fusion can be achieved deploying this scheme.

The design of the dynamical systems for the control of heading direction follows closely (Bicho and Schöner, 1997) and (Bicho et al., 2000). In addition to their work, the speed control is fully integrated here. In earlier studies, this quantity is, usually, set to a constant value. This assumption actually violates the methodological constraint that the system must remain close to an attractor state at all times. Through the integration of the speed, this deficiency could be tackled as illustrated by the stability considerations in section 2.3.2. Another novelty of the implementation presented here, is the use of the dynamical systems approach for realistic real-world tasks. As opposed to simulation studies or the use of real robots on the small scale, the design as set forth here allows a system to navigate in a large-scale environment through appropriate behaviour switches. The integration of this arbitration mechanism into the competitive dynamics assures a smooth control of the platform. Whereas in other frameworks (for example, discrete event systems), discrete behaviour switching is applied, which results in an instantaneous change of the control strategy.

The dynamical systems approach is often compared to potential field methods. In that framework, individual behaviours also erect vector fields, whose weighted superposition determines the control action of the robot. However, the two approaches are very different in nature. In the potential field method, the vector fields are defined in the physical space. Their values determine the direction in which the platform should move at each point. The attractor of this field is the goal point, where the system ends up after completing its mission. In the dynamical systems approach, however, the vector field is defined in the space of the behavioural variables. This field is dynamic and changes as the robot moves about its environment, while the system is close to an attractor state at all times. These differences become obvious, when individual behaviours are combined. In the potential field methods, weighted averaging of the desired moving directions is the only possibility of behaviour combination. Whereas a much richer form of fusion is possible in the dynamical systems approach, since the attractor of the superposition can, basically, be everywhere in the space of the behavioural variables (compare Figures 2.4, 2.5, 2.12, and 2.13).

The dynamical systems approach was also used in other forms and for other tasks as presented here. Bicho and Schöner (1997), for example, chose the turnrate as behavioural variable in order to simulate phototaxis. Moreover, in (Schöner et al., 1995; Steinhage and Schöner, 1997) the common use of behaviour was stretched a bit by designing modules for ego-position estimation, for example. In addition, dynamic fields were used to define representations of the environment (Engels and Schöner, 1995). This was integrated with the dynamic systems for control into one coherent framework for a robotic system (Schöner et al., 1995). However, the use of dynamic fields for representing objects in the platform's surrounding, is computationally expensive and, therefore, not suited for real-world applications. Furthermore, the integration of planning in such an architecture is rather unclear. Hence, these extension of the approach have not been used in the work of this thesis. How such representations of the environment can be integrated into the system is the topic of the next chapter.

Chapter 3

Representations

In order for a control system to work in the real world, it needs to relate to geometrical and symbolic representations of the environment. The two terms, *geometrical representation* and *symbols* are closely related and often overlapping. Geometrical representations are typically used to determine the parameters in a control system. These representations range from simple measurements like distances to obstacles to more complex models of the world including specific features and maps. Symbols, on the other hand, is a term frequently used in AI. Decision making processes are usually based on symbols, which stand for labelled entities in the world.

For a navigation system, it can easily be seen that there is no clear distinction between these two terms. Two obstacles in the robot's path can be viewed as symbols, based on which the system makes a decision of passing between them or avoiding the gap, as illustrated in section 2.3.1. The properties of these symbols (for example their distance to the robot) are a geometrical representation of the current situation. Furthermore, a map of the whole area in which the system operates is a geometrical representation of the environment. However, this map can contain features of a symbolic nature, which are used to make decisions dependent on the navigation strategy. Hence, the terms are often used interchangeably in robotics literature. However, one problem has to be solved by all designers of robotic systems acting in the real world. How can information obtained by sensors be transformed to symbolic representations of the system? This process is commonly referred to as *symbol anchoring*. A general framework in the context of hybrid deliberative systems is proposed here. Its functionality is illustrated through the implementation of the indoor navigation system.

This chapter is organised in the following way. In section 3.1, a brief overview on geometrical representations for navigation systems is given. Further, the map used in the studies of this thesis is presented. Then, in section 3.2 the anchoring problem is discussed and the general framework is presented. The details on all the symbols of the navigation system and their anchoring processes is presented in section 3.3. The chapter concludes with a summary and discussion in section 3.4.

3.1 Geometrical Representations

Geometrical representations used by mobile robot navigation systems can roughly be divided into two categories: representations of individual objects close to the robot and maps reflecting the large-scale structure of the environment.

3.1.1 Local Representations

For mobile robots, local representations facilitate safe and precise navigation taking into account the small-scale structure of the platform's surrounding. Typically, these structures are directly extracted from sensory data. Hence, in a behaviour based approach, the extracted properties are, usually, directly fed to the individual behaviours, such that these modules can also react to unexpected events. The exact type of representation used by a system is highly dependent on the task at hand and the sensing capabilities of the robot. Thus, there are almost as many different implementations as there are robotic systems, and to give a complete overview is rather impossible. However, some commonly used strategies are summarised below.

The most common type of sensor used in navigation tasks is a range sensor: sonar or laser scanner. Based on the time of flight of an ultrasonic pulse or a laser beam respectively, an estimate to the closest object in the direction of the sensor can be obtained. This information can, of course, easily be used by an obstacle avoidance module. The properties of such an obstacle representation are manifold. In purely behaviourist approaches the estimate of the distance is directly used without further processing (Braitenberg, 1984; Brooks, 1986). A more complex and commonly used representation is the vector field histogram introduced by Borenstein and Koren (1991). This method allows to determine the properties of obstacles more precisely, taking into account the physics of the sensor and using multiple readings over time. Nevertheless, it has been shown to be computationally fast enough to work in real time. Range sensors are further commonly used to detect landmarks (Wijk and Christensen, 2000), walls (Forsberg et al., 1995), and other geometrical features of the robot's environment.

Another widely used sensor is a camera. An image of the robot's surrounding contains, of course, much more information than a sonar or a laser scan. This advantage, however, makes the extraction of representations also more complex. The research field of computer vision (Gonzales and Woods, 1992) is dealing with this problem of detecting objects in pictures. These techniques have successfully been applied to obstacle avoidance (Kosecká et al., 1995) and visual servoing (Hutchinson et al., 1996), for example.

In the work of this thesis a robot equipped with a ring of 16 sonars (see section 4.1) has been used. In order to allow safe navigation and dealing with local tasks, different geometrical representations are extracted from the sensory data: obstacles, corridors, and doorways. For the detailed properties and anchoring mechanisms of these objects see section 3.3

3.1.2 Maps

Maps represent, usually, the large-scale structure of an environment. They are essential for a mobile robot in order to reach a goal, which is far away and can not be detected by any sensors initially. Maps often constitute the a priori knowledge the system possesses about the environment. However, they can also be built or updated by sensory data. In general, maps can be divided in two categories: *geometrical* and *topological* maps. These two types are introduced below. Then, the map used in these studies is presented, which is a topological with some minimal geometrical properties.

Geometrical Maps

Geometrical maps are a rather detailed representation of an area. They are rich on information, which makes them convenient to use for navigation purposes given a good estimate of the robot's position. However, the large amount of information poses constraints on the storage capacity of a system. Further, this maps are rather hard to construct due to their detailed description of the environment. There are two main types of these large-scale representations: *grid maps* and *feature maps.*

A grid map divides the environment in many equally large cells of square shape. Each of these cells holds a number reflecting the probability that it is occupied. Occupied, here, means that anything could be there: a wall, a smaller obstacle, or even a person. Since there is no distinction of different objects, range sensors are the optimal choice for creating and using grid maps. Moravec (1988) and Elfes (1987) were the first to employ such a map for navigation purposes and, later, for path planning (Elfes, 1989). These approaches make use of a detailed sensor model and Bayesian statistics to update and use the map. These methods are widely used today and they have been further refined by applying more accurate sensor models (Konolige, 1997) and other forms of reasoning (Pagac et al., 1998).

A feature map reflects not only the occupancy of cells, but contains also information about the actual objects occupying the locations. This representation consists of features and their coordinates in the environment. These features are often lines (Crowley, 1985) or general geometric beacons like points, corners, and walls (Leonard and Durrant-Whyte, 1991), which are extracted by range sensors. The complexity of a feature map can vary from minimalistic models containing only walls (Jensfelt and Christensen, 2001) to a complete CAD model of the environment as in (Christensen et al., 1994). This type of maps have been widely used for different robotic applications. Many refinements have been developed like learning the features with artificial neural networks (Thrun, 1998) or tracking hypothesis about them over time (Arras et al., 2003).

Topological Maps

Topological maps reflect the large-scale structure of an environment containing information of low complexity. The space is segmented into topological units (places) and the connectivity of these units is given. Such a map is, mostly, represented by a graph structure with nodes standing for important places in the environment and edges defining how this places are connected. These types of maps are very compact representations and are, usually, easy to construct due to their low complexity. Another advantage of these maps is that they only contain information which hardly changes over time (rooms or corridors). Hence, they are still valid after, for example, refurnishing an office space. However, they are harder to use for navigation purposes than metric maps, because only limited knowledge about the robot's surrounding is available.

Kuipers and Levitt (1988) were one of the first using this concept. They defined a cognitive map on several levels of abstraction, where one of these levels was topological. They further extended their approach to learning a spatial semantic hierarchy of an area (Kuipers et al., 1993). Horswill (1998) constructed a system using a topological map specifically for office environments assuming angles of 90° between all corridor parts. Corridors can also be divided into large cells, where each cell defines a topological unit as done for the indoor navigation systems Dervish (Nourbakhsh et al., 1995; Nourbakhsh, 1998) and Xavier (Koenig and Simmons, 1998), for example.

Since these types of maps lack a lot of geometrical information, they are often combined with metric maps. One way to do this is defining some hierarchy of information in the environment (Kuipers and Levitt, 1988; Poncela et al., 2002). Another way is to divide a grid map into distinctive parts and defining their topological relation (Fabrizi and Saffiotti, 2002). In Tomatis et al. (2003) only the interesting parts of the environment are mapped geometrically while others (for example corridors) have a pure topological representation.

The Topological Map in these Studies

The map chosen for the navigation system presented in this thesis is almost purely topological. The main structure contains qualitative information about the large-scale connectivity of the environment. This information is reflected in a graph structure containing nodes and edges that connect these nodes. The nodes stand for important places in the environment and locations where a change in the navigational strategy occurs. Hence, there has to be one in front of each door, at each corridor crossing and at other places of interest (for example, 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 this map, it is a matter of minutes to construct a new one for a previously unknown domestic environment.

Figure 3.1 shows the map of our institute. As can be seen, it is topological in nature. Nevertheless, it also contains some minimal geometrical information reflected in the properties of the different nodes, which defines their location in the world. Figure 3.2 shows more details on the placement of these nodes. Nodes in corridors are in the middle of the two walls. The ones in front of doors are aligned

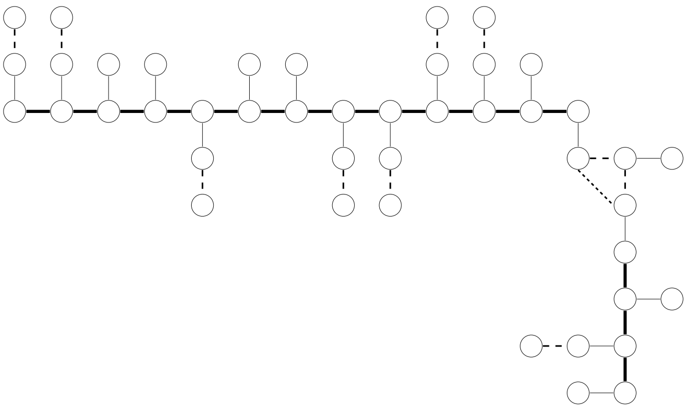

Figure 3.1. The topological map of our institute: The circles depict nodes, which have a location in a coordinate system. Edges are of three different types: corridor (thick line), room (dashed line), and door (thin line). Additional nodes for goal points and starting positions can be added arbitrarily.

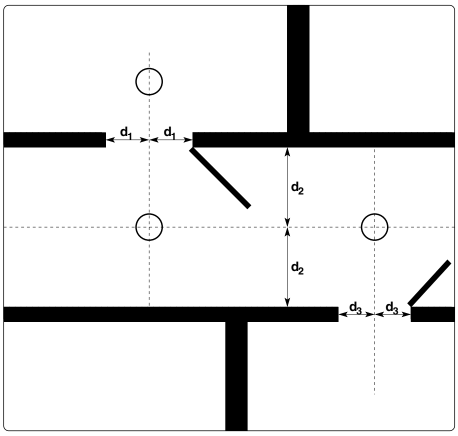

Figure 3.2. Placement of the nodes of the topological map. A schematic drawing of a corridor and two doors leading to rooms. The nodes (depicted by circles) are placed, such that they are centred in the corridor and between the door posts.

with the centre between the door posts. Further, nodes in rooms are positioned at places that are important for the navigational task. This placing allows the navigation system to effectively keep track of its position and orientation (see section 3.3.2). Nevertheless, these coordinates need not to be very accurate, because the nodes in combination with the robot's position estimate are only needed for task switching. Whereas guiding the robot through a door, for example, is controlled by the behaviours which extract the precise location of the door posts from sensory data. Note that there is some redundancy in placing some of the nodes. The distance of a door node in a room (on the top in Figure 3.2) to the actual doorway does not really matter as long as it is not more than about 2 meters. Also the position of a goal point does usually not have to be precise, because the controllers of the tasks to be executed at this goal point can compensate for that with their sensing capabilities. For example, visual servoing can guide the robot accurately in front of a table in order to pick up an object.

3.2 Symbol Anchoring

Symbols are discrete entities labelled with a name. They are, in general, extracted from sensory data and a predefined world model. Symbols are used for *decision making*, which refers to choosing one out of several control strategies, which lead to qualitatively different behaviour. In the dynamical systems approach, these decision making processes are implicitly modelled in the control laws through the use of bifurcation. Let's recall two examples from section 2.3. The gap between two obstacles influences the amount and nature of fixpoints in the dynamics of heading direction (Figure 2.7). Further, in Figure 2.15, it can be seen how an increasing obstacle density alters the fixpoints of the competitive dynamics of the weight w_{goto}, which leads to a task switch.

The process of relating the symbols to sensory data and the world model is commonly called *symbol anchoring*. Some general considerations to this problems are discussed in section 3.2.1. Then, in section 3.2.2, a general framework for anchoring in hybrid deliberative architectures is presented.

3.2.1 Introduction

The signal to symbol transformation is a long standing problem in artificial intelligence. The question is related to the issue of selecting information from a stream of incoming data. The problem has been addressed in AI in general and many related domains such as computer vision (often referred to as figure-ground segmentation) and robotics (mapping). Others have avoided the problem by specifying that everything is a symbol (Simon, 1981). For construction of real-world systems such as mobile robots it is, however, important to consider the relation between perceptual data and symbolic models of such percepts. In addition to the extraction of symbolic information, it is further necessary to consider the maintenance of such symbols over time. As part of this an obvious question is: when and how does one need to perform the anchoring of symbols and maintain their consistency over time? This is the issue addressed here in the context of mobile robotic systems. See (Coradeschi and Saffiotti, 2003) for a broader description of the general problem.

All designers of robotic systems acting in the real world have, of course, solved the anchoring problem in some way. However, it was performed on a system by system basis without a general methodology and explicit treatment of the problem. Hence, the anchoring processes are, usually, hidden somewhere in the code of the programs controlling the robots. Nevertheless, a general framework would be useful as a design methodology and, further, facilitate comparison of different systems. A first attempt to such a general framework has been outlined in (Coradeschi and Saffiotti, 2000) and extended in (Coradeschi and Saffiotti, 2001). Here, the anchoring is placed in a system context as seen in Figure 3.3. Sensory readings are transformed into symbolic information that can be used by a planner and a control system that transforms this information into specific signals carried out through the actuation system. In addition, the planner makes use of a world model that contains a priori information about the environment.

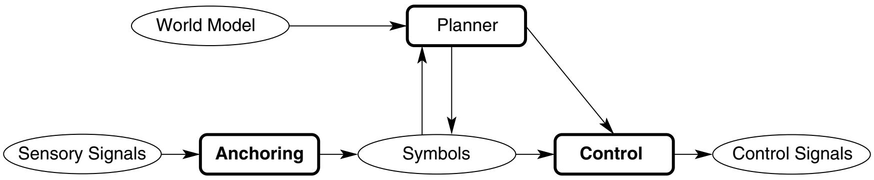

Figure 3.3. Information flow in a general control system.

The anchoring problem can be considered as three sub-problems:

1. Is the symbol to be extracted present at all (recognition/detection)
2. What are the exact properties of this percept (description/parameterisation)
3. How can the representation be updated over time (tracking)

In a general context, the problem of recognition or figure ground segmentation is considered to be NP-hard. The prototypical example being the Marr paradigm proposed in computer vision (Marr, 1982; Tsotsos, 1990). Through consideration of task constraints and careful selection of an appropriate control paradigm it might, however, be possible to convert the problem into a tractable one. An excellent example of how robotics problems can be made tractable is the use of active perception, in which task constraints are used for selection of sensory information and control of the associated processes, as described by (Aloimonos, 1993; Bajcsy, 1988)

As outlined in chapter 2, a lot of robotic systems today use a hybrid deliberative architecture, with a deliberative and a reactive part. The use of symbolic information across these two types of systems is different in the sense that the reactive system uses temporally local and precise information, while the deliberation part is dependent on global and contextual information. Consequently, the processing and handling of anchoring can be divided into two parts in terms of detailed anchoring and maintenance over time. Through adaptation of such an approach, it is possible to reduce the problem complexity considerably.

3.2.2 The Proposed Framework

In a hybrid deliberative system architecture (see section 2.1 for an introduction) the control of the robot is divided into two different levels: a *reactive level* and a *deliberative level*. On the reactive level control solutions to single tasks are implemented in behaviours. These behaviours solve their tasks based on representations that are closely related to immediate sensory input. The deliberative part of such a system is responsible for generating a list of tasks to be accomplished in order to achieve goals. Then, this plan has to be executed, which means the system has to switch between individual tasks. In essence, this level of control is responsible for the coordination of the different behaviours. In addition to sensory data, this coordination scheme makes use of a world model to create the plan and monitor its execution.

Both levels in such an architecture contribute to the control output of the system and include some type of decision making. Hence, they both have to deal with the problem of anchoring: creating and maintaining the correspondence between percepts (sensory information) and internal symbols. Here, it is proposed that the anchoring processes should be clearly distinguished for these two levels. By doing this, the symbol anchoring problems become easier in each of the two levels than it is in a general framework (Figure 3.3). This is due to the fact that not all the three problems identified in the introduction (section 3.2.1) have to be solved for each of the two parts. In addition, the extracted symbols from one level can be used by the anchoring processes of the other one. The proposed system organisation is depicted in Figure 3.4.

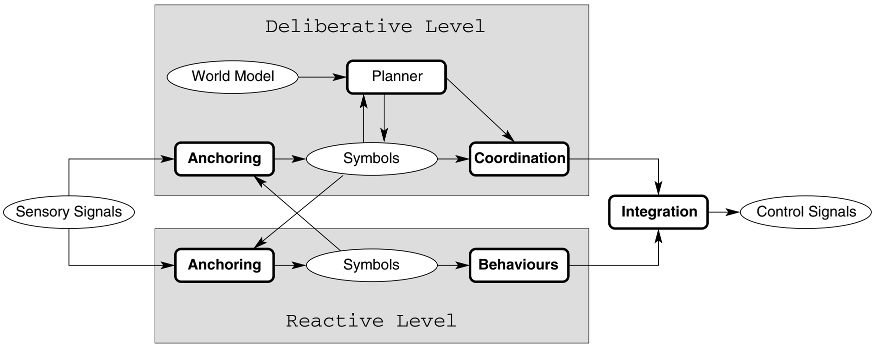

Figure 3.4. The proposed organisation for anchoring on different levels of abstraction.

Anchoring on the Reactive Level

The reactive part has the purpose of dealing with immediate tasks at hand through well defined behaviours. This undertaking requires its own special type of symbols and, thus, its own type of anchoring processes. Let us return to the three problems that a general anchoring process has to solve (section 3.2.1), viewed in the perspective of reactive control:

1. The question if a symbol is present at all in the sensory data does often not have to be addressed here. It can be answered by symbols at the deliberative level, since those are used for behaviour sequencing and and supervision of plan execution. In consequence, the symbols at the deliberative layer can select anchoring processes at the reactive level.

2. The question about the precise properties of a symbol has to be answered on this level. These properties are needed by the behaviours for the system to react in an appropriate fashion. For example, the location of obstacles have to be extracted accurately in order to avoid collisions, or the dimensions of an object to grasp have to be determined to allow visual servoing. Symbols at this level are, thus, predominantly specification of end-conditions and parameterisation of control.

3. The problem of maintaining the connection between a physical entity and an internal symbol of the system over time is of secondary importance at this level. Reacting to immediate sensory input is the main intention of behaviours, while maintaining connections to symbols over time and reasoning about them is of limited relevance to the reactive control structure. For example, properties of obstacles need only be maintained in a limited sense over time, as avoidance can rely on temporally local models and not necessarily on strict maintenance of information about the relevant entities.

Anchoring on the Deliberative Level

The deliberative part has the purpose of creating global plans and supervise the execution of such plans. This requires access to context information that situates the plan in the physical world, which is achieved through detection of certain events. This involves recognition of “locations” and “situations”, which allows the deliberative layer to detect the completion of a particular task. This undertaking requires different types of symbols than the ones of the reactive level and thus different types of anchoring processes. Again, let us return to the three problems that a general anchoring process has to solve (section 3.2.1), this time viewed in the perspective of deliberative control:

1. The question if a symbol is present at all in the sensory data has to be answered on this level (recognition), since the tasks to be executed depend heavily on this information. This problem is often hard to solve in a general anchoring process. However, it becomes easier for “simple” symbols like context information, which is often the type of symbols used by a deliberative module.

2. The question about the precise properties of a symbol is often less important here. A task coordination scheme does not need to know the exact properties of, for example, a box to pick up or an obstacle to avoid. These problems are handled by the behaviours in the reactive layer.

3. The problem of maintaining the connection between a physical object and an internal symbol of the system over time is of major importance here. To be able to reason about objects and to keep track of a plan execution, the anchoring process needs to provide this feature of updating the properties of symbols over time (maintaining a model of the environment). Since these properties are, usually, of a less complex nature than the symbols at the reactive level, the problem of keeping track of them is simplified. In addition, the precise properties of the symbols on the reactive level can often provide useful information to update the properties of the context symbols of this level.

The main advantage of the proposed separation is that the anchoring processes become easier to solve than in a general framework. Only a part of the three problems of a general anchoring process has to be attacked on each individual level. The reactive one can focus on the second question of extracting exact properties. The deliberative level, on the other hand, can work on information about context and solving problems one and three for these “simpler” symbols. Through this distribution of the anchoring problem the complexity of a general perception process can be decreased significantly.

3.3 Implementation

To illustrate the abstract organisation proposed above, the actual implementation for the navigation task in indoor environments is presented. The symbols (and their properties) of the system are, eventually, used by the behaviours and the coordinator to produce appropriate actuator signals as defined in section 2.3. The task at hand is navigation $A \rightarrow B$. Typically A is the recharging station and B can be a number of intermediate points to be traversed as part of a surveillance or delivery mission. The size of the environment is chosen to be a large area (20×70 meters, see section 4.1). If the battery level at anytime goes below a critical level, the mission is to be aborted and the robot is to return to a recharging station. The task was designed to allow the robot to:

- Perform navigation to known positions (recognition of places through anchoring at the deliberate level)

- Handle obstacles (react to unexpected situations, which requires identification of “situation” and switching of control)

- Handling of “system critical” events (requires abort of mission and replanning to achieve arrival at the recharge station)

- Navigation involves several different settings (rooms, doorways and hallways) each of which calls for different types of control.

This setup allows to illustrate anchoring at the reactive and the deliberative layer, and for signalling across the two layers. Further, sensory signals are provided by ultrasonic sensors, wheel encoders (odometry), and a voltage meter at the batteries (see section 4.1 for a detailed description of the robot and its sensors). In addition, the topological map introduced in section 3.1.2 is used as the world model of the planning module.

The actual control schemes have been presented earlier in this thesis (section 2.3). On the reactive level, each of the behaviours defines a dynamical system in the form of equation 2.8. The parameters of these controllers ($f_{b,\phi}$ and $f_{b,v}$) are provided by the symbols on the reactive level. A weighted summation of all behaviours (equation 2.36) is, then, applied to the robot's actuators. The weights used in this aggregation are determined by the competitive dynamics of equation 2.5. The symbols on the deliberative level specify the parameterisation for this coordination framework. In other words, the competitive advantages α_b and competitive interactions $\gamma_{b',b}$ are determined.

Figure 3.5 shows the navigation system in the structure of the organisation proposed in this thesis (compare to Figure 3.2.2). Below, we go into the details of the symbols used and their anchoring processes, separately for the reactive and the deliberative level.

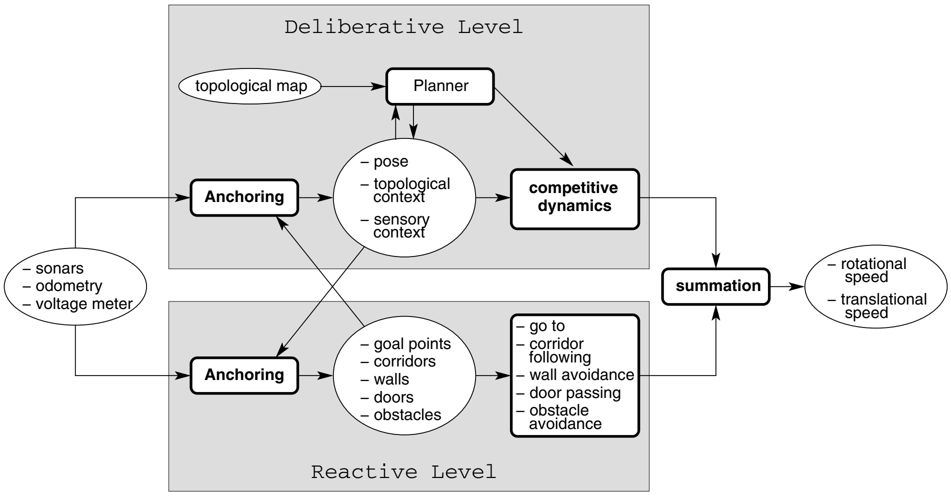

Figure 3.5. The organisation of the navigation system following the general methodology proposed in Figure 3.4.

3.3.1 The Reactive Level

On the reactive level safe navigation has to be assured taking into account the local properties of the environment. Recall section 2.3, where five behaviours have been designed to achieve this: GO TO to reach a given goal point, OBSTACLE AVOIDANCE to avoid collision with any kind of obstruction, CORRIDOR FOLLOWING and WALL AVOIDANCE to navigate along a corridor, and DOOR PASSING to traverse an open door. Each of these behaviours, defined by the dynamical systems $f_{b,\phi}$ and $f_{b,v}$, has its parameters based on the properties of some symbols. In the following, we introduce all these symbols and their anchoring processes. Remember that here, on the reactive level, the information if a certain object is present at all is usually given by the deliberative level of the system. The symbol TOPOLOGICAL CONTEXT of that level determines at what time the anchoring processes described below are invoked at all. The only exception are obstacles, which are extracted from the sensory data at every control cycle. Then, the importance of these processes is to obtain precise information about the physical entities to allow safe navigation. Further, the problem of maintaining these properties over time is neglected here as motivated in the general framework (section 3.2.2), which means that the behaviours rely only on temporally local properties. All the symbols and their associated properties for the reactive level are summarized in Table 3.1.

Symbol	Properties
GOAL POINT	direction, distance
CORRIDOR	direction, distance to wall 1, distance to wall 2
WALL	direction, distance
DOOR	detected, direction
OBSTACLE	direction, distance

Table 3.1. The symbols of the reactive level.

Goal Point

The symbol GOAL POINT is used by the behaviour GO TO. This is the only symbol on the reactive level that is not anchored using the sonar sensors. Its direction relative to the robot is determined using the symbols POSE and TOPOLOGICAL CONTEXT from the deliberative level introduced later in section 3.3.2. POSE provides an estimate of the robot's position and TOPOLOGICAL CONTEXT provides the coordinates of the next node (the goal point in this case) of the topological map (introduced in section 3.1.2). Then, the necessary parameters can easily be calculated.

The direction of the goal point relative to the robot, $\phi - \psi_{goal}$, is used by the dynamics of heading direction $f_{goto,\phi}$ (equation 2.9), while the dynamics of speed $f_{goto,v}$ (equations 2.21 and 2.22) makes use of the distance, d_{goal} from the robot to this point. There are no further properties to be determined for a goal point. However, in future implementations a certain object (for example a table or a person) could be considered a goal point. Then, there could be a need of an additional process to determine the properties of this object.

Obstacle

The symbol OBSTACLE is used by the behaviour OBSTACLE AVOIDANCE. Due to the limited angular resolution of sonar sensors, this symbol relies on a rather simple geometric representation that is closely linked to the actual perception of the robot. Out of the 50 most recent sonar readings that do not belong to detected walls (see symbols CORRIDOR and WALL below), 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.

For each obstacle i, the dynamics of heading direction $f_{i,\phi}$ (equations 2.10 and 2.14) uses the angle ψ_i and the distance d_i as parameters. Further, the dynamics of speed $f_{i,v}$ (equation 2.26) is dependent on the distance to the obstacle. Notice, that our representation only considers the direction and distance to an obstacle but ignores its shape and size. Despite its simplicity, the chosen representation is powerful enough to successfully navigate in cluttered areas (see the results in chapter 4).

Corridor

The symbol CORRIDOR is used by the behaviours 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 (Forsberg et al., 1995) is invoked on the sonar data every five 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.

From these lines, the properties of a corridor can be determined: its direction ψ_{corr}, the direction of the two walls ψ_{wall-1} and ψ_{wall-2}, and the distance from the robot to these walls d_{wall-1} and d_{wall-2}. These parameters define the form of the dynamical systems $f_{corr,\phi}$, $f_{wall,\phi}$, and $f_{wall,v}$ (equations 2.15, 2.16, 2.29, and 2.30). The dynamics of speed for CORRIDOR FOLLOWING $f_{corr,v}$ (equation 2.32) is completely independent of the corridor properties.

Door and Wall

The symbol DOOR is used by the behaviour DOOR PASSING that 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 the door to be open and its direction ψ_{door} is defined as the centre of the free segment. Further, if the robot is in a room the same strategy to detect a door is applied. However, first the properties of the symbol WALL at which the door is located have to be extracted. In order to do this, a Hough transform is invoked on the 100 most recent sonar echos.

The value of ψ_{door} influences the dynamics for heading direction $f_{door,\phi}$ (equation 2.17). The distance to this wall, or the corridor wall respectively, define d_{door}, which influences the dynamics of speed $f_{door,v}$ in equation 2.33.

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.

3.3.2 The Deliberative Level

In addition to sensory information, the system makes use of an a priori defined world model. This world model has the form of the topological map introduced in section 3.1.2. The planner module has two purposes to fulfill. First, given the starting node and the goal node in the topological map, it conducts a breadth first search through the graph to find a path to the goal. Then, during execution of the task, it updates some properties of the symbol TOPOLOGICAL CONTEXT (see below). Based on the battery level, it might also switch back to its first purpose and replan to drive back to the charging station.

At the deliberative layer the set of behaviours have to be selected for activation, which means the parameters for the dynamic weighting (the competition scheme introduced in section 2.2.3) have to be specified. These parameters are the competitive advantages α_b and competitive interactions $\gamma_{b',b}$ of each behaviour as defined in section 2.3.3. They rely on information about the robot's whereabouts (for example in front of a door) and the situation the robot is in (for example surrounded by obstacles). This facilitates switching between tasks like navigating along a corridor and passing through a door. Below, the necessary symbols are introduced the and their properties motivated. Remember from the general framework proposed in section 3.2.2 the type of anchoring problems that have to be solved here. The exact properties of the symbols are of minor importance for task switching, since the behaviours of the reactive level are responsible for precise navigation. However, updating the symbols over time and thereby keeping track of the task execution is crucial on the deliberative level. All the symbols and their associated properties for the deliberative level are summarized in Table 3.2.

Symbol	Properties
POSE	x coordinate, y coordinate, orientation
TOPOLOGICAL CONTEXT	location, next node x, next node y, in front of door
SENSORY CONTEXT	obstacle density, door detected, battery level

Table 3.2. The symbols of the deliberative level.

Pose

The symbol POSE contains an estimate of the robot's position and orientation. At the beginning of the execution of a plan, position and orientation of the robot are known (for example charging station). From there odometry is used to update the estimate of the robot's pose, which actually introduces errors. Anyway, this estimate is totally sufficient to determine, if the system is in the vicinity of a node defined in the topological map. Experiments showed that an accuracy of about 1 meter in enough to assure successful accomplishment of any mission. However, the error would grow bigger than desired on long trials over a great distance. To avoid this deficiency, the pose estimate is corrected based on detected symbols on the reactive level. Such a correction occurs on two different occasions: each time a corridor is detected and each time the robot has passed a door.

Each time a corridor is detected (every 5 seconds) a new estimate on the relative orientation of the corridor and the robot's distance to the two walls is provided (see section 3.3.1). The actual corridor orientation is the direction of the vector between the two nodes adjacent to the corridor edge (see Figure 3.2). This direction is known from the map and can, in turn, be used to update the estimate of the robot's orientation. Also the position of the centerline is known, which allows to correct the robot's location perpendicular to the walls.

Each time the platform is passing a door, the system keeps track of the narrowest gap, which defines the position of the door and the direction of the goal posts. Also here, the centerline through the door is known from the position of the adjacent nodes (see Figure 3.2). Hence, the robot's orientation and its position along the door posts can be updated correctly.

In addition, the location of the charging station is known exactly. Therefore, after each docking procedure the robots pose can be reset to a well defined value.

Topological Context

The symbol TOPOLOGICAL CONTEXT has a number of properties that reflect the robot's whereabouts in the topological map. These properties include "location" (living room, corridor, my office), "next node x", "next node y" and "in front of door" (true or false). "Location" is determined by the symbol POSE, which gives an estimate on the robot's position. "Next node x" and "next node y" are the coordinates of the next node in the topological map. They are influenced by the planner that keeps track of the path to be followed. The two together determine if the robot finds itself in front of a doorway that has to be traversed.

These properties influence the competitive advantages α_{goto}, α_{corr}, and α_{wall} (equations 2.37 and 2.38) of the task switching scheme, which reflect the applicability of a certain behaviour. Further, the property "in front of door" initiates the door detector module presented in section 3.3.1.

Sensory Context

The properties of the symbol SENSORY CONTEXT reflect the current situation the robot has to face. They are, in this implementation, "obstacle density", "door detected" and "battery level". "Obstacle density" is extracted from the symbols OBSTACLE from the reactive level according to equation 2.40. Its value influences the competitive advantage α_{obst} (equation 2.41) and the competitive interaction $\gamma_{obst,b}$ (equation 2.42). This determines the applicability of the behaviour OBSTACLE AVOIDANCE and, further, allows the robot to preempt a task in favour of this behaviour through the competition framework. "Door detected" uses the symbol DOOR and influences the applicability of DOOR PASSING by defining α_{door} (equation 2.39).

"Battery level" can have different levels (high, medium, low), which is determined by the signal from the voltage meter at the battery. This property might force the planner to abort a mission and find a path back to the recharging station.

3.4 Discussion

A framework for symbol anchoring in hybrid deliberative systems was presented. Analogous to the control processes in this approach, the anchoring processes are distributed on a reactive and a deliberative level. This makes the anchoring problem, in general, less complex and tractable. The implementation of the proposed framework for a navigation task validates its functionality in the real world. Although the anchoring processes themselves are rather simple and rely on low level sensors only, the system is completely capable of fulfilling its missions as will be seen in chapter 4. In addition, the distinction of anchoring on the two levels reduces the general problem complexity drastically. Hence, the system consumes less CPU-time. This characteristics is a major advantage to other approaches, which often struggle due to computationally expensive algorithms. For example, systems using grid maps

(Moravec, 1988), Markov decision processes (Koenig and Simmons, 1998), or Extended Kalman Filters (Leonard and Durrant-Whyte, 1991) run into this problem as soon as the amount of symbols (grid cells, states, landmarks) increases too much.

I believe that the system organisation proposed in this paper, can be applied to any robot deploying a hybrid deliberative architecture. The different nature of the symbols used on the two levels is implicitly given by using such an architecture. Then, not all the three sub-problems of a general anchoring process need to be attacked for each symbol.

In addition to sensory data, the deliberative level makes use of a predefined world model. This a priori knowledge is represented by a map of the environment as in most navigation systems. A topological map was chosen, which contains some minimal geometrical properties; namely the coordinates of the nodes. The information in this map is of a rather low complexity. However, it enables the control system to coordinate the individual behaviours and achieve switching between navigational subtasks. It will be shown in the next chapter that these subtasks are successfully executed without any precise estimate of the robot's position and orientation. Another advantage of the simplicity of the topological map is that it can be easily acquired by the system itself. Such an automatic map acquisition scheme will be presented in chapter 5.

Chapter 4

Evaluation

The navigation system introduced in the previous two chapters has been tested extensively in real-world experiments. Section 4.1 introduces the robots and the test environments. Then, in section 4.2, the system is evaluated by studying example trajectories. A few issues on symbol anchoring are discussed in section 4.3 and conclusions are drawn in section 4.4.

4.1 Robot and Environment

Experiments were performed on the premises of the Computational Vision and Active Perception Laboratory at KTH. The whole environment has a size of about 20×70 meters. A rough layout of the institute can be seen in Figure 4.5 in the next section. The topological map of this setting was introduced in section 3.1.2 and is depicted later in Figure 4.6. Although the environment is of a rather large size, it was a minor task to obtain the map due to its simplicity.

As an experimental platform the Scout robot from Nomadic Technologies (Figure 4.1) was used. 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. Each sonar has a beam width of 25° and a detection range of 15 cm to 6.5 m. Other platforms at the institute have more sophisticated sensing capabilities. Nevertheless, sonar based sensing is enough to demonstrate navigation and to verify the methods presented in this thesis. Further, the robot possesses a two wheel differential drive located at the geometric center, which allows omni-directional steering at zero turning radius. The odometric information is obtained from encoders on the two wheels.

The power system has been extended with two electric contacts at the rear of the robot. This enables the platform to autonomously dock with a power supply in order to recharge its batteries without human interaction. This procedure is not implemented as a behaviour and not incorporated in the framework of Figure 3.5 in

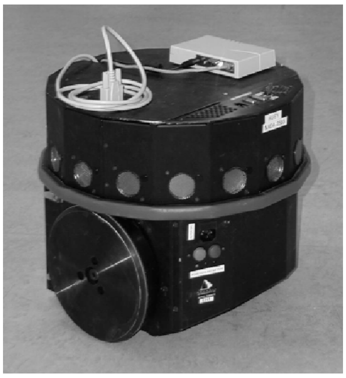

Figure 4.1. The Scout robot used in the experiments.

section 3.3. The docking task is performed by a stand-alone module that interacts directly with the hardware (Wulf, 2001). This module takes control as soon as the robot has reached the node of the docking station. Through active IR beacons, the charging station can be detected and, subsequently, the robot does the docking procedure. This procedure is aborted as soon as a close obstacle is detected. If this is the case, the general navigation system drives the robot back to a position in front of the charging station and the docking is reinitialized.

In addition, the navigation system has been implemented on another robot in a different environment. This system was a humanoid-like robot built on a Pioneer platform as depicted in in Figure 6.1 in chapter 6. The institute of the Intelligent Robotics and Communication group at ATR in Japan served as a test environment. Also this robot is equipped with 16 ultrasonic sensors and a differential drive. The sonars are placed somewhat differently, 8 in the front and 8 in the back. The Pioneer base plate has a diameter of about 45 cm. However, the upper body is larger and constitutes an effective diameter of about 70 cm. Due to this large size, many doors were too narrow for the robot to pass and, at the same time, ensuring a certain safety distance to the door posts. However, the part of the environment, which is accessible by the robot, was big enough (approximately 20×35 meters) to test the navigation system. A topological map of this setting is shown in Figure 4.2.

4.2 Office Navigation

The navigation system has been tested extensively in the area of our institute using the Scout robot introduced above. It successfully avoided collisions and was able to reach any goal point specified in the topological map. Such experimental results are very hard to quantify as outlined in (Gat, 1995). Hence, a few example missions are

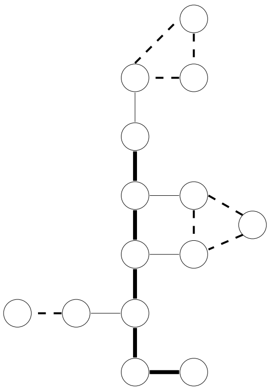

Figure 4.2. The topological map of the environment at ATR. The circles depict nodes, which have an exact location in a coordinate system. Edges are of three different types: corridor (thick line), room (dashed line), and door (thin line). Additional nodes for goal points and starting positions can be added arbitrarily.

discussed to show the functioning of the concepts. To illustrate the obstacle avoidance performance, a trajectory of the platform driving along a corridor is discussed in section 4.2.1. Then, in section 4.2.2, a longer trial is studied to demonstrate the successful use of the topological map.

Experiments were also performed on the premises of ATR using the humanoid-like robot described above and the topological map of Figure 4.2. In these experiments, all the parameters in the control scheme and the anchoring processes were set to exactly the same values as for the experiments with the Scout robot in our institute. The only exception was the value for the robot's radius for obvious reasons. The system at ATR behaved the same way as the Scout. It was also able to reach any position specified in the topological map while successfully avoiding collisions. All the data discussed below are taken from experiments at our institute at KTH. However, they could equally well be from an experiment with the platform at ATR. With the only difference that the gap between obstacles had to be larger, since the humanoid-like robot has a larger radius.

4.2.1 Driving along a Corridor

In this section, the performance of the behaviours OBSTACLE AVOIDANCE, WALL AVOIDANCE, and CORRIDOR FOLLOWING is illustrated. For example, the occurance of the bifurcation in the case of two obstacles (see Figures 2.4 and 2.5) can be observed as predicted by the design of the dynamical systems. The environment was set up accordingly as depicted in Figure 4.3. In this figure the trajectory of the robot has been plotted. However, this line is drawn from collected data of the robot's position estimate, since there were no means of retaining the real position of the robot. Nevertheless, this gives an accurate enough idea of the robot's track. Because the estimate of the position is updated every time the corridor walls are extracted (see section 3.3.2), there are some discontinuities in the plots. Thus, some vertical jumps can be observed in the track, for example at points B and and C. These are artifacts of updating the position estimate and do not reflect the real trajectory of the robot. Figure 4.4 illustrates the activity of the different behaviours by plotting the absolute values of their weights (equation 2.36). The letters on the time axis correspond to the positions labelled in Figure 4.3. These situations are discussed in more detail below. Note that the time difference between two successive tics is not proportional to the path length between the corresponding events, as the robot does not move at a constant speed. The whole trial depicted in Figure 4.3 lasted about 2 minutes.

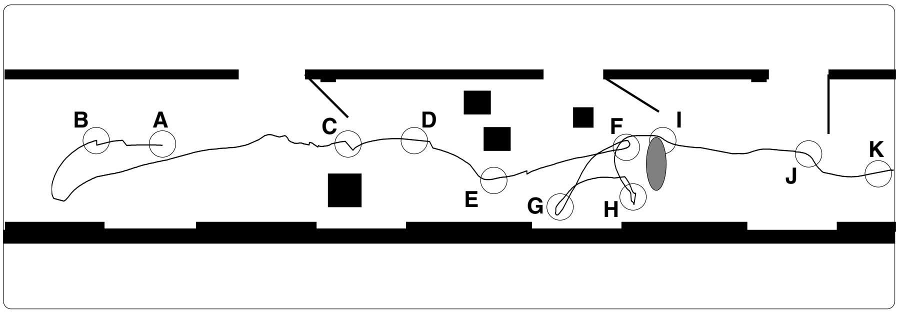

Figure 4.3. The trajectory of the robot in a corridor of a width of about 2.2 meters. There are three open doors to offices. The black rectangles denote obstacles (cardboard boxes). The gray ellipse shows a person standing in the way. This person is leaving shortly after the robot has passed position H. The situations labelled by the symbols A-K are explained in the text. The circles at these locations depict the size of the robot.

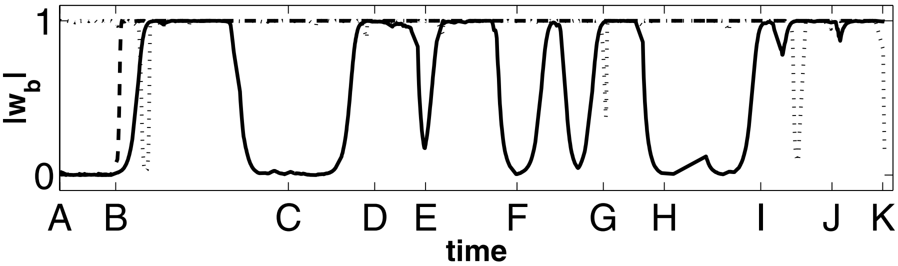

Figure 4.4. Time plot of the absolute values of the weights: $|w_{obst}|$ (dotted curve), $|w_{corr}|$ (solid curve), and $|w_{wall}|$ (dashed curve). The time instances labelled by the symbols A-K correspond to the situations in Figure 4.3.

A: The starting position: The robot was supposed to travel along the corridor towards the right but was initially heading in the opposite direction. The behaviours CORRIDOR FOLLOWING and WALL AVOIDANCE remained switched off until a sufficient number of sonar readings was collected in order to compute the Hough transform (section 3.3.1). OBSTACLE AVOIDANCE was turned on immediately due to detection of the walls. It remained switched on during most of the trial, because the obstacle density ρ (equation 2.40) was usually above ρ_0 (equation 2.41).

B: Enough sonar readings were collected: The direction of the corridor and the distances to the walls were estimated. WALL AVOIDANCE was switched on rather quickly and remained active for the rest of the trial, since the robot did not leave the corridor. The weight of CORRIDOR FOLLOWING increased on a slower timescale (section 2.3.3) and the robot started to align its heading with the desired corridor direction.

C: The robot encountered two obstacles: The gap between the obstacles was chosen such that the robot was able to keep a safety distance D_s on either side just by a few centimeters. Hence, OBSTACLE AVOIDANCE created an attractor in the middle of the two obstructions (Figure 2.5) and guided the robot through the gap. At this time, CORRIDOR FOLLOWING was switched off, since the obstacle density was above the critical value ρ_c (equation 2.42).

D: The gap was passed: CORRIDOR FOLLOWING was turned on again and the robot continued on its way to the right.

E: An obstacle in the middle of the corridor: The gap between the wall and the obstacle was again just wide enough that the platform could keep the safety distance D_s on either side. Thus, the combination of WALL AVOIDANCE and OBSTACLE AVOIDANCE created an attractor in the middle and guided the robot through the gap. When the robot was closest to the obstacle, the weight of CORRIDOR FOLLOWING was reduced for a short while.

F: A person was blocking the corridor: The person was positioned such that the gaps on either side are just a few centimeters smaller than the ones at positions C and E respectively. Hence, the robot could not pass between the two obstructions and OBSTACLE AVOIDANCE erected a repellor in the direction of the gap (Figure 2.4). However, the activity of CORRIDOR FOLLOWING would have made the platform collide (Figure 2.14). Through the competitive dynamics (equation 2.5) CORRIDOR FOLLOWING was switched off, because the obstacle density increased beyond the critical value ρ_c (equation 2.42). Thus, the platform avoided the obstructions.

G: The obstacle density decreased: After the robot drove away from the obstruction, the obstacle density decreased and CORRIDOR FOLLOWING was slowly switched on again. Hence, the platform headed towards the right again.

H: The person was still blocking the corridor: The person has not moved and also the gap along the wall is too narrow to pass. Thus, the obstacle density increased, CORRIDOR FOLLOWING was switched off, and the obstruction is avoided again.

I: The person disappeared: The person walked away shortly after the platform has passed point H. Hence, the way was clear and CORRIDOR FOLLOWING led the robot further along the corridor.

J: An office door is wide open: This obstacle was easily circumnavigated by the platform (Figure 2.13).

K: The corridor was completely clear: The robot continued its way along the corridor. Since there were no obstructions present at this point, the obstacle density ρ decreased even below ρ_0 (equation 2.40) and OBSTACLE AVOIDANCE was switched off.

4.2.2 A Trial over a Long Distance

Now, a longer trial is discussed. It is demonstrated how the system is able to update its position estimate well enough to reach goal points far away. Figure 4.5 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 interesting parts of the trial in terms of behaviour coordination are in the areas of the rectangles denoted by 1 and 2. These regions are shown enlarged in Figures 4.7 and 4.9. The track through the topological map can be seen in Figure 4.6. During this trial the robot covered a distance of about 50 metres.

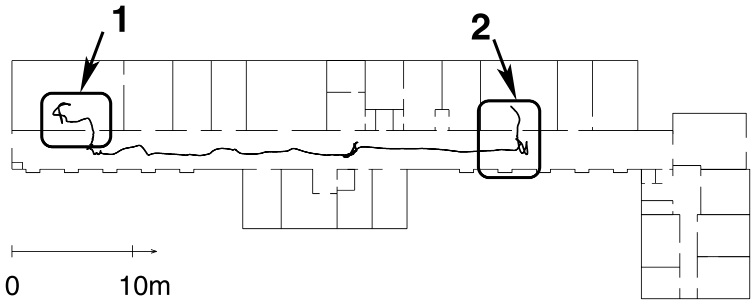

Figure 4.5. The trajectory of the robot in a typical task driving through our institute (from left to right). The rectangle denoted by 1 is shown enlarged in Figure 4.7; the one denoted by 2 in Figure 4.9.

Figure 4.7 shows the trajectory at the beginning of the mission. The robot started at the docking station in the living room and was leaving the room towards

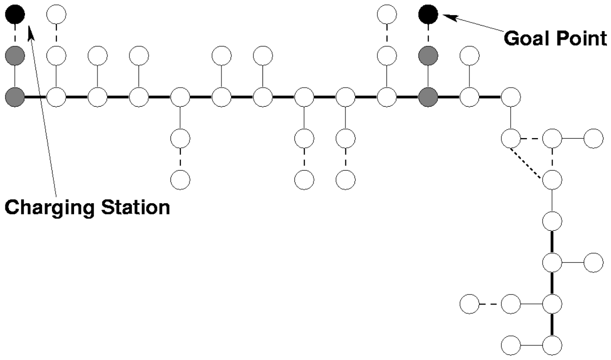

Figure 4.6. The track through the topological map for the trajectory in Figure 4.5. This typical mission begins at the node denoted with "start" (charging station) and ends at the one denoted with "goal". The nodes in gray are the ones used to execute this mission.

the corridor. Different situations are denoted by the symbols A-G, which are described in the text below. Figure 4.8 depicts the evolution of the weights of the behaviours. The labelled tics on the time axis refer to the corresponding locations of the robot.

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 ρ (equation 2.40) was always above ρ_0 (equation 2.41).

B: GO TO was gradually switched on: The behaviour GO TO evolves on a slower time scale than OBSTACLE AVOIDANCE (section 2.3.3). 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 2.42). 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 (compare to Figure 2.5). GO TO was off, due to a high obstacle density.

F: The vicinity of the next node was reached: Through the systems estimate of the robot's position, it was noticed that the next node in the topological map

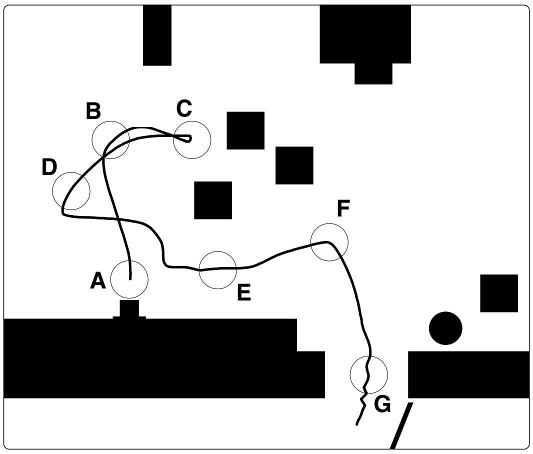

Figure 4.7. The trajectory of the robot starting at the charging station (A) and leaving the room towards the corridor (G). The black obstacles denote chairs, two shelves, a table and a waste bin. The situations labelled by the symbols A-G are explained in the text. The circles at these points depict the size of the robot.

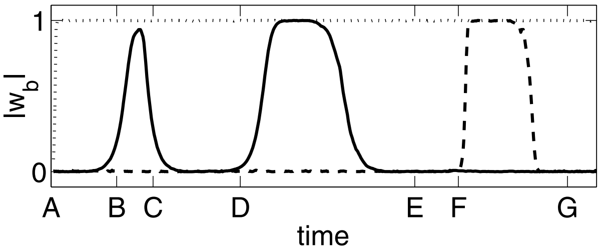

Figure 4.8. Time plot of the absolute values of the weights: $|w_{obst}|$ (dotted curve), $|w_{goto}|$ (solid curve), and $|w_{door}|$ (dashed curve). The time instances labelled by the symbols A-G correspond to the situations in Figure 4.7.

was reached. Then, the direction of the door was extracted from the sonar data (see Section 3.3.1). DOOR PASSING was turned on almost immediately, and the robot turned towards the 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.

Then, the robot continued its mission towards the manipulator lab. About in the middle of the corridor, the way was blocked by people. Hence, the robot drove in circles for a while, similar to the situations F-H in Figure 4.3. This can be recognised in the trajectory of Figure 4.5. This confusing element was an attempt to accumulate odometric errors and decrease the accuracy of the position estimate. Nevertheless, by using the extracted corridor walls the system could correct for these errors and, later, successfully recognize the node in front of the manipulator lab.

Figure 4.9 shows the trajectory at the end of the mission. The robot reaches the door to the manipulator lab and enters it to reach its goal point. To further confuse the system, this door was initially blocked by a person. Different situations are denoted by the symbols H-O, which are described in the text below. Figure 4.10 depicts the evolution of the weights of the behaviours. The labelled tics on the time axis refer to the corresponding locations of the robot.

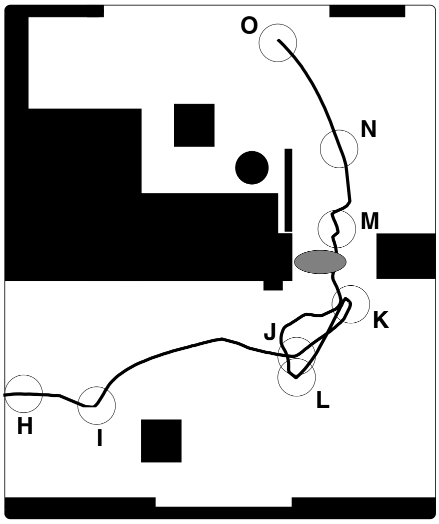

Figure 4.9. The trajectory of the robot from the corridor (H) to a goal point in a room (O). The grey ellipse denotes a person that was leaving the room, when the robot was at location K. The situations labelled by the symbols H-O are explained in the text. The circles at these points depict the size of the robot.

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 and the obstruction was circumnavigated (compare to Figure 2.13).

J: The vicinity of the next node was reached: Although the robot has covered quite a large distance, the position estimate was still rather accurate and the presence of the next node in topological map could be detected. DOOR

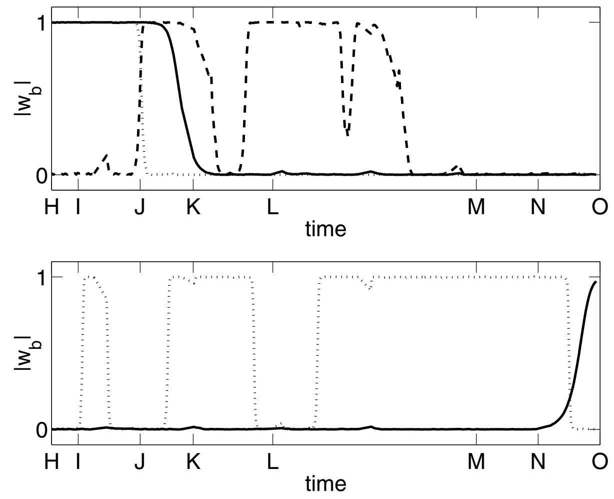

Figure 4.10. Time plot of the absolute values of the weights: $|w_{corr}|$ (upper plot, solid curve), $|w_{wall}|$ (upper plot, dotted curve), $|w_{door}|$ (upper plot, dashed curve), $|w_{obst}|$ (lower plot, dotted curve) and $|w_{goto}|$ (lower plot, solid curve) The time instances labelled by the symbols H-O correspond to the situations in Figure 4.9.

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 2.42). 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.

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.

Note that as mentioned earlier in section 4.2.1, not the real trajectories of the robot are displayed in Figures 4.7 and 4.9, but the position estimates of the system. Hence, the sharp bend right after passing the door at point M in Figure 4.9 is 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 3.3.1).

4.3 Anchoring Issues

Remember from section 3.2.2 that the anchoring processes have been simplified according to the level of abstraction. On the reactive level, the connection of the symbol to the physical object was not maintained over time. On the deliberative level, precision of the symbol properties was considered secondary. One might think that this could pose problems for the navigational task. Some points concerning this issue are discussed below.

4.3.1 The Reactive Level

On the reactive level, the connection between the physical object and the symbols is not maintained over time. This brings about that the symbol is not connected to the same physical object at all times. For example, a cupboard standing in a corridor is detected as a narrower corridor (see Fig. 4.11), or the location of obstacles may be different from different robot perspectives (see Fig. 4.12). These inconsistencies, however, do not matter for the behavioural tasks. Following the supposedly narrower corridor is as good as having a percept of the real corridor. Avoiding supposedly different obstacles does as successfully avoid collisions as having the knowledge that they were perceived differently before. In general, to be able to ignore such inconsistencies, as it is possible here, the symbols and their detection processes have to be chosen carefully. Doing this, one must have in mind the type of task the symbol is later actually used for.

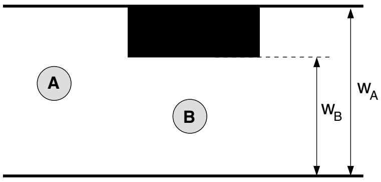

Figure 4.11. The robot at two different locations in the same corridor. At location A the system determines the width of the corridor as w_A. A narrower corridor (w_B) is detected from location B.

4.3.2 The Deliberative Level

For the symbol POSE on the deliberative level, the properties (location and orientation) are not determined precisely. However, the error is small enough to determine

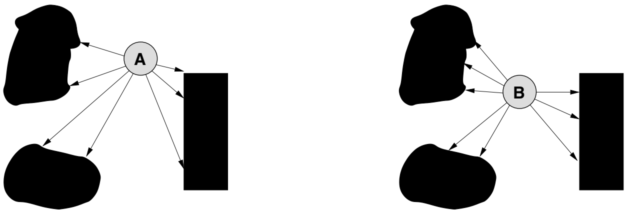

Figure 4.12. The robot at two different locations among the same obstacles. Each arrow visualises the direction and distance to an extracted obstacle. At location A the system detects 7 obstacles. At location B, however, the same physical objects give rise to a perception of 8 obstacles. Also the properties (direction and distance) of the symbols are very different in the two situations.

the necessary properties of the symbol TOPOLOGICAL CONTEXT. This information enables the coordination scheme to activate the appropriate behaviours, which in turn take care of the small scale structure of the environment using the precise properties of symbols on the reactive level.

By repeating the experiment discussed in section 4.2.2, it turned out that the error on the robots position estimate was always within 10–30 cm. This suggest a relative error of about 0.5%. This implies that corridors of up to 200 m in length could be traversed and still keep the error within 1 m. Experiments confirmed that this accuracy of 1 meter in the position estimate is enough to find doors and be able to traverse them. Further, each time a door has been traversed the position can be updated along the corridor direction.

4.4 Discussion

The navigation system developed in the previous two chapters has been tested extensively in the environment of our institute at KTH. A few example trajectories are presented in this chapter. It could be seen that the system acted according to the mathematical design of the behaviours and their coordination described in section 2.3. In particular, the bifurcations, which lead to qualitatively different behaviour, occured as predicted. This facilitated the robot with flexible decision making in order to react to unforeseen events like blocked corridors or miss-detected doors. Not only the control scheme was tested, but also the operation of the framework for symbol anchoring was verified. Although the symbols used on the reactive level were not maintained over time, the behaviours were able to guide the robot accurately. Further, it was shown that the disregard of extreme accuracy of the robot's position estimate is justifiable. The system was still able to recognise locations which correspond to nodes in the topological map. In addition, the system has been implemented on another robot in a different indoor setting. It displayed the same performance without changing any of its parameters. This clearly supports the generality of the developed system.

Another remark concerns the computation time used by the system. One cycle through the main loop of the program (symbol anchoring and control) takes about 3 milliseconds on the on-board 233 MHz Pentium running Linux. The Hough transform takes about 200 milliseconds, but it is only invoked every five seconds. This control rate of about 300 Hz is actually much faster than the daemon process is able to receive and send signals to the hardware via a serial port. This low CPU consumption is due to relatively simple anchoring processes that do not have to solve all the three problems outlined in the introduction. Instead, each anchoring process focuses on the problems necessary for its specific level. This characteristic is a major advantage compared to other indoor navigation systems, which often struggle due to computationally expensive algorithms. For example, systems using grid maps (Moravec, 1988), Markov decision processes (Koenig and Simmons, 1998), or Extended Kalman Filters (Leonard and Durrant-Whyte, 1991) run into this problem as soon as the number of symbols (grid cells, states, landmarks) increases.

The system should definitely be tested in more environments to establish it as a general approach to indoor navigation. Some special situations could potentially pose problems. For example, when two doors are located right next to each other, the robot might choose the wrong one and subsequently get lost. However, problems of this kind can probably be attacked straight forward by using more advanced sensors. For example, the use of a laser scanner could highly increase the quality of the geometrical representations. Another aspect of indoor navigation, which has been neglected in this work, would also need more accurate sensing capabilities in order to be attacked: the problem of localising the robot, when it is completely lost (Jensfelt and Kristensen, 2001). Here, it is always assumed that the estimate of the platform's pose is approximately correct. However, also this can fail for a number of reasons. Odometry data could be corrupted, or slippage of the wheels could be unexpectedly high, for example when driving over a high threshold at a door. A similar issue is the "kidnapped robot" problem, where the platform is lifted and placed somewhere else in the environment. For relocalising the system after such a failure, this failure first has to be detected and second the nodes in the map must be recognised by other means than odometric data. Kuipers et al. (1993) propose a methodology for identification of distinctive places in the context of robot learning. Such a feature of detecting locations, which correspond to nodes in the topological map, was also implemented by Ulrich and Nourbaksh (2000). This capability would increase the potential of the navigation system presented here dramatically.

Chapter 5

Automatic Map Acquisition

In the previous three chapters, the development of a navigation system for indoor environments has been presented and results have been discussed. As most systems for mobile robots, it makes use of a predefined world model in the form of a map of the environment. An interesting question is, whether this map can be acquired automatically by the robot. A semi-autonomous approach to this problem is presented in this chapter. The platform is following a person through the area and receives input about the semantics of the environment from this guide. During such a trial, the system builds a topological map online, which can later be used for navigation. This capability has clearly the advantage that a representation of the environment can be created without a human measuring the whole area. This additional feature has also been integrated into the existing navigation system.

First in section 5.1 the background and actual research in automatic map acquisition is introduced. Section 5.2 presents the behaviour for following a person and the methods for acquiring the map. Results are displayed in section 5.3 and the chapter concludes with a discussion in section 5.4.

5.1 Introduction

Map building is a major research topic in robotics. Naturally, the strategies and algorithms employed depend heavily on the type of the actual map that is constructed. In particular for indoor environments, a lot of research has been devoted to this topic. A detailed description of all methods and their implications would go beyond the scope of this thesis. The major approaches are shortly introduced below.

Metric maps are usually in the form of an occupancy grid or a collection of features as introduced in section 3.1.2. These maps rely heavily on the perception of the robot, which means the sensors and the methods to extract features. Thus, automatic map building by the robotic system is, often, the only way to acquire such a representation of the environment. Hence, not only the positions of map features have to be determined, but also the robot's position has to be estimated in order to relate the sensor measurements to this map. This research topic is commonly known as SLAM; simultaneous localisation and mapping (Castellanos and Tardos, 1999). Different frameworks are used for building and updating a grid map. The most common one is the use of Bayesian probability theory (Moravec, 1988; Elfes, 1987), but also other methods like fuzzy logic have been deployed (Fabrizi and Saffiotti, 2002). SLAM for feature maps is, in most cases, based on some sort of extended Kalman filter (Leonard and Durrant-Whyte, 1991). All these methods are computationally very expensive. Hence, the robot is, commonly, driven through the environment via a joystick collecting the data. Afterwards, the map is computed offline. However, metric maps can also be built by hand, which means detailed measuring of the whole setting. This can range from determining the position of walls with measuring tape (Jensfelt and Christensen, 2001) to defining a large amount of feature points with a sonar (Wijk and Christensen, 2000).

Topological maps are a more compact representation of the environment and, therefore, contain little information. Thus, these types of maps are, usually, built by hand. The map used in this work (see section 3.1.2), for example, was constructed by, first, choosing the locations of the nodes and, then, measuring their positions in a world-fixed coordinate system. In addition, there are some approaches to acquire such a map in the context of robot learning (Kuipers and Byun, 1991).

5.2 Implementation

A semi-autonomous approach to map acquisition has been developed. Instead of driving the platform around with a joystick, the robot is autonomously following a person through the area to be mapped. To achieve this functionality an additional behaviour, PERSON FOLLOWING, has been designed. Since sonar sensors have a rather bad resolution, a SICK laser range scanner has been mounted on the platform (Figure 5.1). This scanner has an angular range of 180° and a resolution of 0.5°. It provides the data from which the direction and distance to the person is extracted. This guide informs the system about the connectivity of the environment via a wireless interface. Using this information and sensory data from sonars the robot is creating the map.

This section is divided into two parts. The person following capability is introduced in section 5.2.1. It has been developed with the same methodologies as the navigation system of the previous chapters. Then, in section 5.2.2 the details on the map building process are presented.

5.2.1 Person Following

An additional behaviour, PERSON FOLLOWING, has been defined in order to follow the guiding person through the area to be mapped. This behaviour is fully

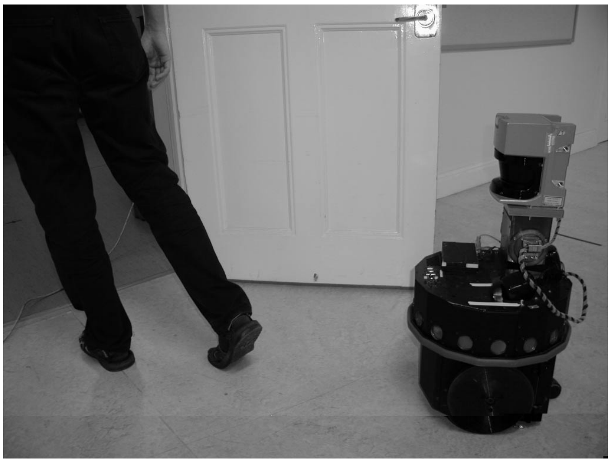

Figure 5.1. The robot is following a person around the institute.

integrated into the navigation system presented in the previous chapters. Hence, it is designed using the dynamical systems approach (section 2.2). The person to be followed is extracted from the laser data deploying the framework for symbol anchoring as proposed in section 3.2.2. Following the general structure of this thesis, the issues on control and representations for this additional feature are treated separately.

Control

As for the design of all the behaviours in section 2.3, a dynamical system has to be defined for each behavioural variable: the heading direction ϕ and the speed v.

The behaviour PERSON FOLLOWING is supposed to align the robot's heading direction with the direction ψ_{guide} of the person guiding the robot through the environment. This is, basically, the same constraint as for the behaviour GO TO. Thus the same functional form for the dynamics of heading direction, ϕ, as in equation 2.9 (Figure 2.2) is used. The constant $\lambda_{follow,\phi} > 0$ defines the strength of the attractor at ψ_{guide}.

$$\dot{\phi} = f_{follow,\phi}(\phi) = -\lambda_{follow,\phi} \sin(\phi - \psi_{guide}) \tag{5.1}$$

The robot has to keep a certain distance to the person during the mapping process. When the guide is walking slowly (around obstacles, for example), this distance should be rather short such that the robot is able to follow. In a corridor, however, where the person can move quicker, this clearance should increase for security reasons. Thus, the desired distance to the guide, d_{guide}, can be expressed as follows.

$$d_{guide} = D_{stop} + \frac{1}{k_{follow}} \cdot v_{guide} \tag{5.2}$$

v_{guide} is the speed of the guide, and D_{stop} defines the distance between the person and the platform when they do not move. The constant $k_{follow} > 0$ determines the increase in distance with increasing speed. Ideally, the robot moves at the same speed as the person. Thus, if we solve equation 5.2 for v_{guide} the desired speed of the platform v_{follow} becomes

$$v_{follow} = k_{follow}(d_{guide} - D_{stop}) \tag{5.3}$$

This form also makes the robot back off, if the person is coming closer than D_{stop}. In addition, choosing the speed as above assures avoidance of obstacles. This will become clear below, where the processing of the laser data is explained. As in the case for the behaviour GO TO (equation 2.22 and Figure 2.10), a linear dynamics for the speed, v, is defined creating an attractor at v_{follow} with strength $\lambda_{follow,v}$.

$$\dot{v} = f_{follow,v}(v) = -\lambda_{follow,v}(v - v_{follow}) \tag{5.4}$$

For the behavior defined here, it has also to be assured that the system stays close to an attractor state at all times. The same stability considerations as in section 2.3.2 can be applied to the dynamical systems defined here. Hence, the following conditions must hold.

$$\lambda_{follow,\phi} \gg k_{follow} \qquad \text{and} \qquad \lambda_{follow,v} \gg k_{follow} \tag{5.5}$$

Remember from section 2.3.3 that the competitive advantage (α_{follow}), the competitive interactions ($\gamma_{follow,b}$ and $\gamma_{b,follow}$), and the timescale (τ_{follow}) have to be defined for behaviour coordination. This is straight forward, since the system does either navigation as presented in the previous chapters or it does mapping. In other words, during mapping the behaviour PERSON FOLLOWING is switched on ($\alpha_{follow} = 0.5$) and all other behaviours are switched off ($\alpha_b = -0.5$). Since there is only one behaviour active, no competition has to be achieved ($\gamma_{follow,b} = \gamma_{b,follow} = 0$). Further, the precise value of the timescale τ_{follow} does not really matter, because the behaviour remains on during the whole mapping process.

Representations

The symbol GUIDE was defined, which is a representation of the person that the platform is following. Its properties are used by the behaviour PERSON FOLLOWING on the reactive level. Remember the general constraints on the symbols on this level of control from section 3.2.2. The exact properties are important, because accurate navigation must be achieved by the behaviour. Maintaining the properties of the symbol over time is neglected here, since the behaviour acts in a reactive manner to sensory events.

Guide is extracted from sensor readings provided by the laser scanner. The echos lying in the frontal 90° of the robot are considered. In general, the shortest one of these readings comes from the person that the robot is following. However, when passing narrow passages such as doorways, the obstacles on the side might be closer to the platform than the person. Also in empty corridors, when the distance to this person is increasing (due to a higher speed), the corridor walls are closer to the robot. To account for this, the sensor reading representing the person to follow has been determined as the one minimising the following expression.

$$d_i \cdot (1 + |\sin(\psi_i)|) \tag{5.6}$$

d_i is the distance returned of laser beam i, and ψ_i is the direction of this beam relative to the robot's orientation. In this way, distances are weighted with their deviation from the robot's orientation. Of course, this is a very crude approach to detecting a person. Nevertheless, it turned out to serve its purpose very well. Furthermore, it also facilitates obstacle avoidance by the person following behaviour. If an obstruction is, for some reason, very close to the platform, the system is backing off in a way defined in equation 5.4. After the laser beam representing the person is determined, its direction parameterises the dynamics of heading direction $f_{follow,\phi}$ (equation 5.1). The returned distance of this beam is used by the dynamics of speed $f_{follow,v}$ (equation 5.4).

The symbols corridor and obstacle are used for the map acquisition process (see next section). They are extracted in the same way as described in section 3.3.1 with one exception. The frontal five sonar sensors (covering 90°) are ignored, because they, usually, return readings from the guide. Note that the echos from the laser scanner are only used for following a person. The only sensory data supporting the whole map acquisition process are from the sonars.

5.2.2 Acquisition of the Map

Before presenting the actual acquisition method, the important properties of the map are quickly reviewed. Remember from section 3.1.2 that the nodes of the topological map have an x- and a y-coordinate in a world-fixed coordinate system. The edges can be of three different types: "corridor", "room", or "door". Thus, nodes have to be placed at every position of the environment where the edge type would change, which means in front of every door. Further, places where the graph splits up (for example corridor crossings) have to be represented by a node. In addition, all the places in the environment that are important for the specific tasks carried out (charging station and goal points) need to be reflected in the map by a node. In order to underline the working of the method presented here for large-scale areas, the environment has been extended to the corridors outside our institute. Figure 5.2 depicts the topological map of this setting (about 60×70 meters). This map has been acquired manually by measuring actual distances between nodes. The x- and y-coordinates of these nodes have been determined, precisely as illustrated in Figure 3.2. In other words, nodes are centered in corridors and an edge of type "door" goes through the middle of the door posts. The map shows all parts of the premises that are actually accessible by the robot, which means no rooms with high thresholds at the doorway. Depending on the tasks, an arbitrary amount of nodes can be added to the map to include, for example, a mail slot where something has to be picked up, office desks where it has to be delivered to, or any other points of interest.

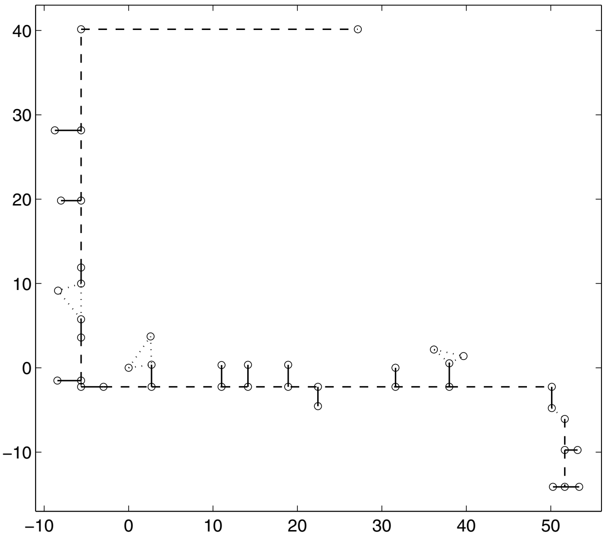

Figure 5.2. The topological map of our institute. The circles depict nodes with a position in a fixed coordinate system (units on the axes are in meters). Edges are of three different types: corridor (dashed lines), room (dotted lines), and door (solid lines). The charging station is depicted by the node at the origin. Goal points and other locations of importance for the navigational task can be added arbitrarily in the rooms and the corridors.

The map is acquired in that a person is showing the robot around. The platform autonomously follows the guide through the environment (Figure 5.1). This person carries a laptop to interact with the robot through wireless ethernet. Via this interface, the system is updated about topological changes during their tour. The guide informs the program every time when they leave a room or a corridor, and then again, when a new room or corridor is entered. Also corridor crossings and other points of interest (for example goal points) are messaged to the system. During such a trial the robot has an estimate of its position and orientation using similar principles as during navigation (see section 3.3.2).

The map acquisition is happening online and begins at the starting location. First, the system places a node at its initial position. The meaning of the first edge ("corridor", "room", or "door") is entered by the guide. The robot keeps track of its position and orientation through the odometric data. When the system is informed about a new node (for example when leaving a room), the robot's position coincides probably not exactly with the nodes correct position as introduced in Figure 3.2. Hence, corridors and doorways extracted from the sonar data are used to position the nodes. This is illustrated in Figure 5.3, where a possible trajectory of the platform is shown (compare to Figure 3.2).

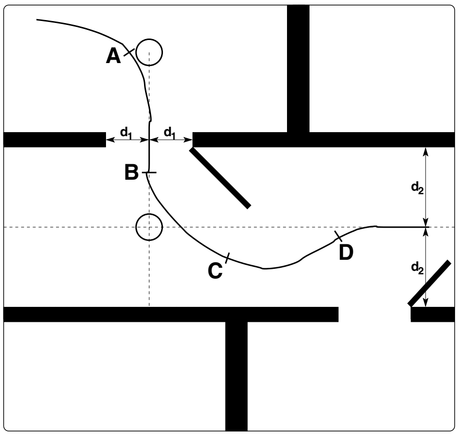

Figure 5.3. Correct placement of nodes during the map acquisition process. The black line shows a possible robot trajectory. Nodes are depicted by circles. Situations A-D are explained in the text.

A: The system is informed about leaving the room. While passing the door the narrowest passage (doorway) is extracted from the sonar data. Its position and the orientation of the door posts are saved.

B: The system is informed about having entered a corridor. The narrowest passage since A is remembered and the upper node can be placed on the centerline of the door. The robot continues following the guide along the corridor.

C: The corridor properties (orientation and distance to the walls) are extracted from the sonar data (see section 3.3.1).

D: The corridor properties are extracted again and averaged with the first guess. This gives a good enough estimate of the actual corridor. Thus the second node can be placed, centred in the corridor and aligned with the first node. Now the direction of the corridor is fixed. Hence, while the trial continues the robot's position can be updated each time the Hough transform is invoked, as described in section 3.3.2.

This whole process happens each time the platform moves from a room to a corridor and, similarly, the other way round. Further, nodes of goal points are placed at the location of the robot's position estimate at the time the system is informed about placing a node. Each time the guide tells the robot about a new node, it also enters the type of the next edge via the terminal interface.

5.3 Results

Figures 5.4 and 5.5 show two topological maps acquired by the system. These are the result of two particular trials. All experiments result in their own maps, which do not look exactly the same. This is, in general, due to odometry drift, disturbances at door thresholds, and unprecise corridor estimates. However, their difference is in the same order as the discrepancy between the two examples presented here. Hence, we focus the discussion on comparing these two with the correctly measured map in Figure 5.2. In general, experiments have shown that the maps acquired by the robot serve equally well for navigating between two arbitrary nodes as the precise one built by hand.

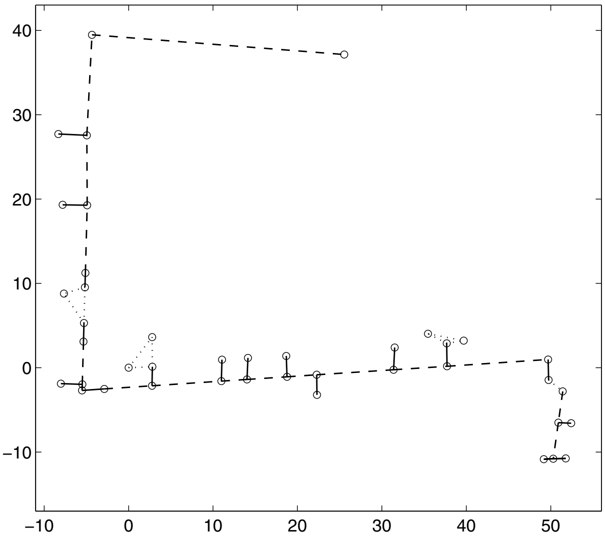

Figure 5.4. A map of our institute acquired by the robot (units are in meters): The circles depict nodes. Edges are of three different types: corridor (dashed lines), room (dotted lines), and door (solid lines). The docking station is depicted by the node at the origin.

The most obvious difference between the two maps is that the angles between edges are not the same. This is particularly well visible at the long corridors.

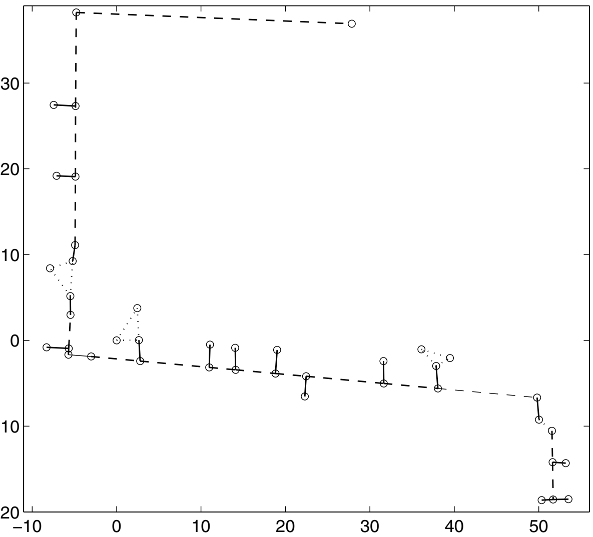

Figure 5.5. A map of our institute acquired by the robot (units are in meters): The circles depict nodes. Edges are of three different types: corridor (dashed lines), room (dotted lines), and door (solid lines). The docking station is depicted by the node at the origin.

However, this does not matter, when the map is used later. Remember that the navigational tasks are split up into subtasks that consist only of navigating from one node to a neighbouring one. At the nodes themselves, task switches are invoked and the behaviours take care of the small scale structure of the environment. For example, when the robot has to enter a room from the corridor, the behaviour "door passing" guides the platform through the doorway, where the robot's orientation relative to the map is corrected again. Hence, the tasks do not fail as long as corridors and doors can not be mixed up. However, this would only occur when angular errors were in the order of 45°.

Another discrepancy between the maps is the length of the door edges. This is hard to see in the figures, but they actually vary up to 1 meter. These differences, however, are not errors, since door nodes in rooms can be placed anywhere on the centerline of the door (see section 3.1.2). This placement does not affect the executions of the navigational tasks.

One aspect of the map that is crucial for the navigation system to succeed is the length of the corridor edges, in order to find the correct door. Comparing these values of the acquired maps with the correct one suggests that the relative error is in the order of at most 0.5%. Remember from the evaluation of the navigation system that the error of the position estimate during navigation is in the same order (section 4.3.2). It can be assumed that the system is able to find the correct door, if its position relative to the map is less than 1 meter. This suggests that the approach presented here could successfully handle corridors up to a length of 100 meters.

Another aspect that the navigation system has to rely on is the length of the edges inside rooms and the angles between them. The errors in length are equally small as in corridors. However, the errors in angle can be of the same magnitude as between individual corridors, because no features of rooms are used to update the estimates of the robot's orientation. This could lead to problems in two cases: 1) if the rooms are very large, and 2) if the robot navigates inside the same room for a long time. Experiments show that rooms of the size of our offices do not pose any problems for navigation to a few goal points. Furthermore, if a task demands frequent navigation between nodes in the same room (for example, repetitive fetch-and-carry between two tables), additional means to update the robot's orientation are, probably, provided (for example extracting the tables from sensory data). However, additional studies would need to be carried out to verify this believe.

The above mentioned simplification, of generally not using any features in the rooms, provides an advantage in modelling the whole environment. Since no assumptions on the structure of a room are made, everything that does not look like a corridor or a doorway can be modelled as a room. However, this does not mean that cluttered corridors need to be defined as rooms. Figure 5.6 shows an example of such a corridor that did not pose any problems to the behaviour CORRIDOR FOLLOWING to safely navigate. Neither was it a problem to successfully update the robot's orientation estimate in both navigation and map construction.

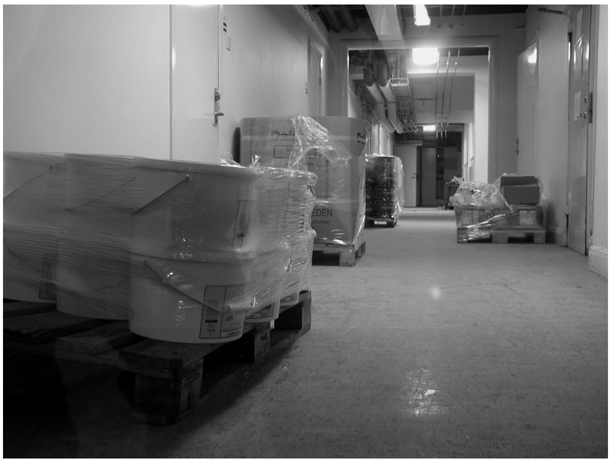

Figure 5.6. A cluttered corridor in our institute. These types of obstructions do not pose any problems for extracting the corridor features during navigation and map building.

5.4 Discussion

A rather simple mechanism for map construction was presented. A new behaviour PERSON FOLLOWING has been defined, which enables a robot to follow a person through an environment and simultaneously building a map. Experiments have shown that this map can successfully be applied to navigate through our whole institute. All the mechanisms have been integrated into one complete navigation system for large-scale domestic environments.

To emphasize the simplicity and robustness of the approach, the same feature extraction mechanisms as for navigation (section 3.3.1) have been used, which rely only on data provided by sonar sensors. Further, the whole environment is modelled as a set of corridors, rooms, and doorways. This simplification allows to acquire a representation of the entire environment with rather basic mechanisms and low CPU consumption. The latter permits to build the map online without any post-processing. This is clearly an advantage to SLAM approaches deploying Kalman Filters and Bayesian theory together with a complex world model. Recent work shows that also these methods can be applied in real-time; see (Guivant and Nebot, 2001) and (Thrun et al., 2000) for two excellent examples. However, in these examples sophisticated algorithms had to be developed to be able to restrain CPU and memory consumption. Here, it has been shown that a map that is only used for navigation can be acquired in a much easier fashion.

The person detection, used for following the guide through the environment, is a rather crude approach. Sometimes, the system looses track of the person, and the guide has to reposition itself in front of the platform in order to be detected again. The implementation of a more advanced tracking algorithm (for example, (Schulz et al., 2001)) would definitely improve this part of the system. In addition, the interaction of the guiding person with the robot is not particularly user-friendly. The little information given by the guide (presence of a node and type of the edge) could as well be communicated via a speech interface. This enhancement would make the system much easier to use.

Chapter 6

Human-Robot Interaction

The research presented in this chapter has been conducted during a four months stay at the Advanced Telecommunications Research Institute (ATR) in Kyoto, Japan. The Intelligent Robotics and Communication Laboratories at ATR focus on robotics research in the context of *human-robot interaction*. Navigation of a mobile platform interacting with people is driven by very different constraints than it is for functional tasks like driving from a point A to a point B. An introduction to this type of research and the navigational constraints are given in section 6.1. Then, in section 6.2, the robotic platform and the interaction scenario tested in these studies are presented. In addition, some results from psychology are considered, which pose constraints on the robotic system for the interaction task. The design and implementation of the actual system is described in section 6.3 and experimental results are presented in section 6.4. Finally, a summary and a discussion of avenues of future research are provided in section 6.5.

6.1 Introduction

In service robotics, which is the research context for the other chapters of this thesis, human-robot interaction is primarily for task specification. Subsequently, the mission the robot has to fulfill is defined. Hence, its behaviour is task driven. This is the main difference to the research approach that is pursued at ATR, which is probably best described by *personal robotics* (Ishiguro et al., 2003). The goal is to develop a socially interactive robot (Fong et al., 2003) that “lives” in people’s homes and coexists with humans in their daily live. Hence, the behaviour of the robot is very much interaction driven. The actions of the robot are dependent on the presence of humans and on their willingness to interact. Further, the type and degree of interaction determines the subsequent behaviour. Also outcomes of a previous encounter can influence this interplay. For example, when a robot greets somebody, the response of this person determines the subsequent actions: shaking hands, engaging in a conversation, following the person to another room, and so forth. If this particular person is met again later, the interaction might be based on the first meeting. For example, the same questions should not be asked again.

For all these types of tasks, navigation of the platform is a basic problem that has to be attacked. However, the constraints on the navigational tasks are in some ways very different than for a fetch-and-carry type mission. Following the general structure of this thesis, some considerations are outlined below in the contexts of control and representations.

The major issue concerning the control is that the robot's behaviour has to look "natural". The movements have to be smooth and "human-like", since a personal robot is supposed to live, communicate and interact with people sharing the same environment. The quality of its movements influences strongly the perceived intelligence of the robotic system. As opposed to a delivery robot, which has to reach a certain destination with few constraints on the motion pattern. There, the only requirements are that the platform gets to the goal safely and in reasonable time. We believe that the dynamical systems approach is especially well suited for navigation in the context of human-robot interaction. Through its smooth switching between subtasks a continuous control signal can be assured at all times. Further, the speed of the platform is an important factor influencing humans' reactions to the robot's movements. In this thesis, it has been shown how the dynamical systems approach can be extended to the control of speed; unlike other approaches, which often only incorporate heading direction and control the speed by some external module.

The representations of the environment for human-robot interaction are different than the ones for navigating from A to B in two ways. On the one hand, the detection of geometrical features for local navigation are more complex. For example, the system has to be able to distinguish between humans and other objects like tables and chairs. In the earlier chapters of this thesis they were both considered as obstacles. On the other hand, the context information, which enables behaviour switching, is of a different nature. In delivery missions, it has mainly the form of locations in the environment as described in section 3.3.2. Here, detected events, which define the state of interaction including its history, determine which behaviour to activate next. These differences in representations require, typically, more complex sensing abilities. In addition, a richer world model than a map of the environment is needed. For this purpose, the topological map introduced in section 3.1.2 has been extended to a state map, which incorporates both geometrical locations and interaction events. Further, the framework for anchoring proposed in this thesis (section 3.2.2) is applied here as well. The symbols can still be separated for the two levels of control, although their properties are somewhat different.

This chapter presents some initial studies and experiments on robot navigation explicitly for human-robot interaction. The emphasis lies not on the complexity of the task as such. Rather on verifying of the following points: 1) Is the dynamical systems approach suited for navigation in interaction tasks? 2) How can the topological map be combined with interaction relevant information? 3) Does the framework for anchoring proposed in this thesis still hold?

6.2 The Experimental Setup

An experimental setup has been defined in order to show the applicability of the methods developed in this thesis to navigation for human-robot interaction tasks. First, in section 6.2.1, the robotic platform Robovie is described. An actual scenario, relevant for a robot sharing an environment with humans, is described in section 6.2.2. In such a scenario, task constraints are usually given by psychological studies on human behaviour. These considerations are presented in section 6.2.3. Finally, the actual implementation is presented later in section 6.3.

6.2.1 The Platform

The platform used in these studies is a humanoid system on wheels called Robovie (see Figure 6.1). The robot has a height of 120 cm and its weight is about 40 kg. The system is built on a Pioneer platform of 45 cm diameter with a two wheel differential drive. The robot has two arms with 4 degrees of freedom each. The head can be rotated around all three axis. Mounted on the head are two cameras that have also 2 degrees of freedom. There are various sensors mounted on the platform. The base has 16 ultrasonic sensors and encoders on the wheel for odometric information. Another 8 sonars are mounted on the upper body. Further, the robot is equipped with 3 IR motion sensors on its shoulders, the two cameras on the head, and an omnidirectional camera on a sort of antenna. A number of touch sensors are distributed all over the body. In addition, two microphones and a loudspeaker provide a speech interface. The whole system is controlled by an on-board computer with a Pentium III processor running Linux.

Robovie has been used in various studies on human-robot interaction (Ishiguro et al., 2003). The system has modules for detecting moving objects and tracking faces. It is able to perform a number of gestures like greeting, waving, pointing, or hugging. Through moving its head it can give a realistic impression of attention. The speech interface allows some basic communication.

6.2.2 The Scenario

A robot sharing its environment with humans as an equal being has to be able to engage in conversations. In order to do so, the system needs several abilities, which can be tested with the following scenario:

1. The robot enters a room and looks around for people.
2. If any humans are present, the platform approaches them.
3. The robot stands next to the person(s) and initiates or joins a discussion.
4. When the discussion is over or the robot is asked to do something else, it leaves the person(s) towards the exit of the room.

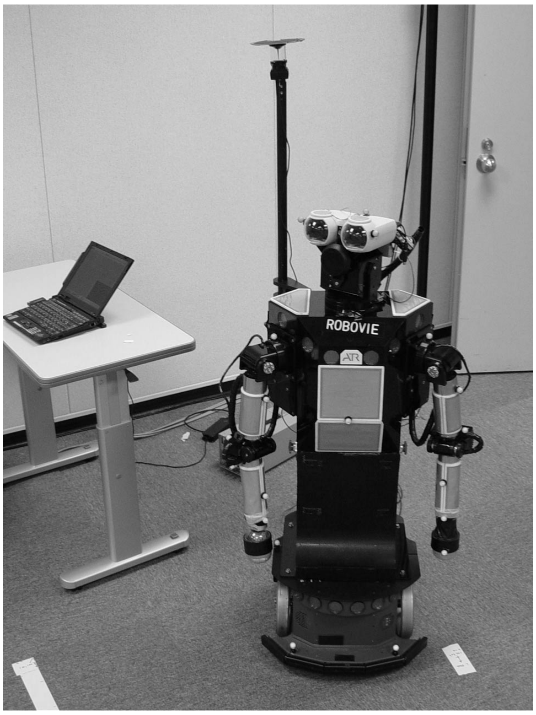

Figure 6.1. The robotic system Robovie.

5. The system resumes some old plan, responds to some new request, or wanders around in its environment.

Note that the emphasis of these studies is navigation. In the actual experiments (see section 6.4) the robot is in fact not really communicating. Keeping up a conversation over a reasonable time is very hard and a research field in itself. In other words, the platform is joining the group of people as a passive listener. This scenario is a minimal approach to demonstrate navigation in human-robot interaction including the following important issues:

- Detecting the presence of people.
- Moving in an appropriate fashion in relation to humans.
- Sequencing of different subtasks.
- Keeping track of position and plan execution.

Before the actual implementation of this scenario is presented, some issues on navigation in relation to humans are considered below.

6.2.3 Psychological Considerations

The design of autonomous systems for human-robot interaction is highly influenced by psychological studies. There exists a great amount of results from this area describing the way how humans interact with each other. It is the aim for a personal robot to follow the same principles when acting in a human environment. This enables the acceptance of a robotic platform as an equal being on the one hand. On the other hand, it increases the perceived intelligence of the system (Kanda, Ishiguro, Imai, Ono and Mase, 2002). Further, psychological methods in the form of questioning human subjects about their impression of a robot, are a valuable method to evaluate and measure the quality of an interacting system (Kanda et al., 2001; Kanda, Ishiguro, Ono, Imai and Nakatsu, 2002).

This chapter deals with the navigation aspects of an interaction task. Hence, studies about people's overall movements and their position relative to each other are taken into account. Kendon (1990) presents results on extensive studies on this topic. The main points being that certain formations among people are maintained, while they are interacting. Although individual people often are moving due to different reasons, the other persons are compensating for this by also moving in order to maintain the formation. In a group conversation, this formation usually has the form of a ring, where all persons keep equal distance to their neighbours. When only two people are engaged in a discussion, they keep a constant distance between themselves. Of course, there are many variations and exceptions of these general rules depending on the relation among the persons and the state of the conversation. However, the main principle described above is rather dominant and will define the constraints for the design of the control system.

6.3 Implementation

The implementation of the scenario described above is described in this section. Following the general structure of this thesis, the issues on control and representations are discussed separately.

6.3.1 Control

The control scheme is designed in the framework of the dynamical systems approach introduced in section 2.2. To provide the functionality of joining a group of people and engaging in a conversation (as outlined in section 6.2.2) two additional behaviours were defined.

- APPROACH HUMAN: driving towards a person or group of persons after they have been detected. When getting close to the people the platform should slow down in a smooth manner.

- KEEP DISTANCE: positioning the robot, such that it is part of a ring formation with the other persons. In other words it has to face the centre of the group, while keeping a constant distance to its neighbours. Alternatively, if only one human is present, the platform should face this person and maintain a constant distance.

These behaviours are integrated into the existing control system presented in section 2.3. This design in one framework allows switching between fetch-and-carry type navigation and interaction type navigation at all times. Hence, the same behavioural variables are chosen (equation 2.7): the heading direction ϕ and the speed v. Also here the dynamics of the two variables will not depend on each other. Thus, the notation defined in equation 2.8 will be used.

Dynamics of Heading Direction

The behaviour APPROACH HUMAN is supposed to align the robot's heading direction with the direction ψ_{human} of the detected person. This is, basically, the same constraint as for the behaviour GO TO. Thus the same functional form as in equation 2.9 (Figure 2.2) is used. The constant $\lambda_{human,\phi} > 0$ defines the strength of the attractor at ψ_{human}.

$$\dot{\phi} = f_{human,\phi}(\phi) = -\lambda_{human,\phi} \sin(\phi - \psi_{human}) \tag{6.1}$$

In the case of one person j present the behaviour KEEP DISTANCE has to turn the robot towards it. This can be achieved by creating an attractor at ψ_j, the direction of this person.

$$f_{j,\phi}(\phi) = -\lambda_{dist,\phi}(\phi - \psi_j)\, e^{-c_{dist1}(d_j - D_{inter})} \tag{6.2}$$

d_j is the distance to the person, D_{inter} defines the desired interacting distance between this person and the platform, the constant $c_{dist1} > 0$ is a decay constant, and $\lambda_{dist,\phi}$ determines the strength of the attractor. In case of multiple persons, the resulting dynamics is computed by adding the individual contributions.

$$\dot{\phi} = f_{dist,\phi}(\phi) = \sum_j f_{j,\phi}(\phi) \tag{6.3}$$

Comparing equation 6.2 to the usual attractor dynamics used in this thesis (for example 6.1) shows two main differences: a linear function instead of a sine and a distance decay term. The reason for the linear function is averaging. In other words, if several people are surrounding the robot at equal distance, the dynamics of equation 6.3 creates an attractor at the average direction. This ensures that the robot faces the centre of a formation as described in section 6.2.3. However, since this function is not 2π-periodic, the angles have to be defined in such a way that $|\phi - \psi_j| \leq \pi$. The distance dependent decay term has the purpose of "repairing" a formation. If one person in the group steps closer to the robot ($d_j < D_{inter}$), its contribution becomes stronger than the others as shown in Figure 6.2. This ensures that the platform is turning towards this person, while backing off. Subsequently, when the distance increases again, it is aligning itself with the center of the formation.

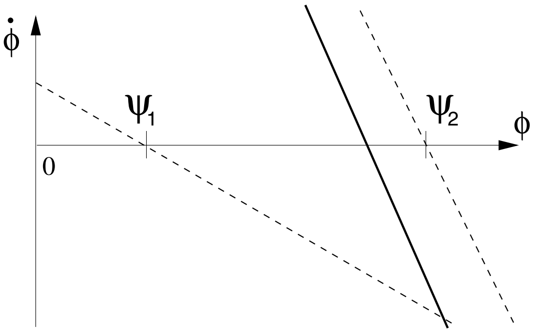

Figure 6.2. The dynamics of heading direction ϕ for KEEP DISTANCE in the case of two persons present (dashed curves). If person 2 steps closer to the robot, its contribution becomes stronger. Thus, the overall dynamics (solid curve) has an attractor close to the direction of that person.

Dynamics of Speed

The desired speed for APPROACH HUMAN is dependent on the distance d_{human} to the people that have to be approached. In the same way as when a human is approaching a group of people, the platform should slow down when getting closer. Further, the speed ought to be bounded by some maximum value $v_{human,max}$ in order not to scare anybody. Hence, the desired speed v_{human} is defined as follows

$$v_{human} = \min(k_{human} d_{human}, v_{human,max}) \tag{6.4}$$

where $k_{human} > 0$ is a constant. As in the case for the behaviour GO TO (equation 2.22 and Figure 2.10), a linear dynamics is defined creating an attractor at v_{human} with strength $\lambda_{human,v}$.

$$\dot{v} = f_{human,v}(v) = -\lambda_{human,v}(v - v_{human}) \tag{6.5}$$

In the case of interacting with a single person, the behaviour KEEP DISTANCE has to make the platform back off when the person gets closer than the desired interaction distance D_{inter}. If the distance, however, becomes greater the robot should approach. In order to avoid abrupt changes in movement, these speeds are also regulated by the distance d_j to that person. Thus the desired speed can be expressed in the following form.

$$v_j = k_{dist}(d_j - D_{inter}) \tag{6.6}$$

with a constant $k_{dist} > 0$. Also for this behaviour the function f is chosen linear.

$$f_{j,v}(v) = -\lambda_{dist,v}(v - v_j)\, e^{-c_{dist2}(d_j - D_{inter})} \tag{6.7}$$

$\lambda_{dist,v}$ determines the strength of the attractor and c_{dist2} is a decay constant. The overall dynamics of speed for KEEP DISTANCE is the sum of the contributions for each individual person.

$$\dot{v} = f_{dist,v}(v) = \sum_j f_{j,v}(v) \tag{6.8}$$

The distance dependent term in equation 6.7 weighs the attractors of contributions from closeby people stronger than the ones from people further away. This always ensures that the platform is avoiding persons that are standing too close (Figure 6.3). For more details and illustration on the behaviour emerging from these dynamics see the results in section 6.4.

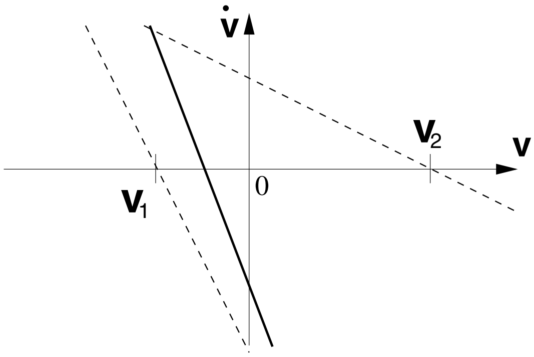

Figure 6.3. The dynamics of speed v for KEEP DISTANCE in the case of two persons present (dashed curves). Person 1 is too close to the robot, while person 2 is too far away. Since the attractor of the closer person (v_1) is stronger, the overall dynamics (solid curve) makes the platform drive backwards.

For the behaviors defined here, it has also to be assured that the system stays close to an attractor state at all times. The same stability considerations as in section 2.3.2 can be applied to the dynamical systems defined here. Hence, the following conditions must hold.

$$\lambda_{human,\phi} \gg k_{human} \quad \text{and} \quad \lambda_{human,v} \gg k_{human} \tag{6.9}$$

$$\lambda_{dist,\phi} \gg k_{dist} \quad \text{and} \quad \lambda_{dist,v} \gg k_{dist} \tag{6.10}$$

Behaviour Coordination

The two new behaviours presented above are integrated into the existing system of the five behaviours defined in section 2.3. Thus, the overall dynamics of the system is obtained from the weighted summation of all behaviours according to equation 2.4:

$$\begin{pmatrix} \dot{\phi} \\ \dot{v} \end{pmatrix} = \sum_{b \in B} |w_b| \begin{pmatrix} f_{b,\phi}(\phi) \\ f_{b,v}(v) \end{pmatrix} + \text{noise} \tag{6.11}$$

with $B = \{goto, obst, corr, wall, door, human, dist\}$. For the coordination of the behaviours, the competitive advantages α_b, the competitive interactions $\gamma_{b',b}$, and the time constants τ_b in the competitive dynamics of the weights w_b (equation 2.5) have to be chosen appropriately. Here, these choices are rather straight forward. Only sequencing of the subtasks has to be achieved. First, entering the room, then, approaching the group of people and keeping the formations, and finally, leaving the room again.

The competitive advantages α_b express the applicability of a behaviour b in the current context. Until the robot has entered the room, navigation is working in the same way as described earlier in this thesis. In other words, $\alpha_{human} = \alpha_{dist} = -0.5$. All other competitive advantages are set to exactly the same values as defined in section 2.3.3. Then, as soon as the humans in the room are detected, the competitive advantage for APPROACH HUMAN, α_{human}, is set to 0.5 and all other behaviours are switched off ($\alpha_b = -0.5$). When this subtask is completed, which means the group of persons are reached, KEEP DISTANCE is switched on by setting $\alpha_{dist} = 0.5$. Further, everything else is turned off ($\alpha_b = -0.5$). Finally, as the interaction is completed, α_{human} is set to -0.5 again. Now, the activity of all other behaviours is, again, determined by the navigation scheme as defined in 2.3.3. Hence, the robot is able to return to the door and fulfill some arbitrary navigation mission.

The competitive interactions $\gamma_{b',b}$ allow an active behaviour b' to suppress the activity of another behaviour b. The simple sequencing of subtasks required here, does not introduce any conflicts among behaviours. This means that all new competitive interactions ($\gamma_{human,b}$, $\gamma_{dist,b}$, $\gamma_{b',human}$ and $\gamma_{b',dist}$) are set to 0. All other $\gamma_{b',b}$ are, of course, set to the exact same values as in section 2.3.3.

Finally, the time constants τ_b determine the rate at which behaviours are turned on and off. Because, here, the behaviours are active one after the other, these values are of minor importance. The only constraint being, that they should not be too high ($\tau_{human} = \tau_{dist} \ll \tau_{obst}$), such that behaviour switches do not occur too abruptly. This ensures smooth control of the platform, which leads to "naturally" looking movements.

6.3.2 Representations

Geometrical representations are used to facilitate the interaction tasks. Basically, properties of the detected humans parameterise the behaviours introduced above. Further, context information is needed to allow switching between the individual navigation strategies. The symbols that represent these properties to allow decision making are introduced below. Following the framework proposed in section 3.2.2 they are, also here, treated separately for the two levels of abstraction.

The Reactive Level

Remember the general constraints on the symbols on this level of control from section 3.2.2. The exact properties are important here, because accurate navigation has to be assured by the behaviours. The platform has to be able to approach humans and maintain formations with them. Further, collisions have to be avoided while doing this. Maintaining the properties of the symbols over time is not important here, since the behaviours act in a reactive manner to sensory events.

The behaviour APPROACH HUMAN makes use of the symbol HUMAN. After the robot has entered the room, a person tracking module is initiated. This module looks for a person using a combination of data from different sensors: the omnidirectional camera, one of the cameras on the robot's head, the sonar sensors and infrared sensors mounted on the shoulders of the robot. The output of this module specifies the direction ψ_{human} of the human. This value serves as a parameter for $f_{human,\phi}$ (equation 6.1). With help of the sonar sensors the distance d_{human} to this person can be determined. This influences $f_{human,v}$ (equations 6.4 and 6.5), the dynamics of speed for the behaviour APPROACH HUMAN.

The behaviour KEEP DISTANCE uses of the properties of the symbol PERSON. The persons surrounding the robot, while interacting are extracted from the sonar data. This is done in a very similar fashion as for obstacles (section 3.3.1). The 30 most recent sonar readings are considered. Persons are reconstructed from these detected echos in ascending order of their distance to the robot. The echo closest to the robot defines the first person whose orientation in the robot frame is given by the axis of the sensor that received the echo. A new person is recorded for every subsequent echo whose orientation differs by an angle of at least 45° from any previously identified person. New persons are added in an incremental fashion until the sonar buffer contains no further echos. For each person j, the dynamics of heading direction $f_{j,\phi}$ (equation 6.2) uses the angle ψ_j and the distance d_j as parameters. Further, also the dynamics of speed $f_{j,v}$ (equation 6.7) is dependent on the distance to the person.

The Deliberative Level

On the deliberative level, decisions about which subtasks to initiate have to be made. In order to do this the system has to be able to keep track of the overall mission by using some kind of world model. For the tasks of moving from a point A to a point B, this world model is in the form of the topological map introduced in section 3.1.2. The nodes in this map correspond to locations in the environment, where a change in the navigational strategy has to occur. In interaction tasks, behaviour selection is not only governed by locations, but also by events in the current interaction of the robot with the humans. In other words, the world model has to be extended in such a way, that it can represent these events.

A state diagram has been defined preserving the graph structure of the topological map. Figure 6.4 shows this diagram for the scenario described in section 6.2.2.

Now, nodes reflect both locations in the environment and events in the interaction task. Since events can not occur in any arbitrary order, the edges have to be directed (as opposed to the topological map). These edges still are of a certain type associated with a subtask. In addition to "room", "corridor", and "door", they can be of types "approach" and "join" as well. These types determine the activation of behaviours. The edge of type "approach" corresponds to the activation of the behaviour APPROACH HUMAN. The edge of type "join" corresponds to the subtask of joining the group of people and maintaining the formation relative to them. In other words, the behaviour KEEP DISTANCE is active.

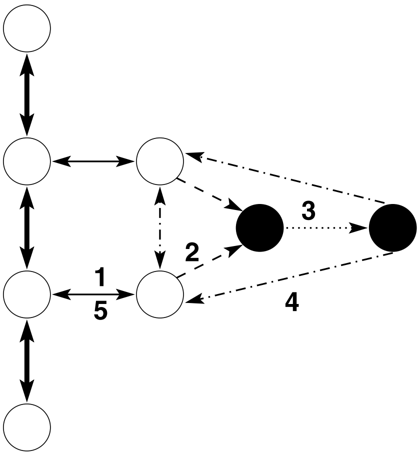

Figure 6.4. The state diagram representing a corridor on the left with two doors into a big room, where the interaction scenario described in section 6.2.2 takes place. White nodes depict locations and black nodes correspond to events. Edges can be of five different types: corridor (thick line), door (this line), room (dashed-dotted line), approach (dashed line), or join (dotted line). In the experiment described in section 6.4, the system follows the edges according to the numbers: 1) entering the room, 2) approaching the group of people, 3) maintaining certain formations with the persons, 4) getting back to the door, and 5) leaving the room again.

Note that, in this diagram, nodes correspond to events and edges to tasks in order to preserve the topological map structure. This is an obvious difference to finite state acceptor diagrams (Arkin and MacKenzie, 1994) and discrete event systems (Kosecká and Bajcsy, 1994), where the role of edges and nodes is reversed. Further, the diagram as depicted in Figure 6.4 is not a perfect solution yet. In a larger environment, the scenario of joining a group conversation could happen at many different places. Which means that the nodes and edges reflecting this interaction would have to be added to the map many times. A solution to this redundancy could be to define a third dimension for event nodes. Then, all location nodes could, in theory, have a connection to the same node in this additional dimension.

Remember from section 3.3.2, that the symbol TOPOLOGICAL CONTEXT has a property “location” which is used by the behaviour coordination system to activate the appropriate set of behaviours (section 2.3.3). The value of “location” is defined by the edge of the map that the platform is currently following. Also for the state diagram, the system has to keep track of the current edge. In order to do this, the abstract symbol SENSORY CONTEXT, introduced in section 3.3.2, provides three more properties: “human detected”, “person approached”, and “interaction over”. All of them having values of either true of false. “Human detected” is determined by the person detection module mentioned above. It’s value defines α_{human} and, if true, initiates the behaviour APPROACH HUMAN. The value of “person approached” is determined by d_{human} the distance to the group of people. If this value is below a certain threshold “person approached” becomes true, which influences α_{dist} and the behaviour KEEP DISTANCE is activated. In the present implementation, the end of the interaction is defined by a timeout after 2 minutes. However, it could easily be determined by some event of the interaction (for example speech or a certain gesture). This sets “interaction over” to true, which in turn switches off the interaction behaviours ($\alpha_{human} = \alpha_{dist} = -0.5$). The robot will switch back to pure navigation mode (see section 2.3) and continue its way through the environment according to its old plan.

Further the symbol POSE reflecting the robot’s position and orientation is constantly updated as described in section 3.3.2. When the interaction is over, this information is needed by the system in order to proceed to the next node in the map and continue its way through the area.

6.4 Results

The scenario described in section 6.2.2 has been tested. The robot entered a room, detected a group of three humans engaged in a conversation, and joined this group. While talking to each other, the persons were moving slightly. Further, one of these persons was leaving the group at some point. Later, this person was coming back and joining the group again. During all these movements of the people, the robot had to adjust its own position to remain in a formation as described in section 6.2.3. Further, the robot turned its head towards the person closest to it in order to give an impression of listening to the conversation. Finally, the robot was leaving the group and heading for the door where it entered the room in the first place. Robovie was not engaged in the conversation and acted as a passive listener. Therefore, leaving the group is not triggered by the end of the conversation. The robot simply returns to its old task after 2 minutes of listening. The path through the state diagram for this task can be seen in Figure 6.4.

The three subjects used for the experiment have never seen the robot before. This was an important condition, because they must not know about the platform’s moving patterns and automatically adjust to it. The main constraints on the navigation are that the robot moves naturally in a similar way as humans do.

The subjects where questioned about their impressions after the experiment. They judged the moving patterns of the robot as natural and had the impression that Robovie was listening to their conversation.

A video has been taken of the experiment. However, these kind of results are very difficult to present on paper. An attempt to give an impression of the experiment is done by displaying trajectories. The room where the interaction took place is equipped with a system for tracking optical beacons with a large number of cameras. Such beacons were placed on the subjects and the robot in order to track their position during the experiment. The movements of the robot and the subjects during a conversation with a constant amount of people are really small and hard to draw and interpret. However, during major events, for example when a person is leaving, the robot has to move bigger distances to remain in formation with the humans. The trajectories during some of these major events are displayed and discussed below.

Figures 6.5 to 6.8 show the trajectories of the robot (dashed line) and the subjects (solid lines) for four short sequences during the interaction task. The units are measured in meters. The circles indicate the approximate size of the robot and the persons. The dashed circles indicate the position at the beginning of the sequence. The solid ones show the final positions when the pictures on the right of the plots were taken.

Figure 6.5 shows the trajectories just after the robot has entered the door and is approaching the group of people. Before the robot got there the persons were arranged in a regular triangle (dashed circles). As soon as the robot came close, they rearranged their positions to open up space for the platform. Robovie positioned itself such that the four have about equal distances to each other.

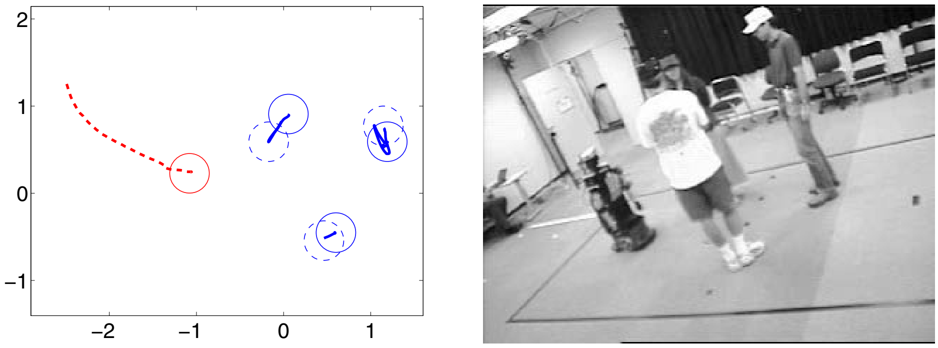

Figure 6.5. The robot approaches the group of people.

In Figure 6.6 maintaining formation after a person was leaving can be observed. At the beginning of the sequence the four stood at about equal distances to each other (dashed circles). Then, the person on the right hand side was leaving. The other two and Robovie were moving in such way that the new formation constitutes a rather regular triangle (solid lines).

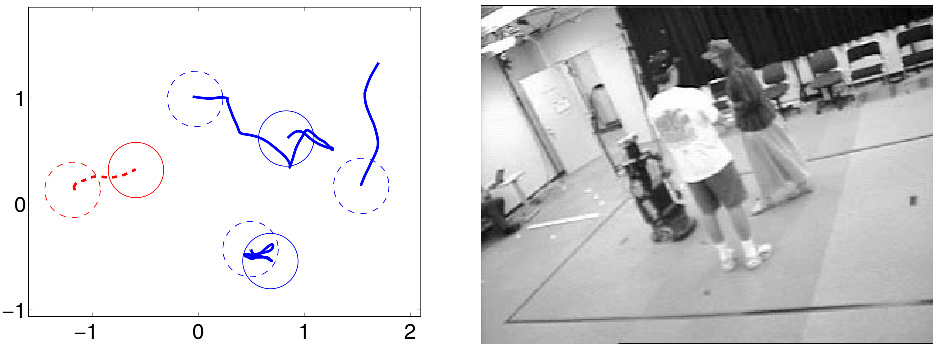

Figure 6.6. The person on the right hand side is leaving the formation.

In Figure 6.7, it can be observed how the subjects were moving when the third person joined the group again. At first, the two persons and the robot were at similar positions as at the end of the sequence of Figure 6.7. Then the third person was approaching from the top left. Robovie and the person on the right are opening up space, such that the four of them constitute a similar formation as in the beginning of the sequence of Figure 6.7.

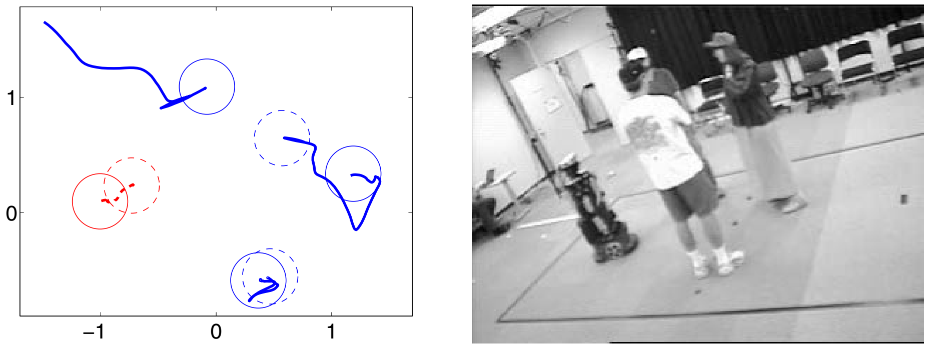

Figure 6.7. The third person is returning and approaching the group.

Figure 6.8 displays the end of the interaction task. The dashed circles show the individuals in regular formation. Then after 2 minutes of interaction, the end of the task was triggered and the robot left the group in the direction from where it came from.

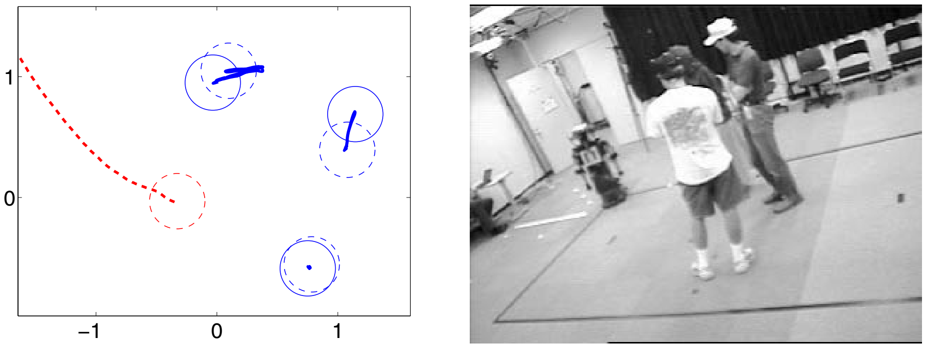

Figure 6.8. The robot leaves the group of people.

6.5 Discussion

A basic scenario for testing navigation in human-robot interaction tasks has been defined. Two new behaviours have been designed, which facilitate the system to approach a group of people and to maintain a certain formation while interacting. Experiments showed that the robot is able to reposition itself when these formations change due to different events. Further, the way the robot moves and finds its position in the group has been judged naturally by the test subjects.

Human-robot interaction poses different constraints on navigation than delivery missions, which were the subject of the earlier chapters of this thesis. However, in this chapter the following has been shown: 1) The dynamical systems approach is also suited for behaviour design in these types of tasks. 2) The world model (topological map) could easily be extended in order to contain information relevant for interaction. 3) The anchoring processes could be organised in the same way as in the proposed framework. In essence, this shows that the methodologies used in this thesis are not restricted to fetch-and-carry type mission. They can easily be extended to form the basis of more general navigation systems.

There are a number of other robotic systems that move around in spaces shared with humans and that engage in some form of interaction (see (Fong et al., 2003) for a survey). One type of systems that attract a great deal of attraction and operate in crowded spaces are museum tour guides. Rhino in the “Deutsche Museum Bonn” (Burgard et al., 1999), its successor Minerva in the “Smithsonian's National Museum of American History” in Washington (Thrun et al., 1999), and Robox at the “Swiss National Exhibition Expo02” (Siegwart et al., 2003) are the most popular examples. All these systems possess the ability to interact with humans. However, this interaction is based on speech, gestures, and other visual cues. Navigation is not considered in the context of an interactive purpose. In Robox, people are treated as dynamic obstacles only, which are fed to an obstacle avoidance module (Philippsen and Siegwart, 2003). In Rhino and Minerva, humans are detected by cameras in order to initiate interaction tasks (Schulte et al., 1999). These tasks are restricted to the upper body of the robot and do not influence the navigation system.

Only recently, some research has been devoted to the role of navigation for socially interactive robots. Simply turning towards persons of interest allows the robot to express some personality (Bruce et al., 2002; Okuno et al., 2003). Tan et al. (2003) made some first considerations of human-robot formations for cooperative manipulation. Further, Nakauchi and Simmons (2000) discussed the personal spaces of humans similar to the psychological arguments in section 6.2.3. They considered these constraints when implementing a robot for standing in a line to get a coffee. The work presented in this chapter is another initial example of incorporating navigation into the interaction loop. We believe that this is necessary and will be seen in personal robotic systems in the future.

The design and results presented in this chapter are clearly some simplified initial studies. Human formation are, in general, more complex than discussed here. For example, the distance to other people is dependent on the personal relations among them (Hall, 1966). Also the type of interactive means used (gestures or speech) influence the formations among the participants. This information could be incorporated into a system by using modules for recognising individual humans and interpreting their gestures (Furukawa et al., 2003). This system would also facilitate to notice differences between humans and objects like chairs. This again will lead to more advanced formation control than in the present implementation.

Chapter 7

Discussion

In the future, different indoor applications of autonomous robots are expected to open up markets for the industry. Various tasks in domestic environments can be executed by mobile systems, such as vacuum cleaning, delivery missions, or help for elderly people. This thesis deals with several aspects of indoor navigation for such mobile platforms. These systems have to work successfully in any kind of area consisting of rooms and corridors. Hence, the emphasis in the work presented here is on minimalistic models that rely only on the very basic structures of such environments. Thus, the system is designed in a general fashion without any detailed assumptions on the specific setting. Furthermore, no means of engineering the environment have been used. Such investments are believed to be too expensive for the mass market.

The core of this thesis is the development of the indoor navigation system for delivery type missions. Its design, the methods used, and the results are summarised and discussed in section 7.1. In section 7.2, the two extensions presented are reviewed: automatic map acquisition and navigation in human-robot interaction. Finally, the thesis concludes in section 7.3 with a discussion of the major issues that were not tackled in this work.

7.1 Indoor Navigation

The dynamical systems approach was the framework of choice for designing the navigation system in a behaviour based fashion. This led to a hybrid deliberative architecture, as in most existing navigation approaches. This architecture defines two levels of abstraction. On the one hand, the reactive level assures safe navigation based on behaviours taking into account the local features of the environment. On the other hand, the deliberative part coordinates these behaviours using abstract context information and a predefined world model. Hence, this part is responsible for task switching and long-term mission planning.

For the control on the reactive level, five different behaviours have been designed motivated by the general structure of a domestic environment: GO TO, OBSTACLE AVOIDANCE, CORRIDOR FOLLOWING, WALL AVOIDANCE, and DOOR PASSING. Each of these behaviours provides the robot with the ability to solve a navigational subtask. Their control laws are defined by a dynamical system. The individual task constraints are expressed as fixpoints in the space of the two behavioural variables: the heading direction ϕ and the speed v. These fixpoints and other parameters of the dynamical systems are specified by the geometrical properties of objects surrounding the robot. The perception of these objects is mainly governed by the type of sensors the platform is equipped with. Here, sonars provide the data from which abstract representations of the objects are extracted. This is done by feature detectors for OBSTACLES, CORRIDORS, DOORS, and WALLS. The precision of the properties of these symbols is rather important, since the behaviours are responsible for safe and accurate navigation. On the other hand, the problem of maintaining these properties over time can be neglected here. The behaviours are reactive modules that respond to immediate sensory events. At what time the feature detectors have to be run and the symbols anchored is determined by the context information of the deliberative level.

The properties of the symbols parameterise the dynamical system of each behaviour. This allows decision making processes on the reactive level. In the dynamical systems approach, this decision making is expressed by bifurcations in the dynamics of the behavioural variables. This was illustrated in more detail by a situation where the robot has to decide if a gap between two obstacles is wide enough to pass between them or not. At such a bifurcation, the amount and nature of fixpoints changes, which leads to a qualitatively different behaviour of the robot. Not only constraints on the individual behaviours, but also the conditions for the occurance of such a bifurcation can be expressed precisely in the mathematical formalism of this approach.

On the deliberative level, the activation of the different behaviours is coordinated. They are switched on and off on individual timescales through a competition framework defined by another dynamical system. Next to these timescales, the competitive dynamics is parameterised by so called competitive advantages and competitive interactions. They express the applicability of a behaviour and conflicts among them, dependent on the current situation. To express this information, abstract symbols are defined: SENSORY CONTEXT and TOPOLOGICAL CONTEXT. The SENSORY CONTEXT reflects detection of symbols on the reactive level, like open doors and the presence of obstacles. Furthermore, the battery level is monitored to determine when the platform has to get back to the charging station. The TOPOLOGICAL CONTEXT keeps track of the the robot's position within a topological map. Such a map has been defined constituting the world model of the system. It reflects the large scale structure of the environment consisting of rooms, corridors, and doors. This map facilitates task switching by recognising places in the environment using the symbol POSE, which is an estimate of the robot's position and orientation.

When anchoring the symbols on this level, emphasis is put on recognition of context and maintenance of the pose over time. The exact properties are of minor importance, since the execution of subtasks on the reactive level is deals with accurate navigation. Switching between these subtasks is the form of decision making on the deliberative level. Also here, this is expressed by bifurcations, this time in the competitive dynamics of the weights.

The navigation system has been tested on the premises of our institute, an environment of about 20×70 metres. For avoiding obstacles, passing between them, and avoiding walls, the performance predicted from the mathematical analysis in the system design was displayed. Furthermore, the robot was able to accomplish missions over long distances and to cope with unforeseen situations. The pose estimate of the robot could be maintained throughout the trial. The accuracy of this estimate was sufficient to use the topological map successfully and perform the appropriate task switches. In addition, the same system with identical parameters has been used to successfully navigate a different platform in a different setting. This further underlines the generality of the design.

In this work, the dynamical systems approach has been applied for the first time to realistic tasks in large-scale real-world environments. In addition, it has been extended to the entire control of the platform via integrated speed control. As opposed to most earlier approaches, where only the heading direction is controlled through this framework. Furthermore, the behaviour coordination scheme has been exploited more thoroughly. It was shown that this scheme allows a rich variety of combinations of individual behaviours. In particular, both arbitration and fusion can be achieved. The form of the topological map defined in this thesis, facilitates this coordination of sets of behaviours. By using this map to parameterise the competitive dynamics, it was demonstrated how continuous values of geometrical properties and discrete information about the environment's topology can be integrated into the same framework. Moreover, the dynamics ensure a continuous control signal at all times, since the behaviours are switched on and off on finite timescales. In essence, the system presented shows that one can relax the need for continuous pose estimation while maintaining a possibility for smooth behaviour switching. This continuous control signal can even be achieved for decision making, which is actually a discrete process, through the occurance of bifurcations in the dynamical systems.

All these representations, if continuous geometrical properties or discrete context information, must be anchored in the real-world. This problem, which is computationally very expensive in general, was considerably simplified in this work. The symbols used for the two levels of control need to have rather different properties. Hence, their anchoring processes can be distinguished, which was expressed in proposing a general framework for this problem. This framework could be a design guideline for anchoring in any hybrid deliberative architecture.

In this thesis, the issues of control and representations were mostly described separately for clarity of presentation. However, they are in no way independent of each other. For example, the design and desired performance of an OBSTACLE AVOIDANCE behaviour determines the properties that an OBSTACLE representation needs to have. On the other hand, all the available information on the representation side (for example the topological map) poses obvious constraints on the control system. This interdependence does not necessarily complicate matters. On the contrary, carefully chosen forms of representations can simplify the design of a controller. Alternatively, as seen here, a suitable design of the control system can relax the need for a complex geometrical map.

7.2 Extensions

It was important to show that the system discussed above is not restricted to navigation for delivery missions. A certain generality of the approach including its methods was demonstrated by two additional applications: automatic map acquisition and navigation for human-robot interaction tasks.

The choice of the map representation was validated by presenting a rather simple method for acquiring it in a semi-autonomous fashion. The framework has been extended by the behaviour PERSON FOLLOWING, which allows the platform to follow a guide through the environment. Via a terminal interface, this person informs the system about important places on their way. The system builds the map online by determining the position of the nodes using some basic geometrical computations. This is clearly an advantage to general SLAM approaches deploying complex world models and demanding high computational power. In these approaches the map most often can only be computed offline after the sensory data has been collected. Although the method presented here results in much simpler representation of the environment, it can be used successfully for navigation purposes.

Another advantage of the map representation used in this work is that it easily could be extended to a state diagram. This diagram reflects both locations in the environment and events in an interaction task. This information facilitates task switching using the same principles as in pure fetch-and-carry navigation. Two additional behaviours have been designed in order to approach a group of humans and to join this group in a conversation. Experiments showed that the platform is able to position itself relative to the people and maintain this formation in a similar way as humans do. In essence, the suitability of the principles of the control system, namely the dynamical systems approach, has been established further.

7.3 Open Issues

Many open issues and suggestions for future research were addressed in the discussion sections of each individual chapter. Here, only the main points are summarised.

Localisation is a major issue for mobile robots. In order to reach a destination in an environment the system needs to know its position and orientation. It has been shown that the need for precise information can be relaxed. Through the use of the topological map and an approximate estimate of the robot's pose, successful missions to any position in the area are possible. However, also this can fail for a number of reasons. Odometry data could be corrupted, or slippage of the wheels could be unexpectedly high, for example when driving over a high threshold at a door. A similar issue is the "kidnapped robot" problem, where the platform is lifted and placed anywhere in the environment. In that case, the system described in this thesis would be lost. To resolve this deficiency, recognition of places (in other words, the nodes in the topological map) has to be possible by other means than odometry. This feature combined with a reasoning module could be able to relocate the robot in the map.

In addition to a solution to the "kidnapped robot" problem, an error detection mechanism is needed to realise that this failure had occured in the first place. Also for other kinds of problems a detection mechanism would prove useful. For example, corridors that are permanently blocked or general inconsistencies of the map with the environment should force the system to update the map and replan its missions accordingly.

Both the recognition of places mentioned above and the detection of errors would probably need more complex sensing capabilities. Here, only sonars were used to extract features. The emphasis of this thesis was not on sensor processing. Thus, minimalistic models of objects were deployed. Nevertheless, this was enough to demonstrate the functioning of the navigation system. However, there probably are possible settings were the system could fail. For example, two open doors located right next to each other could make the platform enter the wrong room. Also the mapping algorithms and the human-robot interaction control could only gain from more advanced sensing capabilities. However, no complete geometrical models of the environment are needed. It should always be considered what exactly is needed for a navigational task to be executed. In particular, when the amount of information that the map needs to contain is to be determined, the competence of the control system should not be forgotten.

Bibliography

Albus, J. S. (1991). Outline for a theory of intelligence, *IEEE Transactions on Systems, Man, and Cybernetics* **21**(3): 473–509.

Aloimonos, Y. (ed.) (1993). *Active Perception*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Althaus, P. and Christensen, H. I. (2002a). Behaviour coordination for navigation in office environments, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2298–2304.

Althaus, P. and Christensen, H. I. (2002b). Smooth task switching through behaviour competition, *Intelligent Autonomous Systems 7*, IOS Press, Amsterdam, NL, pp. 9–17.

Althaus, P. and Christensen, H. I. (2003a). Automatic map acquisition for navigation in domestic environments, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1551–1556.

Althaus, P. and Christensen, H. I. (2003b). Behavior coordination in structured environments, *Advanced Robotics* **17**(7): 657–674.

Althaus, P. and Christensen, H. I. (2003c). A framework for anchoring in hybrid deliberative systems, *Autonomous Robots* . submitted.

Althaus, P. and Christensen, H. I. (2003d). Smooth task switching through behaviour competition, *Robotics and Autonomous Systems* **44**(3-4): 241–249.

Althaus, P., Christensen, H. I. and Hoffmann, F. (2001). Using the dynamical system approach to navigate in realistic real-world environments, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1023–1029.

Althaus, P., Ishiguro, H., Kanda, T., Miyashita, T. and Christensen, H. I. (2004). Navigation for human-robot interaction tasks, *Proceedings of the IEEE International Conference on Robotics and Automation.* submitted.

Arbib, M. A., Iberall, T. and Lyons, D. (1987). Schemas that integrate vision and touch for hand control, *in* M. A. Arbib and A. R. Hanson (eds), *Vision, Brain, and Cooperative Computation*, MIT Press, Cambridge, MA, chapter 13, pp. 489–510.

Arkin, R. C. (1990). The impact of cybernetics on the design of a mobile robot system: a case study, *IEEE Transactions on Systems, Man, and Cybernetics* **20**(6): 1245–1257.

Arkin, R. C. (1998). *Behavior-Based Robotics*, MIT Press, Cambridge, MA.

Arkin, R. C. and MacKenzie, D. (1994). Temporal coordination of perceptual algorithms for mobile robot navigation, *IEEE Transactions on Robotics and Automation* **10**(3): 276–286.

Arras, K. O., Castellanos, J. A., Schilt, M. and Siegwart, R. (2003). Feature-based multi-hypothesis localization and tracking using geometric constraints, *Robotics and Autonomous Systems* **44**(1): 41–53.

Bajcsy, R. (1988). Active perception, *Proceedings of the IEEE* **76**(8): 996–1005.

Bajscy, R. and Large, E. W. (1999). When and where will AI meet robotics, *AI Magazine* **20**(3): 57–65.

Beer, R. D. (1995). A dynamical systems perspective on agent-environment interaction, *Artificial Intelligence* **72**(1-2): 173–215.

Bicho, E. (2000a). Detecting representing and following walls based on low-level distance sensors, *Proceedings of the Second International ICSC Symposium on Neural Computation.*

Bicho, E. (2000b). *Dynamic Approach to Behavior-Based Robotics: Design, Specification, Analysis, Simulation and Implementation*, Shaker Verlag, Aachen, D. PhD thesis, University of Minho, P.

Bicho, E., Mallet, P. and Schöner, G. (2000). Target representation on an autonomous vehicle with low-level sensors, *The International Journal of Robotics Research* **19**(5): 424–447.

Bicho, E. and Schöner, G. (1997). The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform, *Robotics and Autonomous Systems* **21**(1): 23–35.

Borenstein, J. and Koren, Y. (1991). The vector field histogram – fast obstacle avoidance for mobile robots, *IEEE Transactions on Robotics and Autonomous Systems* **7**(3): 278–288.

Braitenberg, V. (1984). *Vehicles. Experiments in Synthetic Psychology*, MIT Press, Cambridge, MA.

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

Brooks, R. A. (1990). Elephants don't play chess, *Robotics and Autonomous Systems* **6**: 3–15.

Brooks, R. A. (1991a). Intelligence without representations, *Artificial Intelligence* **47**(1-3): 139–159.

Brooks, R. A. (1991b). New approaches to robotics, *Science* **253**: 1227–1232.

Brooks, R. A. (1997). From earwigs to humans, *Robotics and Autonomous Systems* **20**(2-4): 291–304.

Bruce, A., Nourbakhsh, I. and Simmons, R. (2002). The role of expressiveness and attention in human-robot interaction, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4138–4142.

Burgard, W., Cremers, A. B., Fox, D., Hänel, D., Lakemeyer, G., Schulz, D., Steiner, W. and Thrun, S. (1999). Experiences with an interactive museum tour-guide robot, *Artificial Intelligence* **114**(1-2): 3–55.

Castellanos, J. A. and Tardos, J. D. (1999). *Mobile Robot Localization and Map Building*, Kluwer Academic Publishers, Boston, MA.

Chatila, R. (1995). Deliberation and reactivity in autonomous mobile robots, *Robotics and Autonomous Systems* **16**(2-4): 197–211.

Christensen, H. I., Kirkeby, N. O., Kristensen, S. and Knudsen, L. (1994). Model-driven vision for in-door navigation, *Robotics and Autonomous Systems* **12**(3-4): 199–207.

Coradeschi, S. and Saffiotti, A. (2000). Anchoring symbols to sensor data: preliminary report, *Proceedings of the 17th AAAI National Conference on Artificial Intelligence*, pp. 129–135.

Coradeschi, S. and Saffiotti, A. (2001). Perceptual anchoring of symbols for action, *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pp. 407–412.

Coradeschi, S. and Saffiotti, A. (2003). An introduction to the anchoring problem, *Robotics and Autonomous Systems* **43**(2-3): 85–96.

Crowley, J. (1985). Navigation for an intelligent mobile robot, *IEEE Journal of Robotics and Automation* **1**(1): 31–41.

Elfes, A. (1987). Sonar-based real-world mapping and navigation, *IEEE Journal of Robotics and Automation* **3**(3): 249–265.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation, *Computer* **22**(6): 46–57.

Engels, C. and Schöner, G. (1995). Dynamic fields endow behavior-based robots with representations, *Robotics and Autonomous Systems* **14**(1): 55–77.

Fabrizi, E. and Saffiotti, A. (2002). Augmenting topology-based maps with geometric information, *Robotics and Autonomous Systems* **40**(2-3): 91–97.

Fong, T., Nourbakhsh, I. and Dautenhahn, K. (2003). A survey of socially interactive robots, *Robotics and Autonomous Systems* **42**(3-4): 143–166.

Forsberg, J., Larsson, U. and Wernersson, Å. (1995). Mobile robot navigation using the range-weighted hough transform, *IEEE Robotics & Automation Magazine* **2**(1): 18–26.

Furukawa, M., Kanbara, Y., Minato, T. and Ishiguro, H. (2003). Human behavior interpretation system based on view and motion-based aspect models, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4160–4165.

Gat, E. (1995). Towards principled experimental study of autonomous mobile robots, *Autonomous Robots* **2**(3): 179–189.

Gonzales, R. C. and Woods, R. E. (1992). *Digital Image Processing*, Addison-Wesley, Reading, MA.

Guivant, J. E. and Nebot, E. M. (2001). Optimization of the simultaneous localization and map-building algorithm for real-time implementation, *IEEE Transactions on Robotics and Automation* **17**(3): 242–257.

Hall, E. T. (1966). *The Hidden Dimension*, Doubleday, Garden City, NY.

Ho, Y. C. (1991). *Introduction to Discrete Event Systems*, IEEE Press.

Horswill, I. (1998). The Polly system, *in* D. Kortenkamp, R. P. Bonasso and R. Murphy (eds), *Artificial Intelligence and Mobile Robots: Case studies of successful robot systems*, MIT Press, Cambridge, MA, chapter 5, pp. 125–139.

Hutchinson, S., Hager, G. D. and Corke, P. I. (1996). A tutorial on visual servo control, *IEEE Transactions on Robotics and Automation* **12**(5): 651–670.

Ishiguro, H., Ono, T., Imai, M. and Kanda, T. (2003). Development of an interactive humanoid robot "Robovie" - An interdisciplinary aproach, *in* R. A. Jarvis and A. Zelinsky (eds), *Robotics Research*, Springer, Berlin, pp. 179–191.

Jensfelt, P. and Christensen, H. I. (2001). Pose tracking using laser scanning and minimalistic environmental models, *IEEE Transactions on Robotics and Automation* **17**(2): 138–147.

Jensfelt, P. and Kristensen, S. (2001). Active global localization for a mobile robot using multiple hypothesis tracking, *IEEE Transactions on Robotics and Automation* **17**(5): 748–760.

Kanda, T., Ishiguro, H., Imai, M., Ono, T. and Mase, K. (2002). A constructive approach for developing interactive humanoid robots, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1265–1270.

Kanda, T., Ishiguro, H. and Ishida, T. (2001). Psychological analysis on human-robot interaction, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4166–4173.

Kanda, T., Ishiguro, H., Ono, T., Imai, M. and Nakatsu, R. (2002). Development and evaluation of an interactive humanoid robot "Robovie", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1848–1855.

Kendon, A. (1990). *Conducting Interaction*, Cambridge University Press, chapter 7: Spatial organization in social encounters: the F-formation system, pp. 209–237.

Khalil, H. K. (1995). *Nonlinear Systems*, 2nd edn, Prentice Hall, New Jersey.

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

Koenig, S. and Simmons, R. G. (1998). Xavier: A robot navigation architecture based on partially observable markov decision process models, *in* D. Kortenkamp, R. P. Bonasso and R. Murphy (eds), *Artificial Intelligence and Mobile Robots: Case studies of successful robot systems*, MIT Press, Cambridge, MA, chapter 4, pp. 91–122.

Konolige, K. (1997). Improved occupancy grids for map building, *Autonomous Robots* **4**(4): 351–367.

Kosecká, J. and Bajcsy, R. (1994). Discrete event systems for autonomous mobile agents, *Robotics and Autonomous Systems* **12**(3-4): 187–198.

Kosecká, J., Christensen, H. I. and Bajcsy, R. (1995). Discrete event modeling of visually guided behaviors, *International Journal of Computer Vision* **14**(2): 179–191.

Kosecká, J., Christensen, H. I. and Bajcsy, R. (1997). Experiments in behavior composition, *Robotics and Autonomous Systems* **19**(3-4): 287–298.

Kuipers, B. J. and Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations, *Robotics and Autonomous Systems* **8**(1-2): 47–63.

Kuipers, B. J., Froom, R., Lee, W.-Y. and Pierce, D. (1993). The semantic hierarchy in robot learning, *in* J. Connell and S. Mahadevan (eds), *Robot Learning*, Kluwer Academic Publishers, Boston, MA, pp. 141–170.

Kuipers, B. J. and Levitt, T. S. (1988). Navigation and mapping in large-scale space, *AI Magazine* **9**(2): 25–43.

Large, E. W., Christensen, H. I. and Bajcsy, R. (1999). Scaling the dynamic approach to path planning and control: Competition among behavioral constraints, *The International Journal of Robotics Research* **18**(1): 37–58.

Latombe, J.-C. (1991). *Robot Motion Planning*, Kluwer Academic Publishers, Boston, MA.

Leonard, J. J. and Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons, *IEEE Transactions on Robotics and Automation* **7**(3): 376–382.

Lyons, D. M. (1993). Representing and analyzing action plans as networks of concurrent processes, *IEEE Transactions on Robotics and Automation* **9**(3): 241–256.

Lyons, D. M. and Hendriks, A. J. (1995). Planning as incremental adaptation of a reactive system, *Robotics and Autonomous Systems* **14**(4): 255–288.

Marr, D. C. (1982). *Vision*, W.H. Freeman and Co., San Francisco, CA.

Mataric, M. J. (1992). Integration of representation into goal-driven behavior-based robots, *IEEE Transactions on Robotcis and Automation* **8**(3): 304–312.

Mataric, M. J. (1997). Behavior-based control: Examples from navigation, learning, and group behaviour, *Journal of Experimental and Theoretical Artificial Intelligence* **9**(2-3): 323–336.

Monteiro, S. and Bicho, E. (2002). A dynamical systems approach to behavior-based formation control, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2606–2611.

Moravec, H. P. (1983). The Stanford Cart and the CMU Rover, *Proceedings of the IEEE* **71**(7): 872–884.

Moravec, H. P. (1988). Sensor fusion in certainty grids for mobile robots, *AI Magazine* **9**(2): 61–74.

Nakauchi, Y. and Simmons, R. (2000). A social robot that stands in line, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 357–364.

Neven, H. and Schöner, G. (1996). Dynamics parametrically controlled by image correlations organize robot navigation, *Biological Cybernetics* **75**(4): 293–307.

Nourbakhsh, I. (1998). Dervish: An office-navigating robot, *in* D. Kortenkamp, R. P. Bonasso and R. Murphy (eds), *Artificial Intelligence and Mobile Robots: Case studies of successful robot systems*, MIT Press, Cambridge, MA, chapter 3, pp. 73–90.

Nourbakhsh, I., Powers, R. and Birchfield, S. (1995). DERVISH: An office-navigating robot, *AI Magazine* **16**(2): 53–60.

Okuno, H. G., Nakadai, K. and Kitano, H. (2003). Realizing personality in audio-visually triggered non-verbal behaviors, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 392–397.

Pagac, D., Nebot, E. M. and Durrant-Whyte, H. (1998). An evidential approach to map-building for autonomous vehicles, *IEEE Transactions on Robotics and Automation* **14**(4): 623–629.

Payton, D. W., Rosenblatt, J. K. and Keirsey, D. M. (1990). Plan guided reaction, *IEEE Transactions on Systems, Man, and Cybernetics* **20**(6): 1370–1382.

Perko, L. (1991). *Differential Equations and Dynamical Systems*, Springer, New York.

Peterson, G. L. and Cook, D. J. (2003). Incorporating decision-theoretic planning in a robot architecture, *Robotics and Autonomous Systems* **42**(2): 89–106.

Philippsen, R. and Siegwart, R. (2003). Smooth and efficient obstacle avoidance for a tour guide robot, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 446–451.

Pirjanian, P. (1998). *Multiple Objective Action Selection & Behaviour Fusion using Voting*, PhD thesis, Aalborg University, DK.

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

Poncela, A., Perez, E. J., Bandera, A., Urdiales, C. and Sandoval, F. (2002). Efficient integration of metric and topological maps for directed exploration of unknown environments, *Robotics and Autonomous Systems* **41**(1): 21–39.

Rimon, E. and Koditschek, D. E. (1992). Exact robot navigation using artificial potential functions, *IEEE Transactions on Robotics and Automation* **8**(5): 501–518.

Rosenblatt, J. K. (1997). DAMN: A distributed architecture for mobile navigation, *Journal of Experimental and Theoretical Artificial Intelligence* **9**(2-3): 339–360.

Rosenblatt, J. K. (2000). Optimal selection of uncertain actions by maximizing expected utility, *Autonomous Robots* **9**(1): 17–25.

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

Schöner, G. (1991). Dynamic theory of action-perception patterns: The 'moving room' paradigm, *Biological Cybernetics* **64**: 455–462.

Schöner, G. and Dose, M. (1992). A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion, *Robotics and Autonomous Systems* **10**(4): 253–267.

Schöner, G., Dose, M. and Engels, C. (1995). Dynamics of behavior: theory and applications for autonomous robot architectures, *Robotics and Autonomous Systems* **16**(2-4): 213–245.

Schöner, G. and Kelso, J. A. S. (1988). Dynamic pattern generation in behavioral and neural systms, *Science* **239**: 1513–1520.

Schulte, J., Rosenberg, C. and Thrun, S. (1999). Spontaneous, short-term interaction with mobile robots, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 658–663.

Schulz, D., Burgard, W., Fox, D. and Cremers, A. B. (2001). Tracking multiple moving objects with a mobile robot using particle filters and statistical data association, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1665–1670.

Siegwart, R., Arras, K. O., Bouabdallah, S., Burnier, D., Froidevaux, G., Greppin, X., Jensen, B., Lorotte, A., Mayor, L., Meisser, M., Philippsen, R., Piguet, R., Ramel, G., Terrien, G. and Tomatis, N. (2003). Robox at Expo.02: A large-scale installation of personal robots, *Robotics and Autonomous Systems* **42**(3-4): 203–222.

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

Simon, H. A. (1981). *The Sciences of the Artificial*, 2nd edn, MIT Press, Cambridge, MA.

Steinhage, A. (1998). *Dynamical Systems for the Generation of Navigation Behavior*, Shaker Verlag, Aachen, D. PhD thesis, Ruhr-University, Bochum, D.

Steinhage, A. and Schöner, G. (1997). Self-calibration based on invariant view recognition: Dynamic approach to navigation, *Robotics and Autonomous Systems* **20**(2-4): 133–156.

Tan, J., Xi, N., Goradia, A. and Sheng, W. (2003). Coordination of human and mobile manipulator formation in a perceptive reference frame, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 374–379.

Thrun, S. (1998). Finding landmarks for mobile robot navigation, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 958–963.

Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Hähnel, D., Rosenberg, C., Roy, N., Schulte, J. and Schulz, D. (1999). MINERVA: A second-generation museum tour-guide robot, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1999–2005.

Thrun, S., Burgard, W. and Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 321–328.

Tomatis, N., Nourbakhsh, I. and Siegwart, R. (2003). Hybrid simultaneous localization and map building: a natural integration of topological and metric, *Robotics and Autonomous Systems* **44**(1): 3–14.

Tsotsos, J. K. (1990). Analyzing vision at the complexity level, *Behavioral and Brain Sciences* **13**(3): 423–469.

Tsotsos, J. K. (1995). Behaviorist intelligence and the scaling problem, *Artificial Intelligence* **75**(2): 135–160.

Ulrich, I. and Nourbaksh, I. (2000). Appearance-based place recognition for topological localization, *Proceedings of the IEEE International Conference of Robotics and Automation*, pp. 1023–1029.

Veelaert, P. and Bogaerts, W. (1999). Ultrasonic potential field sensor for obstacle avoidance, *IEEE Transactions on Robotics and Automation* **15**(4): 774–779.

Wijk, O. and Christensen, H. I. (2000). Triangulation based fusion of sonar data with application in robot pose tracking, *IEEE Transactions on Robotics and Automation* **16**(6): 740–752.

Wulf, O. (2001). *Recharging station for a team of mobile robots*, Master's thesis, Royal Institute of Technology, Stockholm, S.

Yen, J. and Pfluger, N. (1995). A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation, *IEEE Transactions on Systems, Man, and Cybernetics* **25**(6): 971–978.