Analysis and Synthesis in the Design of Locomotor and Spatial Competences for a Multisensory Mobile Robot

James Grant Donnett

Doctor of Philosophy
University of Edinburgh
1992

Declaration

In accordance with postgraduate degree Regulation 3.4.7 of the University of Edinburgh, I declare that the work described in this document is my own, except where otherwise indicated, and that this dissertation was composed by myself.

Abstract

This dissertation reports on experiments in the design of a free-ranging robotic vehicle with locomotor and spatial skills. These two skills are suggested to be fundamental to any general mobile agent. No attempt has been made to incorporate manipulation, and motivational issues are not treated (the system's objectives are specified by a human).

Two schools of thought with regard to robot design prevail: they are the representation-based or *analytic*, and representation-free or *synthetic* schools. These guiding philosophies are compared and contrasted in terms of their approaches to locomotion and localisation.

In the course of this work, it has been found that the smoothness of a mobile robot's locomotion benefits from adjusting its speed to the density of obstacles it is encountering. A reactive, synthetic-type procedure for achieving this is offered, based on the frequency at which obstacle detection interrupts the robot's navigational system. However, it will be shown that a predictive, representation-based approach is more efficient.

On the other hand, it will emerge that a very limited representation of the robot's operating environment and sensing systems is sufficient to allow the robot to localise accurately anywhere within its operating space, based only on the goodness of match between mapped and currently sensed data. The scheme treats the infrared, sonic, and ultrasonic sensors of an autonomous mobile robot as devices which partition its operating space qualitatively. It will be shown that the infrared and ultrasonic systems combined lead to position estimates accurate to within 10 cm of where the robot actually is, on average. These systems are found to be superior to the sonic system, surprisingly, because they have distinct regions within which they do not work: it is better to have accurate sensors with dead zones than inaccurate ones which can be used anywhere.

Since it is not obvious which aspects of robot behaviour are well-suited, and which are unsuited for representation versus representation-free implementation, a conclusion drawn from the work is that elements of both approaches bear further consideration, and might be usefully combined.

Acknowledgements

I am indebted to a large number of people for their help and encouragement through the past four years. It is impossible to acknowledge in a few lines the myriad ways in which I have benefitted from the company of my fellow students and the staff of the Dept. of Artificial Intelligence. These paragraphs (in chronological order) can only hope to recognise the most tangible aspects of this support.

I wish to thank Mike Moran and Tom Alexander for technical assistance and advice in electronic design matters. Hugh Cameron, Joe Cairney, and David Wyse suffered (gladly?) many urgent requests for mechanical work in the construction and modification of robot prototypes. Douglas Howie produced a large volume of photographic printed circuit board masks, and his ready anecdotes brightened my many hours spent etching PCBs in the darkroom.

I am grateful to Brendan McGonigle for making available an embarrassment of laboratory space, to which I had almost exclusive access for a considerable time. His boundless enthusiasm was much appreciated, as were his interesting ideas, including the instigation for the locomotion work described in Chapter 4.

John Hallam motivated the qualitative localisation investigation that forms the core of the thesis. I am enormously beholden to John for gracefully shouldering the burden of leading me from despair and no written text to relative tranquility and a finished document. He was faced with many frantic requests for advice, and this was always offered pleasantly.

Several of my colleagues and friends read chapters of this dissertation, and replied with insightful and occasionally challenging comments; they were John Hallam, Gill Hayes, Chris Malcolm, Graham Deacon, Kate Jeffery, and David Lee. The final presentation has benefitted greatly from their opinions, and I thank them for their time. My parents took it upon themselves to proofread the text from cover to cover; although such a thankless task is in the spirit of parenthood, they have my gratitude for this and an uncountable multitude of other reasons.

The University of Edinburgh and the Government of Canada sustained me financially, with a studentship and an NSERC postgraduate scholarship, respectively.

Table of Contents

Chapter 1

Introduction

An autonomous mobile robot is a machine capable of moving around freely in some operating environment and effecting useful interactions with objects within this space. Disregarding questions of motivation, the fundamental behaviours such a device must exhibit are principally locomotor and spatial: in most cases, it must be able to move around without colliding in a damaging way with objects in its vicinity, and it must have some sense of where it is and how to get to other places. Since the mid-1960s (Nilsson, 1969), it has widely been regarded that understanding how to engineer such a machine is tantamount to understanding intelligent behaviour generally.

Efforts in the academic robotics community towards solving the problems of building autonomous agents have in recent years become divisible into two recognisably distinct categories (Malcolm *et al.*, 1989), here labelled *analytic* and *synthetic*, for reasons to be elaborated below. Systems ascribable to the more traditional of the two (analytic) are predicated on obtaining explicit and rich representations of the robot's relationship to its world; by analysing this relationship they plan a sequence of actions that are the means to some desired end. Conversely, systems falling into the second category (synthetic) dispense with explicit representations and their manipulation: empirical insights derived from bottom-up synthesis of the system lead to an arrangement of control elements from which appropriate behaviour emerges as a reaction mainly to the present relationship between robot and world.

Robot designs of the first kind are adversely affected when the representations they maintain fail to represent the true state of the robot, so that the actions that are planned and executed from these models are out of place. Designs of the second variety, where the "world is its own representation" (Brooks, 1991a), are appealing primarily because they are potentially immune to misrepresentation and thus always act in a manner appropriate to their immediate circumstances. Both broad approaches have discernible advantages and disadvantages. However, the current mood in the field appears to be that the two perspectives are incompatible (Brooks, 1991b).

In this dissertation, it will be argued that the design of autonomous robotic systems is better served by merging aspects of both approaches than by either one alone, and that strong partisanship at one extreme point of view or the other is therefore unwarranted. This position will be supported by contrasting observations made in developing autonomous vehicles capable of locomotion and localisation: on one hand, a minimal-representation approach is shown to be of limited utility for a simple locomotor task; on the other hand, it is found to be adequate for intuitively more complicated problems in the spatial domain. Furthermore, it will be noted that even a brief review of current thinking in neuroscience reveals that more complex animal behaviour (e.g., of mammals) can be reasonably characterised as having both representational ("declarative", Squire, 1992) and non-representational elements.

1.1 General problem space: "analysis" versus "synthesis"

While walking, talking, seemingly sentient androids like C3P0 remain the stuff of science fiction, it is nevertheless the case that literally tens of thousands of fixed and mobile robots already operate in our own world (Müller, 1983). Most technological sophisticates are aware of the highly automated practice of manufacturing motor vehicles, but the image of robots most frequently associated with this process is of fast-moving (and therefore eminently photogenic) manipulators performing welding, painting, or assembly tasks. Less prominent but nevertheless representing a huge financial investment are the free-ranging machines that transport raw and processed materials between fixed manufacturing cells and warehousing facilities (Milberg and Luts, 1981).

Beyond these traditional factory roles, many novel applications for mobile robots are being explored, from service tasks like mail delivery for office environments or delivery of medical supplies in hospitals (Kimura *et al.*, 1986; Layman, 1986), to the monitoring and maintenance of nuclear power stations (Carlton and Bartholet, 1987; White *et al.*, 1987; Mann *et al.*, 1988). Like the now-ubiquitous digital computer, it seems to follow that the increased visibility of robots will lead to an alternation between heightened expectations of their capabilities and refinement of their designs, until ultimately these machines become as apparently indispensable to our daily existence as have computers.

At first glance, it is surprising that robots have not already come into as widespread use as computers; surely the list of possible applications for both is equally

limitless. One possible explanation for the disparity might be that robots are more complex mechanisms than computers, as the latter have no "moving parts" and are therefore easier to engineer. That is, manipulators and robot vehicles consist of physical components that must be controlled within a framework of dynamic effects like gravity and inertia, often in the presence of other animate and inanimate objects. Although this distinction is partly true, it is also clear that at their lowest levels of functioning, computers are equally "active", and are exposed to unpredictable perturbations like electrical noise; but by organising their components in certain modular and layered ways, it has become possible to interact reliably and predictably with them, in remarkably plain language.

The proliferation of computers is seen, therefore, to be the result of the discovery of successful architectural principles. Unfortunately, no corresponding robust organisational scheme has yet been demonstrated for the control of autonomous robots. Most existing machines have been engineered for a particular application; while some appear to be capable of quite sophisticated behaviour, their competence is generally derived from highly specialised and hard-wired control, not easily transferable to new tasks, and often quite sensitive to perturbations even in their nominal operating environments (Arkin and Murphy, 1990).

Research efforts directed at discovering control architectural principles that lead to "intelligent" robot behaviour (in terms of increased flexibility and robustness) form a substantial subsidiary enterprise of robotics as an academic pursuit. In fact, the basic goal of building robots as "appliances" has been transcended, and some workers contend that solving the problems of building an autonomous robot is equivalent to achieving the objectives of artificial intelligence research generally (Brooks, 1986a).

Dramatic tension between conflicting approaches

Conventionally, robot architectures have been structured as a chain of processing operations that start at the robot's sensors and end at its actuators. The sensors are the system's "input" devices. They are used for obtaining information about the robot's relationship to its environment. This information updates the control system's internal ("mental") model of the robot's current circumstances. Given some specified objective, and the internal world model, the control system plans the set of motor commands that ought to result in the robot's state changing from what it is at present, to what it is desired to be. Following the performance of the planned action, the sensors are again used to assess the robot-world relationship, and the process repeats.

For example, suppose a robot controlled according to this scheme is commanded to travel from one end of a room to another. It might begin by obtaining from a camera an image of the room it is required to pass through. From the image, it attempts to distinguish objects from free space, and then from a model of the camera optics is able to determine the geometry of the room. With a geometric map, it is able to plan a trajectory that will allow it to pass in the free space between potential obstacles, from its start position to its eventual goal. It then attempts to execute this trajectory by translating it into a sequence of motor actuation instructions. The control system is depicted in Figure 1.1.

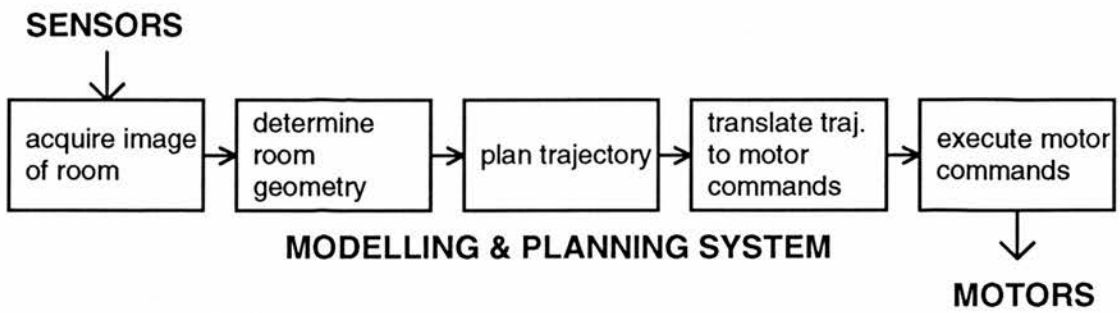

Figure 1.1. Typical conventional approach to basic motion control

There is much room for error in an approach like this. Extracting the geometry of the room depends on the accuracy of the camera model and segmentation of a possibly noisy image; any errors here would mean that the subsequent trajectory would not be appropriate to the true layout of the room. Even if the geometric modelling and trajectory planning are successful, almost inevitable mechanical inaccuracies could cause the robot not to follow the desired trajectory. Furthermore, a change in environment layout that occurs after the modelling phase has been completed, such as the introduction of a new object, could invalidate the pre-determined trajectory.

This well-entrenched conventional control procedure has been called the classical sense-think-act cycle by Malcolm *et al.* (1989), and the sense-model-plan-act framework by Brooks (1991b). Here, it will be called the *analytic* approach, partly because a robot thus controlled typically contains explicit procedures for *analysing* its situation in order to decide what to do next, partly because the approach tends to rely on formal mathematical analysis as an ingredient of the implementation, and partly because control problems are usually solved by a top-down analysis of the problem, rather than by an empirical, bottom-up synthesis of a working machine. At the root of most criticisms of it as a control architecture are the observations that any system implemented as a succession of processing steps is only as robust as the weakest link in

the chain, and that the ubiquitous modelling and planning stages are notoriously unreliable wherever the combination of robot, task, and environment is anything but trivial.

Given a system structured like this, there is only one route available for dealing with errors or dynamic changes such as those described in the room-traversing example: the system must cycle through sensing, thinking, and acting sufficiently frequently that the rate of errors or changes is balanced by the rate at which the system is able to compensate for them. As the complexity of most modelling and planning procedures requires that substantial computational effort be expended in thinking about what to do next, the robot in question ends up moving fairly slowly.

Dissatisfaction with this general procedure has led to the development of a number of alternative architectural initiatives by several different groups. These include the subsumption architecture of Brooks (1986b), Malcolm's (1987) SOMASS system, the situated automata of Rosenschein and Kaelbling (1986), and others. Such enterprises share some features, but are united into a class of ideas primarily in that they all represent attempts to overcome the problems inherent in the classical approach. Although it goes by many names in the literature, here this class of alternative ideas will be called the *synthetic* approach; this designation applies partly because a robot designed thus is controlled by *synthesising* or combining numbers of separate behavioural components into a connected whole, and partly because control problems are usually solved by an empirical, bottom-up synthesis of a working machine, rather than by a top-down analysis of the problem.

The most pervasive approach is Brooks' (1986b) subsumption architecture, by means of which a robot's macroscopic behaviour emerges from the interactions of individual, robust, task-achieving modules; each of these modules incorporates exactly those aspects of sensing and motor actuation that are directly relevant to achieving its task. In the case of the room-traversing example, the system might be structured as in Figure 1.2.

To achieve the required task of travelling from one side of the room to the other, this system uses three basic modules. The first simply causes the robot to move forwards. The second has some sensory device for determining the direction of the "other side of the room", and takes control of the motors as necessary to orient the robot appropriately. Finally, a third component is able to sense obstacles in the robot's path, and if one is encountered, suppresses both orientation and forward propulsion and steers the vehicle away from collision with the obstacle.

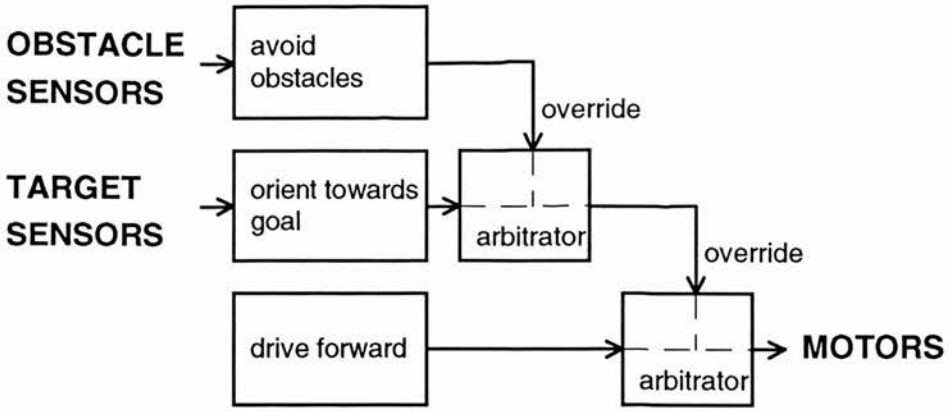

Figure 1.2. Typical subsumption architecture for basic motion control

Such a system performs no explicit modelling operations, and plans no trajectories, so that it eliminates these potential inroads for error. The interaction between moving forward and orienting towards the goal eventually gets it there regardless of room geometry. Once facing and moving towards the target, any differential slippage between its wheels that might cause it to drift off course is noticed and compensated for by the orientation mechanism. Furthermore, dynamic changes to environment layout are detected by the obstacle avoidance system, which suppresses motion towards the goal if ever it is necessary to steer the vehicle away from any obstructions.

Although this example and the two representative solutions for it are somewhat oversimplified, they are meant merely to give a flavour of the two contrasting approaches to structuring robot control systems. Much fuller treatments of the various issues involved are given by Malcolm *et al.* (1989) and Brooks (1991b), and in Chapter 2. In summary, it is important for the moment to recognise that:

- the analytic approach is characterised by largely centralised control systems, generally founded on all-encompassing representations of the relationship between the robot and its operating space; and

- the synthetic approach is characterised by control systems largely distributed into separate modules, such that these individual components interact, but lack shared models of robot and world, and instead encapsulate only the sensing and action elements of a particular aspect of the robot's observable behaviour.

1.2 Specific problem space: locomotion and localisation

On the surface, the synthetic approaches are very seductive because they appear to eliminate at source the causes of the problems of conventional schemes, namely modelling, representation, and planning. If the robot actions in the short term are always dictated by its current situation, then the concept of making an error does not exist. However, a number of criticisms have been raised in the context of these approaches (see e.g., Kirsh, 1991; Arkin and Lawton, 1990; Durrant-Whyte, 1988b), of which two are quite important: the first is that the demonstrated systems perform only simple tasks, and thereby easily sidestep modelling and representation requirements that might be inescapable in complex tasks; the second is that implementations tend simply to be labelled as either "successful" or not, and it is often unclear how to gauge the performance of a system in any absolute way, or even relative to other systems.

The work described in this document addresses these criticisms. It is an attempt to engineer a mobile robot that can move both robustly and *efficiently* through, and localise itself within, its operating space. These are notionally the two fundamental competences a free-ranging mobile agent must possess, if its mobility is to be expressed as anything other than random walking or perhaps wall-following. In terms of the first criticism (task difficulty), it would seem reasonable to predict that locomotion should be easy to implement in a strictly synthetic manner, while localisation is much more dependent on modelling and representation and should therefore be resistant to a model-free solution. In fact, the opposite turns out to be the case. Furthermore, in reply to the second criticism, it is possible to contrive a scheme for quantifying the performance of an otherwise qualitative system.

Smooth goal-directed locomotor competence

Despite the oversimplification, the room-traversing solution depicted in Figure 1.2 is quite typical of how basic locomotion ought to be implemented according to the synthetic philosophy. This treatment of obstacle avoidance as a relatively simple (albeit fundamental) on-line component of a robot's behavioural repertoire is appealing, in that it allows the execution of other aspects of behaviour to proceed on the assumption that they are underpinned by an independent control system that guarantees collision-free robot movement.

However, it will be shown in Chapter 4 that this appeal is deceptive: while obstacle avoidance may be successfully implemented as a self-reliant module in a distributed control system for a robot that bumbles around at random, the resulting impenetrability of the mechanism tends to preclude the kinds of subtle interplay between competences that appear to be necessary to turn robust locomotion into efficient locomotion when the robot is required to move towards some goal.

It will be argued that there are features of the analytical approaches to locomotor competence design which couple well with the obvious advantages of on-line control typical of the synthetic enterprises. In particular, it will become apparent that it makes sense to try to "fit" the robot's speed to the environment. That is, beyond just avoiding potentially damaging collisions with objects in its path, it should move smoothly in the world, slowing down where there is a great clutter of potential obstacles, and moving more quickly in open spaces. This can be accomplished only to a limited degree without some predictive capability such as the kind of environment representations conventionally the preserve of analytical robotics.

Qualitative map-based localisation competence

In general, *localisation* is the process of determining the robot's position in its environment. It can be accomplished in a number of ways (more fully discussed in Chapter 2). Broadly, these may be classed as either map-based, or map-free. In map-based systems, the robot has some internal representation of the objects or surfaces in its environment, usually including geometric relationships between them. In order to determine where it is, it establishes the relationship between itself and environmental features, and then between these features and their isomorphisms in the map, thereby resolving where it is located on the map, and thus in the environment the map represents.

If its environment contains some locative references (landmarks) that can be referred to from anywhere, then the need for a map vanishes, as the robot can always determine its position relative to the landmarks. Where there are no such perpetually detectable environmental features, map-free localisation normally requires that the robot start from some known position and constantly monitor its own motion; given initial position and displacement in this way, it is possible to compute current position relative to the starting point.

Synthetic approaches to localisation apparently face a dilemma: on the one hand, map-free localisation is intensely quantitative in most of its forms, and requires accurate models of the robot's sensory systems; on the other hand, map-based methods depend by definition on representations of the world. The small number of responses to this problem documented in the literature (e.g., Kuipers and Byun, 1987; Flynn, 1988; Nehmzow *et al.*, 1989; Levitt and Lawton, 1990; Mataric, 1990) adopt a qualitative stance: maps are generally serial or path-based ("strip-like" maps, Tolman, 1948), listing actions to take at critical junctures (landmarks) and thus effectively segmenting space into known routes and "everything else".

In Chapter 5, a strategy is proposed for segmenting the working space of a mobile robot by means of combinations of both active and passive sensors. No rigorous sensor models are used, and space is represented minimally as regions and adjacency relationships between them. Despite the presumed impoverished nature of such a simple topological representation, the resulting qualitative map is of sufficient resolution that the robot can localise itself to some extent anywhere within its operating space, and not just in the vicinity of landmarks.

Furthermore, by subjecting the qualitative maps constructed to a series of statistical tests, the performance of the system under various assumptions is characterised numerically. This provides a basis (common currency) for the quantitative comparison of different localisation systems, both non-metric (qualitative) and metric.

1.3 Thesis epitome: integrating analysis and synthesis

This report documents experiments in the design of a free-ranging robotic vehicle with locomotor and spatial skills. The latter competences are posited to be fundamental to any general mobile agent. No provision has been made for manipulation, and motivational issues are not dealt with (the system's "high-level" objectives are specified by a human operator).

Several observations have been made in the course of this work. It has been found that even for a relatively simple locomotor task (smooth goal-directed locomotion), a representation-free solution may lead to robust but not necessarily efficient behaviour. However, it also appears that localisation by means of a variety of sensor systems can be accomplished *without* extensive sensor models, and by means of statistical tests on

map data it is feasible to characterise the performance of an essentially qualitative system in a quantitative way.

These results suggest that it is not intuitively obvious which aspects of robot behaviour are well-suited, and which are unsuited to representation versus representation-free implementation. It would seem that elements of both the conventional (analytic) and putatively revolutionary (synthetic) approaches bear further consideration, and might be usefully combined ("ideological adultery", Malcolm *et al.*, 1989); in any case, extremist positions are usually unwarranted.

Moreover, it should be noted that synthetic robotics, in most of its guises, claims insights from the study of animal intelligence. Although this discipline presents many faces, the claim seems justified as it is common to view intelligent systems as modular or encapsulated, and to approach understanding the organization and development of intelligence by seeking to identify specific subsystems supporting particular adaptive behaviours (Fodor, 1983). Therefore, it is significant that much current work in neuroscience argues that complex animal behaviour has both representational and non-representational aspects (Squire, 1992).

1.4 Dissertation layout

This dissertation is organised as follows: Chapter 2 summarises the general problems of locomotion and localisation, and reviews documented techniques for achieving these competences in mobile robots, with emphasis on the differences between the analytic and synthetic world views; Chapter 3 details the software and hardware of the robot used in the locomotor and spatial experiments; the locomotor competence work is discussed in Chapter 4; Chapter 5 outlines the proposed basis for constructing qualitative maps from sonar, infrared, and audible-acoustic sensors; Chapters 6 and 7 present performance analyses for individual sensors, and combinations of sensors, respectively; and Chapter 8 points to directions for future work.

Chapter 2

Review of Approaches to Robot Locomotion and Spatial Sensing

This chapter expands on the particular problems addressed by the work documented in this dissertation, and reviews the relevant literature. It begins with a more detailed discussion of the kinds of issues that currently seem to divide analysis-oriented from synthesis-oriented robotics initiatives. These include questions of deliberative versus reactive control, explicit modelling and representation versus none, functional versus behavioural system decompositions, centralisation versus distribution of control, and simulation versus implementation. Next, it considers the general problems of locomotion and the apparently disparate aspects of them that are considered relevant from each of the two perspectives. Following this, the problems of localising and navigating are summarised, again to highlight those aspects that each of the two general world views maintain are important. The chapter ends with a synopsis of the discussions of the foregoing sections, and with a strictly parenthetic look at the question of biological inspiration.

2.1 Architectural approaches compared and contrasted

Chapter 1 provided only a superficial comparison of the two broad approaches to designing autonomous agents, which have loosely been grouped under the rubrics *analytic* (the agent analyses its current circumstances in order to decide what to do next) and *synthetic* (the agent's behaviour emerges from the synthesis of a bunch of individual but interacting subcomponents, each tuned to underpin a particular aspect of its outwardly visible behaviour). This section recapitulates the principal differences between the two approaches somewhat more thoroughly. These points are also discussed, to some extent from a different philosophical standpoint, by Malcolm *et al.* (1989), and in great detail by Brooks (1991b).

2.1.1 Representation and planning versus none

The pivotal question in the analytic-synthetic debate is, "in designing control systems for free-ranging robots, what guiding principles lead to implementations that exhibit robust yet flexible behaviour?" There are several issues which are relevant to this debate. The first and arguably most important of these is what Brooks (1991b) calls *situatedness*. Real robots are situated in the world; they should not manipulate abstract symbolic descriptions (world models) of their current circumstances, but rather should respond directly to the events that instantaneously impinge on their sensors. That is, they should react to prevailing conditions rather than deliberate about what to do next based on an internal model assumed to represent those conditions. Traditionally, the converse belief is held, partly because rich representations can, to some extent, compensate for sensory poverty.

Traditional (analytic) viewpoint

The legacy of autonomous robotics' origins in artificial intelligence research is that robot control systems are organised along the same lines as classical AI programs. In particular, control systems are centred on the manipulation of some explicit symbolic representation of the robot's relationship to the world around it. This takes place inside the "brain" of the robot, presumably a computer. There is a well established sequence of operations that lead from sensing to acting. First, the relationship between the robot and its environment is determined by consulting the sensors. This sensor data feeds some objective model of the robot-in-the-world; that is, there is a *perception* subsystem that supplies a *modelling* subsystem with data which the latter tries to construct into an expressive representation of the robot's state.

For example, Goldstein *et al.* (1987) convert distance-to-object measurements obtained from a range sensor into spheres (of radius proportional to their distance and the angular resolution of the range sensor); the boolean union of the sensed set of spheres specifies the three-dimensional geometry of the robot's environment in the notation of constructive solid geometry. However, world models need not be strictly geometric. They can include information about "the work space, objects, properties of objects, relationships among them, events that can occur, and any other relevant information" (Roth-Tabak and Jain, 1989).

Given such a model or environmental description, a *planning* subsystem determines what to do based on the current state of the model. A typical geometric planning operation

is the calculation of a trajectory through areas modelled as empty. Finally, an *execution* subsystem has the task of actually implementing robot actions specified by the planner. Explicit modelling and planning have been integral to most major robotics initiatives (see Moravec, 1983; Giralt *et al.*, 1984; Goto and Stentz, 1987; Kriegman *et al.*, 1987).

Synthetic approaches

The alternative to keeping an internal model and using it to decide what to do next is to rely on finding local constraint in the world such that the correct action of the robot at any time can be determined by this constraint. In effect, this amounts to finding some function of the relationship between the robot and its surroundings that can be sensed, and an appropriate device for sensing it, which governs what robot actions are currently appropriate. Although this sounds rather behaviourist, Kirsh (1991) points out that this idea is not as limited as it seems: such an agent need not be driven to a response by some particular stimulus; instead, it is free to motivate actions of its own in order to control how it gets its information. The task of the synthetic robot designer according to Brooks' (1986a, 1991a, 1991b) way of thinking is to find the necessary local constraint that makes it possible to build robots that can function robustly without internal representations and planning.

Essentially, the synthetic (or nondeliberative, Ferguson, 1992) agent has its beliefs and goals built into its structure by its designer. In Brooks' (1986b) case, this structure is the "wiring" pattern of a set of control modules. Rosenschein and Kaelbling (1986) start with a symbolic behavioural representation (sets of rules describing what action to take under different conditions), but compile it into digital logic to arrive at a fixed, reactive structure.

Moreover, autonomous robots controlled by neural networks have been implemented, demonstrating that it is feasible for a system not only to have behaviour that is free of explicit representations, but that the encoding of beliefs and goals into structure can be automatic. Sekiguchi *et al.* (1989) connect the sensors and actuators of their robot through a hierarchy of networks that allows it to learn sequences of actions in response to inputs from touch, light, and ultrasonic sensors. The learning process is supervised; that is, the robot is presented with combinations of sensory inputs, and the desired responses to them.

Nehmzow *et al.* (1989) use self-organising networks augmented by simple instincts (e.g., "touching obstacles with your whiskers is unpleasant") so that their robot can learn how to control itself without supervision. In the limit, this kind of technique might lead to systems completely devoid of explicit symbolic representations, and which are able to discover for themselves the local constraint in their sensory experience that resolves what action is appropriate at a given time.

Points and counter-points

As remarked in Chapter 1, the disadvantage of using an explicit representation and planning from it is that if the representation is incorrect, then the resulting plan fails to do what it was meant to. Also, it is impossible to guarantee that a given plan will be executed correctly, so that some effort must be expended in watching for and recovering from errors. Obviously, the synthetic idea is very seductive because by reacting mainly to local constraint in the world, a robot thus controlled is always able to act in a way that suits its current circumstances.

However, as Kirsh (1991) points out in great detail, there are tasks for which there simply cannot be constraining stimuli (such as when deciding criteria are out of the robot's sensory range), or in which it seems essential to do some prediction (such as in situations involving other purposeful agents, whose possible future actions may have to be predicted). I have found (see Section 2.2.3) that this is even the case for a simple locomotor task.

Hybrid arrangements

Some workers have acknowledged the possible benefits to be derived from a combined approach. Malcolm's (1987) system for assembling soma cubes is a good example of this. A conventional planner operates at a high level to determine how a set of variously-shaped components must be arranged in order to arrive at a particular composite shape. However, this planner interfaces to a robot manipulator through a virtual machine comprising individual, representation- and planning-free modules; the planner can rely upon these modules to execute certain primitive assembly operations reliably. While not completely eliminating it, the provision of this synthetic substrate greatly simplifies representation and planning at the top level.

Arkin (1989) takes a similar view in mobile robot control. He uses low-level, reactive, stimulus-response kinds of packages called *motor schemas*, but incorporates these within a

higher-level representational framework. These motor schemas are the mobile robot equivalents of the robust, task-achieving assembly modules in Malcolm's (1987) SOMASS system. Correspondingly, they are available to a (thus much simplified) high-level planner as reliable motion primitives (typical self-explanatory examples are "move-to-goal" or "avoid-static-obstacle"). Unfortunately, Brooks (1991b) dismisses these hybrids as having simply delayed the ultimate demise of representation-based control.

2.1.2 Functional versus behavioural decomposition and loci of control

The second most important issue in the division of the two groups is the question of system decomposition. That is, given that any complex system is likely to be most easily implemented in a modular manner, what should these modules comprise, and how should they be structured? This issue Brooks (1991b) calls *emergence*. Now, as Minsky (1963) has long ago pointed out, the discernible behaviour of something *always* emerges from the interaction of its components. Therefore, the difference between the approaches is in the kinds of components that are there to interact, and as a corollary, whether the system has a central or distributed locus of control.

Traditional (analytic) schemes

Traditionally, the control system is essentially a linear chain of transformations (information processing operations) from sensors to actuators. These operations are the perception, modelling, planning, and execution subsystems discussed at the beginning of Section 2.1.1. As Brooks (1991b) puts it, individual elements of the robot's behaviour, like moving through a room without colliding with obstacles, emerge from the interaction of the operations of perception, modelling, planning, execution. There is a single locus of control that passes along the processing chain, so that control is effectively centralised.

Synthetic approaches

The alternative is to synthesise the robot from a number of modules which subserve particular behaviours. As Brooks (1991b) puts it, the imputed intelligent activities like modelling and planning emerge from the interaction of individual elements of behaviour. That is, the simple interaction of move-forward and obstacle avoidance modules gives the robot the appearance of modelling, planning, and executing a collision-free trajectory. Although some arbitration is necessary to decide which component of the robot's behavioural repertoire gets expressed in the actual motion of the robot at any time, at least

the sensory aspects of individual behaviours may occur simultaneously (watching for obstacles can proceed in parallel with monitoring the direction of a target location). The control of the system is therefore effectively distributed.

Points and counter-points

As mentioned in Chapter 1, the disadvantage of the traditional approaches comes from the fact of a chain of operations from sensors through modelling and planning to actuators: any errors along the way will be propagated (modelling errors lead to incorrect plans which lead to incorrect actions). The weakest link in the chain (i.e., the least robust or slowest subsystem) determines the strength of the chain (i.e., the reliability or throughput of the entire system). Furthermore, if any component implementing one of the subsystems along the chain fails completely, then the whole system fails.

Synthetic systems are seductive, because they contain virtual machines for particular aspects of behaviour that can be relied upon to subserve those aspects. For example, it is possible to engineer a fast and reliable obstacle avoidance competence; the functioning of this subsystem is not influenced by an error-prone goal-orientation module (the robot might estimate incorrectly where the goal is, but never has to worry about generating an incorrect path plan that might cause it to collide with another object). Likewise, there is a route for graceful degradation, for if a high-level control component (such as goal-orientation) fails completely, enough of the rest of the control system can persist to allow the safe continuation of more primitive behaviours.

However, while a decomposition by behaviour rather than function has appealing features, the issue of arbitrating between modules which must all express themselves through the shared medium of the robot's motion is not trivial. Furthermore, graceful degradation should not be overplayed: if a very basic module fails in a synthetic system, then other specialised components that were underpinned by it may also fail (e.g., if the obstacle sensors break down, then obstacle avoidance will fail, so even if the robot continues to orient correctly towards its target, nothing stops it from crashing into an incidental obstacle).

Hybrid arrangements

By implication, the hybrid systems of Malcolm (1987), in assembly, and Arkin (1989), in mobile robotics, offer some mechanism for decomposing systems into both analytic-type components, and synthetic-type components. The software architecture described in

Chapter 3, and implemented on a robot used for locomotor and localisation experiments (Chapter 4, and Chapters 5 to 7, respectively), provides for such a hybrid system structuring.

2.1.3 Simulation versus implementation

Certainly the strongest point that synthetic roboticists like to make, is that only real robots are a suitable vehicle for studying organisational principles, while results derived from computer simulations are of no use. This is what Brooks (1991b) calls *embodiment*. Those who actually synthesise robots as electromechanical entities argue that it is impossible to understand the true problems of building robots without actually doing it. Simulations cannot be trusted until the ideas developed by means of them have been validated in the real world. The fundamental issue is that in building a simulation, problems sometimes need to be solved that would not be problems on a real robot; likewise, it is easy to miss solving some problems that cripple subsequent attempts to implement a real machine.

Points and counter-points

Anyone with experience in robot construction would be hard-pressed to deny that real robots are different from simulations. However, it should be kept in mind that simulations have some role to play. In Chapter 4, for example, a simulation of fitting a robot's speed to the clutter of its environment acts as a scratchpad for demonstrating a principle. In designing complex machines, such as airliners, simulations do not supplant wind tunnel and flight tests, but they do act as a useful search space constraining mechanism. It is a question of the exact role that simulations play. Also, while it is clear that traditional (analytic-type) robotic enterprises often take for granted that simulation results will be applicable on real robots, it is important to remember that many of the analytic exercises criticised by synthetic roboticists also involve the construction of and experimentation with real robots (as before, see Moravec, 1983; Giralt *et al.*, 1984; Goto and Stentz, 1987; Kriegman *et al.*, 1987).

2.1.4 Summary

There are several basic differences between what have here been called the analytic and synthetic approaches to robot design. These relate to issues of representing versus not representing the world; whether systems should be decomposed functionally or

behaviourally, and thus how control is localised in the system; and finally, whether simulations have any role to play in the design process. The point and counter-point subsections tried to emphasise that the issues are much less clear-cut than some partisans might like to think, and that attempts at combining the approaches might lead to fruitful compromises. The next two sections deal with the problems of locomotion and localisation in general, and are meant to illustrate those aspects of the two problems that each of the world views considers to be relevant. The work documented in this dissertation will be set in perspective where appropriate.

2.2 Engineering a locomotor competence

Locomotion for the purposes of this discussion is meant to designate "getting around in a non-destructive manner", rather than just "moving". Thus, there are two main problems that are considered to fall under the locomotion heading: these are generating movement through space itself, but also avoiding potentially mutually damaging predicaments with other things in space (obstacle avoidance). Because of the differences in the way the analytic and synthetic approaches would decompose movement plus obstacle avoidance as locomotion, they cannot be said to be different attempts at solving the same problem. Rather, the two world views consider *different* problems as being the ones to solve in order to get locomotion. This section examines these apparent differences. It is not intended to be a detailed review of approaches to any particular subproblem of locomotion, as some of these have a spawned quite an extensive literature. The coverage attempts merely to situate the subsequent discussion.

2.2.1 Analytic approaches

Typically, the analytic approaches decompose the robust locomotion activity into path planning, trajectory generation, and kinematic control. They start with some representation of the space the robot lives in, and the specification of a destination position within that space. Depending on the representation, different methods are available for determining a collision-free route. Generally, routes are planned on the assumption that the geometry of the robot's environment will be relatively static. From the planned route, the trajectory generation phase is the computation of the parameters of lines and curves that the robot would be able to follow, given a faithful model of its

kinematics. A motion control module then tries to implement the generated trajectory (see also Figure 1.1).

Representations of space and path planning

There are many data structures that may be used for representating space, each more or less suited to particular kinds of manipulations (summarised in McKerrow, 1991). The fundamental role of the spatial model is to specify parts of space that are occupied or parts of space that are empty (or both). Then, the path planning problem is to determine how the robot must move in order to stay within the free-space regions.

For example, the robot's operating space can be characterised by a two- or three-dimensional Cartesian coordinate system, anchored at some arbitrary point. Then, the robot's environment might be described by the positions and dimensions of objects, with the assumption that wherever there are no objects, the space is empty. Lozano-Pérez's (1983) configuration space path planning acts on such a representation. Configuration space is a space with as many dimensions as the robot's position and orientation have degrees of freedom. Configurations of the robot that are not allowed because of the presence of obstacles in real space are marked as occupied in configuration space, and he develops algorithms for computing collision-free paths through this space.

Brooks (1983, writing prior to his reincarnation as a synthetic roboticist) develops instead a representation that encodes only free space, since it is this that is of direct interest in determining where it is feasible to go. From a description of the world consisting of the parameters of "cones" of unoccupied space, it is possible to derive safe routes much more efficiently provided there is a sufficient amount of open space (the algorithms do not always work in areas of high obstacle clutter).

Moravec and Elfes (1985) represent space as a two-dimensional pattern of squares called an occupancy grid (discussed further in Section 2.3.2). By repeated sonar scanning, they compute the likelihood that each of the squares in the grid is occupied. For example, if emitting an ultrasonic pulse straight ahead results in a reflection from something 2 m away, then they assume that squares in the grid within 2 m in front of the robot are empty, but that a square 2 m away has a high likelihood of being occupied. Once such a grid has been constructed, the robot plans paths that take it only through grid squares very likely not to be occupied.

Trajectory generation

Trajectory generation is the derivation of trajectories that the robot can feasibly follow, given its geometry, and usually respecting some specified curvature-smoothness constraints (this improves the chances of success of dead-reckoning-based motion control, see below). Although trajectory generation may be incorporated within path planning, this is not always so. For example, Nelson and Cox (1988) call this part of the process "local path planning"; here, the spatial specification of a path to be followed is specified by a global path planner, and is converted by the local path planner into the detailed temporal path data necessary to cause the robot to follow the path smoothly and at some specified speed.

Motion control

Finally, there is the problem of getting the real robot to follow the generated trajectory. This requires a kinematic model (see e.g., Appendix B) of the robot and what Crowley (1989a) calls a "vehicle controller", which is the mobile robot analogue to the "arm controller" on a robot arm (i.e., some mechanism for running the motors of the arm's joints so they achieve a visibly coordinated motion). Generally, this problem is attacked by equipping the robot with devices for sensing its actual motion relative to the world (often displacement measuring devices attached to its wheels). For instance, if the robot is to travel in a straight line for some distance, its displacement sensors are monitored while its drive and steering motors are actuated; if there is a deviation to one side of straight ahead or the other, the steering compensates accordingly. In fact, this is already a basic form of navigation, called dead reckoning.

2.2.2 Synthetic approaches

The synthetic decomposition of robust locomotion is substantially different. In fact, the strategy is very much like the example quoted in Chapter 1. Moving around robustly only requires two modules, probably structured as in Figure 2.1. One simply causes the robot to move forwards. The other is able to sense obstacles in the robot's path, and if one is encountered, suppresses forward propulsion and steers the vehicle away from collision with the obstacle.

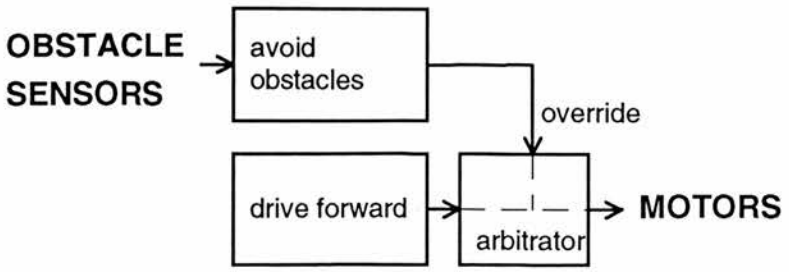

Figure 2.1. Typical subsumption architecture for basic motion control

Of course, this system only implements random driving. It would have to be equipped with some additional directional motivator like the goal-orientation layer in the example in Chapter 1 (see Figure 1.2). This entirely on-line or reactive form of control system typifies the synthetic or behaviour-based approach to locomotion (Brooks, 1986a).

Similarly, the controllers of Sekiguchi *et al.* (1989) and Nehmzow *et al.* (1989) are characterised by simple locomotor components, except that the motor actuation patterns for driving forward versus turning left or right to steer away from an obstacle are learned by neural networks. In the case of the former, this is by means of explicit instruction (e.g., the robot is shown an object on its left sensors, and told that the thing to do is turn right); in the latter, skill acquision is mediated by a simple instinct which tells the robot that touching things with its whiskers is not good, so that it learns to turn left if its right whisker is touched, and vice versa.

2.2.3 Discussion

The two broad approaches to locomotion highlight the major points of architectural difference discussed in Section 2.1. The analytic approaches depend on representation and planning, and thus suffer from the problems mentioned earlier in terms of inroads for error. Locomotion in the synthetic style is not susceptible to incorrect representations or badly executed plans, and its tight coupling between sensing and action make it potentially very fast. On the other hand, significant advantages are conferred on an agent that has some form of representation. It be will shown in Chapter 4, for example, that even basic goal-directed locomotion can be done much more efficiently with the predictive power that a map would provide.

Furthermore, there also exists potential here for hybrid schemes. For example, Borenstein and Koren (1990) describe obstacle avoidance in real-time based on a combined grid and potential field representation of space. As applied to mobile robots, the

idea is that the robot's goal is represented by an attractive force, and obstacles by repulsive forces. The robot moves through space like a charged particle through attractive and repulsive fields, alternately being pulled towards the goal and pushed away from obstacles. It does require a representation of space, but (in this case) the representation is quite simple, and by fast sonar-based updating it can in principle respond to dynamic changes to the environment in real-time (at robot speeds of the order of 0.8 m/s).

Leonard's (1990) objective is to incorporate obstacle avoidance within a dynamic map building process (see Section 2.3.2). Mapped features have both a position estimate and a credibility measure. Unexpected objects in the robot's vicinity are treated as features to be mapped, and avoided during motion, but initially they are assigned little credibility. If the presence of a particular object is ephemeral, then it never becomes an important map feature, unlike persistent objects whose credibility increases through time. Leonard's "directed sensing" bears some of the hallmarks of a synthetic approach (active perception potentially in real-time), all the while offering the advantages of spatial representation.

2.3 Engineering spatial competences

In locomotion, it seems that the two world views are effectively disjunctive in what they consider to be important versus irrelevant issues. The points of difference in spatial competence research lie roughly on either side of a quantitative versus qualitative dividing line, with regard to the treatment of maps and the sensing activities from which they are built, as will be discussed shortly. First, however, it is necessary to note the distinction implicit in this work between *localisation* and *navigation*. Localisation is simply the process of determining current position, and is basically a static operation, except where "active" sensing systems are in use which require that the robot move in order to facilitate a reliable positional fix. Navigation, on the other hand, is the inherently dynamic activity of travelling between a start position and a destination.

Localisation effectively subserves navigation, in that it provides a mechanism for resolving current position, the position from which the path to be navigated starts; it may also be used to fix waypoints along the journey, to correct for possible navigational errors *en route*; and it may again come into play at the end of an excursion in order to confirm that a desired position has in fact been reached. The spatial competence of primary interest in this work (Chapters 5 to 7) is localisation.

Navigation and localisation are well-studied problems, and analysis-based approaches abound. They typically involve geometric maps as models of space and quantitative treatments of the capabilities of sensors, with considerable attention paid to questions of certainty and uncertainty in the robot's estimates of where it is. Synthetic-type approaches to spatial control are quite few, as will be shown shortly, but in all existing forms are based on qualitative treatments of space, and heuristic models of sensing.

2.3.1 Maps versus none

In mobile robotics, most users of the term would accept a simple definition of "map" as some representation within the robot of the relative positions of objects or surfaces in its environment, usually including geometric relationships between them; here "representing" is meant as "corresponding in a such a way that features and relationships in the *represented* space exist as features and isomorphic relationships in the *representation*". The kinds of geometric relationships preserved in a map determine its usefulness for various tasks. Gallistel (1990) recalls Klein's (1926) hierarchy of geometric relationships, ranging from metric to topological: in metric maps, angles and distances between environmental features have faithful isomorphisms in the map; whereas in topological maps, only feature congruence is conserved. In other words, it is possible with a metric map to determine the distance between two objects in the environment it represents without measuring the actual distance, while with a topological map it is only possible to determine whether or not such objects are adjacent.

If *localisation* is the process of determining the robot's position in its environment, then for a map-based system this amounts to establishing the relationship between the robot and environmental features, and then between these features and their isomorphisms in the map; then is it possible to resolve where in the map, and thus in the environment it represents, the robot must be. As mentioned in Chapter 1, it is possible to imagine situations in which a robot's environment contains some locative references (landmarks or beacons) that can be referred to from anywhere, so the "map" need only be a record of the relative positions of the beacons, as the robot can always determine its position relative to the references.

Where there are no such perpetually detectable environmental features, map-free localisation normally requires that the robot start from some known position and constantly monitor its own motion; given initial position and displacement in this way, it is

possible to compute current position relative to the starting point. This is inertial localisation, and requires that the system be equipped with some device for measuring displacement, velocity, or acceleration. Where displacement is measurable directly, determining current location from past location is trivial. Most often, however, inertial navigation centres on an accelerometer, with displacement computed as the double-integral over time of acceleration (a detailed discussion of inertial navigation is provided by Kuritsky and Goldstein, 1983).

The elegance of inertial navigation lies in its conceptual simplicity. However, the brief description given above belies its complexity in terms of implementation. Although inertial positioning systems find extensive application in earth-bound and inter-planetary (even inter-stellar) navigation, at present it seems that the material and computational expense of engineering them argues for simpler approaches to mobile robot localisation. Indeed, the majority of localisation schemes studied are map-based, both in analytic- and synthetic-oriented robotics. Also, although initially very accurate, inertial systems typically suffer from increasing inaccuracy of their position estimates; this is often compensated for by regular recalibrations referenced to stars or terrain features (i.e., maps).

2.3.2 Qualitative versus quantitative maps

In fact, the major difference between the world views is in the nature of the maps used. While the set of all possible mapping schemes is not strictly dimorphic, it is compelling to split mapping into a loose equivalence class on the one hand of approaches that are *quantitative* or metric, as distinct from the class of largely *qualitative*, topological, serial, or path-based representations on the other hand. Quantitative maps are characterised by geometric relationships that faithfully depict the angular and linear geometry of the environment they describe; this is the kind of representation that usually springs to mind when the word "map" is used in normal conversation. In qualitative systems, the map serves simply to partition space into distinguishable segments: topological maps mainly encode regions and adjacency relationships between them; serial or path maps ("strip-like" maps in the terminology of Tolman, 1948; and Tolman *et al.*, 1946) list actions to take at critical junctures, and thus segment space into known routes and "everything else".

Quantitative mapping of environment geometry

The kinds of spatial representation used in the analytic approaches to locomotion (Section 2.2.1) are easily coopted as a basis for estimating its position. For example, a scheme such

as that of Goldstein *et al.* (1987, see Section 2.1.1) allows the three-dimensional geometry of the robot's environment to be deduced from range measurements to object surfaces. Such a model can also be constructed from camera images (e.g., Ayache and Faugeras, 1989). Once such a model has been established (or given it *a priori*), the robot's position at any time can be determined from its current (range- or visual-image) view of its environment. Crowley (1989b) uses a ring of ultrasonic transducers to determine the limits of free space, and thus constructs only a two-dimensional geometric world model.

Quantitative mapping by occupancy grids

Geometric models of the kind described above are notoriously difficult to construct, and matching a current view with the stored model is computationally intensive. The grid-based scheme of Moravec and Elfes (1985) was essentially a response to these problems. As noted in Section 2.2.1, the technique decomposes space into a grid of squares, and an ultrasonic rangefinder is used to determine the likelihood that a particular grid is occupied. Since occluding objects might prevent the occupancy of some squares from being tested, occupancy likelihoods from multiple views are combined (Elfes, 1987); the robot can even be programmed to explore automatically areas about which it lacks information (Elfes, 1991).

Quantitative mapping by features

The disadvantage of grid-based maps is that they represent, by definition, the robot's complete environment. While well-suited for finding obstacle-free paths, localisation by means of them is computationally expensive and time-consuming: in order to estimate where it is, the robot must scan its environment, construct an occupancy grid relative to its current point of view, and then cross-correlate this with, for example, a reference grid or a geometric map. Leonard (1990) remarks that the major problem with occupancy grids for localisation is that each square in the grid is associated only with a single number (occupancy probability); there is thus no basis for assessing the correspondance between squares other than by cross-correlation with a previously acquired map.

The logical alternative is to map features. Feature-based localisation is a procedure through which a system determines its egomotion by sensing the relative displacements of other things in its environment. Hallam's (1985) sonar-based approach for undersea robots is able to compute position by tracking motion in three dimensions relative to some number of both fixed and moving environmental constituents; in this case, the tracked

features are different kinds of ultrasonic specular reflections (Hallam, 1986). No *a priori* map is needed, although the parameters of tracked features once position estimation has commenced constitute an *a posteriori* map. Leonard's (1990; also Leonard *et al.*, 1992) system works on land, and tracks the subset of specular reflection types that are applicable to operating in two rather than three dimensions.

A pervasive attribute of quantitative feature-based navigation schemes is the attempt to quantify explicitly the uncertainty of feature position estimates (Durrant-Whyte, 1988a). Robust methods for propagating uncertainties across successive localisation attempts are used, principally based on extended Kalman filter (EKF) techniques. Smith *et al.* (1986, 1988) describe a map structure which contains estimates of the positions and relationships between objects in the robot's environment, including uncertainties. Procedures for building, interpreting, and altering the map are developed in the mathematically sound EKF framework. Brooks (1982) earlier proposed a symbolic method for uncertainty analysis, but most current quantitative localisation work (e.g., Hallam, 1985; Leonard,1990) is Kalman filter-based.

Qualitative mapping

The kinds of mapping schemes outlined above are considered inappropriate to the synthetic approach. Since the philosophy is that a robot's actions should be situation-determined, only the relationship between the robot and presently distinguishable environmental features is an appropriate basis for localisation. It is acknowledged that some concession is necessary in the direction of representation: maps are indisputably representations, and the need for maps is inescapable if a robot is to perform any spatially-constraining task (i.e., where random walking or following a wall or sensory gradient are not sufficient) in any extended environment that does not offer perpetually detectable landmarks. However, only (qualitative) feature-based maps with at best topological connectivity are admitted.

Three such schemes have been developed. Connell's (1989) robot is founded on a robust wall-following behaviour. Walls are considered to be reliable "pre-existing paths" in the environment. As implemented, it actually uses a fixed rule to decide which way to turn at path junctions. To retrace its steps (return home from some distant location), it need only reverse the turn rule. He suggests replacing the fixed rule scheme by a list of turns made. Then, the robot can choose to turn in any direction at an intersection, and the record of the turns made consititutes a serial map.

Nehmzow's (1990) robot also wall-follows in a maze-like environment. It maps the sequence of convex and concave corners it encounters as it moves about using a self-organising neural network. Mataric's (1990) robot similarly traces fixed boundaries in its environment, and records the sequential relationship between landmarks in an "active-constructive" map. The landmarks are actually combinations of the robot's motion and sensory input (e.g., a corridor landmark is defined by straight motion while detecting objects on both sides of the robot). The map is active in that landmarks are active processes; visiting one is implemented by causing it to send out an invitation to all connected landmarks. These in turn attach their names to the invitation, and propagate to their neighbours. Eventually, the landmark where the robot is currently located will receive a message indicating which landmark it is meant to travel to, and the attached list of nodes the message has passed through tells it how to get there, serially.

2.3.3 Quantitative versus qualitative spatial sensor models

It must be noted that since the objective of analysis-based approaches is to construct quantitative maps, they rely on relatively precise characterisations of sensors used in the mapping process. This is particularly important when disparate sensors are to be used in combination to estimate position, as the ultimate uncertainty of the estimate is a function of the combined uncertainties of individual sensors (Durrant-Whyte, 1988b).

In contrast, this kind of sensor modelling is not done in synthetic robotics. For example, Sarachik (1989) uses stereo cameras to extract the shape of rooms her robot finds itself in, such that the camera calibration parameters are determined automatically, rather than specified in advance. Similarly, Flynn (1988) proposes heuristic rules derived from empirical investigations of sensor characteristics in combining sonar and infrared range data.

2.3.4 Discussion

It is certainly easy to imagine tasks for which quantitative mapping is essential (e.g., directing a ship through treacherous waters with sub-surface obstacles, in the absence of an effective device for detecting these obstacles in advance of colliding with them). Also, the conspicuously few qualitative mapping experiments performed to date are rather limited. Since they are invariably constrained to follow boundaries, much of their potential operating space may remain inaccessible. Nevertheless, the fact of the significant bias in

robot navigation research in favour of quantitative methods, to the distinct disadvantage of qualitative work, may be unjustified. Also, the mathematical analysis associated with quantitative mapping schemes does tend to mark them as computationally demanding.

Furthermore, it is clear that there are advantages to be derived from having explicit sensor models, particularly since this makes it possible to estimate *a priori* how well a system will work in specified circumstances, which is not easily achieved without models (Durrant-Whyte, 1988b). However, it is difficult to deny that obtaining realistic models of sensors (especially cheap and easy to manufacture devices) is not necessarily easy. For example, despite the widespread use of Polaroid ultrasonic modules, it could be argued that many users of these devices have not attempted to model them as carefully or as physically convincingly as, for example, Kuc and Siegel (1987).

The work described in Chapters 5 to 7 endeavours to look at the construction of qualitative maps from raw data from largely unmodelled sensors. It diverges from existing synthetic-qualitative efforts by attempting to borrow some mathematical machinery from the traditionally analytic approaches in order to characterise multi-sensor localisation under these circumstances. Interestingly, it is found that such an approach obtains the most advantage (at least in the particular robot/environment circumstances considered here) from sensors for which there are regions in the operating environment within which the devices do not work. This qualitative segmentation of space into usable and unusable areas for the sensors might be considered undesirable in a quantitative mapping scheme, but is here shown to be directly beneficial (see, e.g., discussion in Section 6.4.3).

Convergence

As mentioned several times in the foregoing pages, there seems to be no reason to shun combined approaches. One such exercise is the primarily vision-based navigation work of Kuipers and his colleagues (Kuipers and Byun, 1987; Kuipers and Levitt, 1988; Lawton *et al.*, 1986; Levitt *et al.*, 1987; and Levitt and Lawton 1990). They describe a four-level hierarchy of spatial descriptions. At the lowest level are the kinds of sensorimotor interactions with the environment that characterise the purely synthetic approaches to navigation. The next level consists of slightly more abstract "procedural behaviours" which essentially relate short-term actions to longer term goals (e.g., "on the way to goal location X, turn left at doorway Y").

The third level is a topological map; it assimilates procedural behaviours, which are defined in terms of the robot's egocentric sensorimotor encounters, into a unified structure that relates previously fragmented paths and places. At the fourth and final level is the metric map, which adds quantity to previously qualitative relationships. For example, at the sensorimotor level the robot could be made to record that it turns left by a certain angle, and that it then drives forward for a certain period of time (or distance). While these quantities would be irrelevant to lowest-level reactive navigation, they can be propagated upwards to the metric map, from which it is then possible to make inferences about spatial relationships not directly measured. Clearly, the accuracy of the metric map is constrained by the accuracy of the low-level metric information obtained from sensorimotor experiences.

2.4 Summary and arguments from biology

This chapter began with a review of the major ideological differences between two disparate approaches to organising robot control. It was suggested that their most important distinguishing feature lies in whether or not the robot should use explicit models of its relationship to its world. An exposition of the problems of locomotion and localisation for mobile robots showed that in the case of the former, the analytic and synthetic world views simply see different problems as the ones to be solved. In the case of the latter, it emerged that some form of map representation seems inescapable, although the synthetic camp admits only maps preserving qualitative (basically topological) spatial relationships, rather than quantitative (metric) ones.

In both cases, it was pointed out that there are arguments in favour of taking a combined approach, and work described in Chapters 4 to 7 will offer further support for this suggestion. To conclude this chapter, it seems appropriate to consider, if only superficially, questions of representation versus non-representation in the biological literature. The motivation for this is that the synthetic approaches often take a biological inspiration "high ground", claiming that their approaches are more similar to the way robust intelligent behaviour might be organised in animals and humans (or at least in insects), and are therefore more likely to succeed.

Reactive or predictive locomotion?

Given the robust way that animals get around their environments, and often at high speeds, it is compelling to think that locomotion must be a reactive, on-line, low-level behaviour. Yet, Gallistel (1990) discusses some maze learning experiments with rats, performed by Carr and Watson in the first decades of this century:

> When the rats were thoroughly accustomed to running the maze in a given orientation and did so rapidly and without error, Carr and Watson removed a segment of the maze [...] thereby shortening many of the pathways the rats had to traverse. On the first several trials following the removal of this segment, the rats ran headlong into the walls at the ends of the shortened pathways, often hitting the wall so hard their body flattened up against the wall and they appeared stunned for some while afterwards. (p. 96)

This is hardly situation-determined behaviour! The rats presumably ran into the walls because they had expectations of how far they were going to be running, and simply set off.

Cognitive maps or rule-based strategies?

The question of whether animals use representations of space that are map-like is hotly debated in the biological literature (see, e.g., O'Keefe and Nadel, 1978). On the one hand, there is evidence that even large, global navigators like whales, dolphins, and porpoises may use path-based strategies like following geomagnetic flux density contours (Klinowska, 1988). On the other hand, Gallistel (1990) documents a very wide spectrum of experiments on many different species that argue not only for topological representations of space, but also for metric ones. Interestingly, the kinds of navigational skills that seem more allied to manipulable metric representations than qualitative maps are seen not just in "higher" animals, but also in insects, such as bees (e.g., Cartright and Collett, 1983).

Summary

There is a debate in biology between representation-based and representation-free models of cognition, not unlike the analysis versus synthesis controversy in robot design. In either field, it seems that aspects of both points of view have a place.

Chapter 3

Implementational Aspects

The preceding chapter reviewed the problems of locomotion and navigation in autonomous mobile robots. It highlighted how these problems have been approached in various robotic enterprises, and pointed to those aspects of them that are relevant to the work discussed in this dissertation. In comparing and contrasting the disparate world views, it was argued that combining elements of both approaches might lead to more effective systems than by either guiding philosophy on its own. Before considering locomotor and spatial capabilities further (in Chapter 4 and Chapters 5 to 7, respectively), this chapter outlines the technological basis for the later discussion.

In particular, it is founded on the acknowledgement that autonomous mobile robotics is not a well established disclipline like, say, aerospace engineering (there is no robotics equivalent to the rigorous process of designing an airplane, for instance), and that therefore, most experimentalists construct their robot control systems largely in software to maximise the ease with which they can be reconfigured. The chapter thus begins by summarising the basic architectural requirements which arise in designing a robot vehicle capable of moving and finding its way around; a software-based system architecture is outlined that has components based on these features, and which is amenable to both analytic- and synthetic-style engineering.

The chapter concludes with a section documenting the technical details and evolution of the prototype robot, mainly in terms of sensory systems for locomotion and localisation. Although these technological aspects are to some degree parenthetic, later discussions will assume awareness of the experimental robot's characteristics. However, where such assumptions are made, backward references to the relevant sections in this chapter are provided, and the essential technical details of the prototype robot are summarised at the end of the chapter.

3.1 System architecture implementation

The term "architecture" can be defined variously. Here, it is meant to denote the set of

recognisably distinct hardware or software components that in concert make it possible for an otherwise inanimate piece of machinery to act purposefully, and the interrelationships between these components. There are several characteristics which make a system architecture suitable for a robot required to have locomotor and spatial skills, assuming a combined analytic- and synthetic-type philosophy is to be followed:

- modular specification of the system by competence; this is the (mainly synthetic) notion that it should be possible to structure the system such that the sensing and action aspects of a particular piece of the robot's global behaviour are encapsulable into a single, coherent unit;

- channels for sensing aspects of the robot's relationship to its environment, and for effecting actions in it; this requirement is almost tautological, but serves as a reminder that the emphasis here is on real robotic devices rather than simulations;

- mechanisms for combining all or part of existing competences in order to derive new capabilities;

- a means for controlling the sequencing of competences through time; this enables the system to perform (essentially analytic-type) planning operations; and

- mechanisms for behaviour modification contingent upon past actions or perceptions; this subserves, among other things, representation-construction activities like mapping.

In order to facilitate the prototyping of robots displaying locomotor and spatial capabilities, a system architecture incorporating features including those listed above was implemented in software. It is based on an object-oriented version of the Pascal computer programming language (Turbo Pascal®), enhanced in a number of ways. The object-oriented basis seems to be naturally suited to implementing the kinds of architectural features outlined above, because some of these requirements are provided for implicitly, and it is a relatively straightforward matter to integrate the rest. In particular:

- "objects" in the object-oriented sense are modular by definition, and already have encapsulation properties, in that they present some unified interface which is independent of their internal structure;

- objects can be created that act as interfaces to physical sensory and motor devices, and the encapsulation properties afford the advantage that changing actual devices need not have any effect on the way the devices look to the rest of the system;

- they have adjustable inheritance properties, in that they can consist of other

modules, with local changes to elements of the modules they inherit (virtual method concept);

- they can exist and operate concurrently, with mechanisms for synchronising them where required, and they can be sequenced by other modules (perhaps performing planning operations); and

- their internal structure comprises both procedures and memory (process and state), and modifying the contents of local memory in response to sensed conditions and past actions can be used as a basis for adjusting the way local procedures execute (and consequently how the robot behaves).

Thus, an object-oriented model (with some extensions) can provide for the kinds of architectural components that would support an approach to engineering autonomous agents that incorporates aspects of both analytic and synthetic robotics. In the sections below, the important elements of the architecture are discussed, with examples of their relevance and use.

3.1.1 Object-oriented programming

Object-oriented languages were developed in order to provide software engineers with a programming tool that afforded more effective data and code structuring, abstraction, modularisation, and reusability facilities than had been previously available. They have three principal characteristics (Borland, 1988):

1. encapsulation, the combination of data with the (exclusive) set of routines that manipulate it into a unit called an "object";

2. inheritance, the ability to define objects as "descendants" of other objects, such that the former implicitly have access to code and data belonging to the latter; and

3. polymorphism, the ability to modify locally some characteristics of inherited code, so as to operate differently on inherited data.

At first glance, it is not difficult to see that the central properties of this particular discipline represent for computer programming some of the basic attributes that it would be convenient to have in system architectures for mobile robots.

3.1.2 Modularity and encapsulation

The concept of modularity is fundamental to and inherent in the object-oriented

philosophy. Each object is a program module that combines state variables and a set of procedures and functions (called "methods") which are the only entities able to operate on this local memory. As objects can be arbitrarily complex virtual machines, it is feasible to create one that is a coherent unit containing all of the operations and storage space necessary to implement a particular robot competence. Objects encapsulating a particular competence can be executed as concurrent processes, interacting where necessary.

For example, consider the following specification of a simplified robust random-walk module; it causes the robot to move around with no set goal, but avoiding any potential collisions by detecting and circumnavigating obstacles (Appendix A explains the syntax):

```
type  { definitions of objects }

RobustLocomotion = object
    procedure Activate;
  end;

{ implementations of object methods }

procedure RobustLocomotion.Activate;
  var Detecting: Sensors;
      Motors: Actuators;
  begin
    Motors.Speed(100);
    repeat
      if Detecting.ForwardIR
        then begin
               if random(2) = 0    { flip a coin to pick a
                                     random direction }
                 then Motors.TurnLeft
                 else Motors.TurnRight;
               Delay(AVOID_TURN_TIME);
             end
        else Motors.Ahead;
    until false;
  end;
```

This definition of a `RobustLocomotion` object specifies that it has only one method, and inherits no data or methods from other objects, although it calls two other objects of types `Sensors` and `Actuators`. These will be discussed in the next section, but in essence, the former is an interface to the infrared emitter/detector pairs used for detecting obstacles (see Section 3.2.3) and the latter to the robot's drive motors (Section 3.2.2). The statement `Detecting.ForwardIR` can be interpreted as sending a message to a `Sensors` interface object, asking it whether the forward-facing detector is currently perceiving the reflection of its associated emitter's light from a

potential obstacle. Similarly, the statement `Motors.Speed(x)` asks an actuator interface object to run the robot's drive motors at speed x, and so on.

When an object of type `RobustLocomotion` is activated, it simply executes an infinite loop checking for IR reflections. If it detects one, it randomly chooses a direction to turn in to avoid the obstacle, asks the `Motors` object to set the chosen direction of turn, and turns for some time determined by the constant `AVOID_TURN_TIME`. If there are no obstacles, it requests that the `Motors` object make the robot travel forwards.

The complete control program for a robot required to display this behaviour might look something like:

```
var MotionCompetence: RobustLocomotion;

begin  { main robot control program }
  MotionCompetence.Activate;
end.
```

That is, the `RobustLocomotion` object is declared as something called `MotionCompetence`, and the control program's only action is to activate this competence, which runs as described above.

3.1.3 Sensors and actuators

Objects can be defined that act as interfaces between physical devices and control programs. In the previous section, two examples were given, namely `Sensors` and `Actuators`. A simplified version of the first of these might be defined as follows:

```
type  { definitions of objects }

Sensors = object
    function ForwardIR : boolean;
  end;

{ implementations of object methods }

function Sensors.ForwardIR : boolean;
  begin
    HardwareOutput(IR_EMITTERS, FORWARD_EMITTER_ACTIVE);
    if HardwareInput(IR_DETECTORS) = FORWARD_DETECTOR_ON
      then ForwardIR := true
      else ForwardIR := false;
    HardwareOutput(IR_EMITTERS, FORWARD_EMITTER_INACTIVE);
  end;
```

This definition says that when an object of type `Sensors` is asked to perform its

ForwardIR method, what it should do is turn on the forward-facing IR emitter, check if the forward-facing detector is sensing anything, and then turn the emitter off. It returns true if there was a detection, and false otherwise. The HardwareOutput procedure and HardwareInput function are meant to be syntactic conveniences for accessing the hardware ports to which the actual devices are connected. Similarly, an Actuator object might be defined as:

```
type  { definitions of objects }

Actuators = object
    procedure Speed (v: integer);
    procedure Ahead;
    procedure TurnLeft;
    procedure TurnRight;
  end;

{ implementations of object methods }

procedure Actuators.Speed (v: integer);
  begin
    HardwareOutput(LEFT_DRIVE_MOTOR_SPEED, v);
    HardwareOutput(RIGHT_DRIVE_MOTOR_SPEED, v);
  end;
procedure Actuators.Ahead;
  begin
    HardwareOutput(LEFT_DRIVE_MOTOR_DIRECTION, FORWARDS);
    HardwareOutput(RIGHT_DRIVE_MOTOR_DIRECTION, FORWARDS);
  end;
procedure Actuators.TurnLeft;
  begin
    HardwareOutput(LEFT_DRIVE_MOTOR_DIRECTION, REVERSE);
    HardwareOutput(RIGHT_DRIVE_MOTOR_DIRECTION, FORWARDS);
  end;
procedure Actuators.TurnRight;
  begin
    HardwareOutput(LEFT_DRIVE_MOTOR_DIRECTION, FORWARDS);
    HardwareOutput(RIGHT_DRIVE_MOTOR_DIRECTION, REVERSE);
  end;
```

Again, this object provides a lucid access to the hardware of the robot's drive motors. For instance, turning left is implemented as switching the robot's left motor to reverse, while the right motor runs forwards. Of course, it also hides the details of the particular hardware technology employed, so that it should be possible to modify the hardware without affecting how it is used by other parts of the system.

3.1.4 Combination and inheritance

Thus far, the advantages of the object-oriented approach have not been obvious. The simple control programs offered in the previous sections, for example, could be

implemented in any reasonable programming language, and modularity and detail hiding are fairly commonplace. The paradigm's most important properties, however, are its inheritence and polymorphism mechanisms.

To illustrate these properties, suppose that the `RobustLocomotion` object is redefined as follows:

```
type  { definitions of objects }

RobustLocomotion = object
    constructor Activate;
    function TurnDirection : integer; virtual;
  end;

{ implementations of object methods }

procedure RobustLocomotion.Activate;
  var Detecting: Sensors;
      Motors: Actuators;
  begin
    Motors.Speed(100);
    repeat
      if Detecting.ForwardIR
        then begin
               if TurnDirection = LEFT
                 then Motors.TurnLeft
                 else Motors.TurnRight;
               Delay(AVOID_TURN_TIME);
             end
        else Motors.Ahead;
    until false;
  end;

function RobustLocomotion.TurnDirection : integer;
  begin
    if random(2) = 0    { flip a coin to pick a
                          random direction }
      then TurnDirection := LEFT
      else TurnDirection := RIGHT;
  end;
```

All that has changed is that the main loop of the `Activate` method now calls another method, `TurnDirection`, which tells it which way to turn on encountering an obstacle. This method just picks a random direction to turn in. Now, however, suppose there is a new object that inherits all of the variables and methods of `RobustLocomotion`, but chooses to redefine one of the methods locally:

```
type  { definitions of objects }

GoalDirectedLocomotion = object (RobustLocomotion)
    function TurnDirection : integer; virtual;
    { Activate is inherited automatically }
  end;
```

```
{ implementations of object methods }

function GoalDirectedLocomotion.TurnDirection : integer;
  var Beacon: Tracking;
  begin
    if Beacon.Direction = LEFT
      then TurnDirection := LEFT
      else TurnDirection := RIGHT;
  end;
```

In a similar manner to the earlier example, the robot's control program is then declared to be:

```
var MotionCompetence: GoalDirectedLocomotion;

begin  { main robot control program }
  MotionCompetence.Activate;
end.
```

These changes have the following semantics. The `RobustLocomotion` object now has two methods: its `Activate` method (main loop) is essentially the same as before, except that it calls another of its methods (`TurnDirection`) in order to decide in which direction to turn. This method, by default, returns a random turn direction. The new object, `GoalDirectedLocomotion`, inherits everything that was defined for `RobustLocomotion`, including both its `Activate` method and its `TurnDirection` method. However, the latter is redefined locally. Thus, `GoalDirectedLocomotion.Activate` works in exactly the same way as `RobustLocomotion.Activate`, except that when it tries to decide which direction to turn in, it calls the new, locally defined `TurnDirection` test.

The new `TurnDirection` behaves differently from the random one inherited by default. Rather than picking an arbitrary direction of turn, it calls a `Tracking` object which is assumed to be able to answer the question, "what is the direction of the beacon I am meant to be heading towards?" by means of its `Direction` method.

This example illustrates two features of the approach: the first is the ease with which it is possible to reuse code from a simpler competence (presumably tested and known to work) in order to construct a more sophisticated one; the second is the "virtual" method mechanism which maximises the reusability of code by making it possible to leave "hooks" in the code that allow later enhancements to redefine only those aspects that actually need to be modified.

This particular example was presented here because it is directly relevant to (and

therefore motivated fully in) Chapter 4, and because it is quite simple. However, there are many other (longer-winded) ways in which the usefulness of this aspect of an object-oriented architecture are revealed. For instance, it will be shown in Chapter 4 that smooth locomotion requires that the robot slow down in areas of dense clutter. A `SmoothGoalDirectedLocomotion` object can easily be created as the descendant of basic `GoalDirectedLocomotion` by redefining the `Activate` method so that when the robot encounters many obstacles per unit time, it sends the `Motors` interface object a message to decelerate (and accelerate when there are relatively few collision detections).

3.1.5 Concurrency

One observation not made explicit in the previous sections was that objects can send other objects messages in order to request information or that some action be performed. The principal modification made by the author to the basic commercial object-oriented programming system is to allow the concurrent operation of objects. As there is only a single processor on which the robot control programs execute, simulated concurrency is implemented by means of a simple scheduling process which maintains a list of activated objects, and arranges for these each to receive an appropriate share of the processor's time.

In practice, some objects execute constantly, while others only perform a particular operation and then terminate. For example, the `Sensors` and `Motors` interface objects are of the latter variety. When a `Sensors` object is asked about the state of a particular hardware detector, say, it finds out, replies, and does no more. The locomotion `Activate` methods, on the other hand, are obviously infinite loops, so that if the system is required to do something else at the same time, these must be made to share the processing resource.

Several scheduling mechanisms have actually been used in the various single and multi-processor robot prototypes (see Section 3.2.2) developed during the course of the work documented in this dissertation. In general, these were based on conventional machine-code-level task context switching, in which the processor interrupts regularly (at high speed), stores the current state of all of its hardware registers (the "context" of the currently executing process), loads the context of another process, and continues; thus, several processes appear to be running concurrently.

This scheme is straightforward to implement and obviates the need to think about

concurrency while programming (context switching is implicitly provided by the system), but is somewhat rigid and inaccessible: for example, suppose that there are two sections of code vying for the processor; the first is very short, perhaps only checking the state of a sensor and doing nothing if the sensor is not active, and only needs to run once per second; the other is perhaps performing a time-intensive calculation. Simply dividing the processor's time equally between the two processes is unsatisfactory, as the first of the two needs to run only for a short time each second.

In fact, experience in developing modular competence-based programs showed that the concurrent modules almost invariably take the form of a small set of instructions that would execute very quickly, nested within an infinite loop; each time a module has its turn running on the multi-tasking processor, it can loop through its particular set of instructions redundantly many times. This can be avoided by inserting instructions in the source code that forces the processor to run the next module at the end of each pass of the loop.

Therefore, the strategy eventually adopted here was to make the context switching explicit in this way, and at source code level. It works as follows: rather than implementing local infinite loops of their own, objects that are meant to stay activated are instead coded as a terminating set of instructions, and a set of parameters that specify how often and for how long these operations are to be executed, and how critical they are. A relatively simple source-level object scheduling program then maintains a list of active objects, and makes sure (the terminating section of code of) each of these is invoked exactly as required[1].

For example, on the robot described in Section 3.2, there is a bumper mechanism that detects if an actual collision occurs between the robot and something else. This is considered to be a catastrophic event, and the object that handles it is defined as follows:

```
type  { definitions of objects }

Collision = object
    alive: boolean;
    constructor Activate;
    procedure Loop; virtual;
    destructor Terminate;
```

[1]This is implemented as a time-stamped list of object methods to invoke. If the scheduler detects that it has "gotten behind", that is, if the system is ever overloaded so that it fails to activate an object by the time it was supposed to, then it shuts down the robot for safety. In practice, this never occurred.

```
  end;

{ implementations of object methods }

constructor Collision.Activate;
  begin
    alive := true;
    ScheduleObject(3, 10000, Self.Loop);
  end;

procedure Collision.Loop;
  var Bumper: Sensors;
      Motors: Actuators;
  begin
    if not alive
      then Exit;
    if Bumper.LeftContact or Bumper.RightContact
      then begin
             Motors.Speed(0);
             HaltSystem;
           end;
    ScheduleObject(3, 10000, Self.Loop);
  end;

destructor Collision.Terminate;
  begin
    alive := false;
  end;
```

The robot's control program would then contain something like:

```
...
var CollisionDetection: Collision;
...
procedure ScheduleObject (starttime, priority: integer;
                          code: procedure_pointer);
  { adds a new object to the list of things to do,
    set to start within "starttime" milliseconds }
...
procedure Scheduler;
  { runs through the list calling the appropriate
    procedure as specified }
...

begin  { main robot control program }
  CollisionDetection.Activate;
...
  Scheduler;
end.
```

Thus, when the main program starts, the collision detection module is activated. Its `Activate` method sets a boolean flag to true, and has the object's main "loop"

code scheduled as a task that will start in 3 milliseconds, and which has a high priority[2]. When the `Scheduler` eventually runs the `CollisionDetection.Loop` method, the left and right bumper switches (Section 3.2.3) will be tested for contacts. If one has occurred, the robot's motors will be stopped, and the system will shut down. Otherwise, the loop requests that the `Scheduler` run it again in 3 milliseconds.

Thus, the collision detection process executes for exactly as long it needs to, and only as often as necessary. Also, if its `Terminate` method is called, the boolean flag asserted by the `Activate` method is set to false. This has the effect that when next the `Loop` method is called, it does not actually do the collision test, and does not add itself onto the scheduler's list; in other words, it terminates.

This approach to multi-tasking is therefore both transparent and flexible, and is an effective basis for sharing resources between modules which have dissimilar temporal processing requirements.

3.1.6 Control mechanisms and behaviour modification

The code for objects is implemented in a programming language that is sufficiently powerful to specify complex operations. Furthermore, objects can be activated or terminated by other objects cyclically, or just once in response to certain specified conditions. The architecture proposed here thus allows for the concurrent effectuation of competences as well as the stringing together of segments of robot behaviour in arbitrary temporal sequences. As a simple example of this, consider the following modified `RobustLocomotion` competence:

```
type  { definitions of objects }

RobustLocomotion = object
    alive: boolean;
    constructor Activate;
    procedure Loop;
    function TurnDirection : integer; virtual;
    destructor Terminate;
  end;

{ implementations of object methods }

constructor RobustLocomotion.Activate;
  begin
```

---

[2]This is used as follows: if the timing is such that two methods end up with the same start time, the one with the higher priority (ranging between -32768 and 32767) is called first.

```
    alive := true;
    ScheduleObject(10, 1000, Self.Loop);
  end;

procedure RobustLocomotion.Loop;
  var Detecting: Sensors;
      Motors: Actuators;
  begin
    if not alive
      then Exit;
    if Detecting.ForwardIR
      then begin
             if TurnDirection = LEFT
               then Motors.TurnLeft
               else Motors.TurnRight;
             Delay(AVOID_TURN_TIME);
           end
      else Motors.Ahead;
    ScheduleObject(10, 1000, Self.Loop)
  end;

function RobustLocomotion.TurnDirection : integer;
  begin
    if random(2) = 0   { flip a coin to pick a direction }
      then TurnDirection := LEFT
      else TurnDirection := RIGHT;
  end;

destructor RobustLocomotion.Terminate;
  begin
    alive := false;
  end;
```

Suppose that `GoalDirectedLocomotion` is defined exactly as before (i.e., inherits everything from `RobustLocomotion`, but redefines `TurnDirection`). Then, there can be a control program like the following:

```
var RandomWalk: RobustLocomotion;
    HeadForBeacon: GoalDirectedLocomotion;
    Beacon: Sensors;
    Motors: Actuators;

begin  { main robot control program }
  repeat
    if Beacon.Detected
      then begin
             Motors.Speed(0);
             RandomWalk.Terminate;
             HeadForBeacon.Activate;
             Motors.Speed(100);
           end
      else begin
             Motors.Speed(0);
             HeadForBeacon.Terminate;
             RandomWalk.Activate;
```

```
            Motors.Speed(100);
          end
  until false;
end.
```

This simple program would alternate between heading towards a beacon if one is visible, and just doing a random walk if none is seen. This effect can naturally be realised in different ways; however, this one illustrates a way of implementing two competences which can then be treated essentially as subroutines in a programming language, even though these routines actually represent complete, task-achieving virtual machines that sense external conditions and perform appropriate actions.

Moreover, the architecture is "open" to external access. That is, a control program running on the robot determines how it behaves, but it is possible to modify this behaviour from the outside. For example, it is easy to superimpose a human operator interface on top of a control program that already implements robust locomotion. With `RobustLocomotion` defined as in this section, suppose that the main control program is specified as follows:

```
var CollisionAvoidance: RobustLocomotion;
    Motors: Actuators;
    Operator: SerialPort;

begin  { main robot control program }
  CollisionAvoidance.Activate;
  repeat
    case Operator.Input of
        'l': Motors.LeftTurn;
        'r': Motors.RightTurn;
        's': Motors.Speed(0);  { stop }
        'g': begin
               Motors.Ahead;
               Motors.Speed(100);
             end;
        'b': begin
               Motors.Reverse;
               Motors.Speed(50);
             end;
      end;
  until false;
end.
```

This program implements a loop that reads commands from the robot's serial port, awaiting instructions from an operator that dictate how the robot should move. However, the `RobustLocomotion` competence has been activated. Therefore, should the operator accidentally or deliberately steer badly, no harm will come to the robot or anything in its path, as the collision detection and avoidance system will

temporarily override the operator under these circumstances.

Finally, it should be noted that since objects have local variables, this local memory can be used to determine operational parameters; these can be adjusted automatically in response to sensed conditions or the results of past actions in order to modify details of the way a particular competence works. For example, the predictive speed profiling discussed in Section 4.3.4 could be implemented by means of a `SmoothGoalDirectedLocomotion` object descending from the basic `RobustLocomotion` and `GoalDirectedLocomotion` competences, but which has a local array in which it has stored the densities of obstacles encountered on previous journeys along a particular path; by looking ahead in this array, it can set the robot to travel at a speed that optimises the smoothness of its trajectory (see Chapter 4 generally). Moreover, this density array itself could be a method that is overridden if a sophisticated spatial mapping mechanism is provided.

3.1.7 Summary

The system architecture proposed here has all of the usual advantages of modular structures, plus several additional features. It allows modules to inherit the processes and states of other modules so that more complex things can be built out of simpler ones; moreover, particular parts of inherited objects can be modulated locally, while preserving other aspects. Thus, the approach offers a very powerful combinatorial mechanism. By means of a flexible, source code level multi-tasking mechanism, robot behaviour can be composed both from concurrent interacting modules, as well as from sequences of actions.

In this way, it appears to supply all of the conceptual architectural needs of a control system for prototyping the design of locomotor and spatial competences. As a basis for the experiments discussed in Chapters 4 to 7, it was found to be both reliable and easy to use, at least as implemented on the particular configuration of robot hardware to be documented in the rest of this chapter.

3.2 Prototype robot vehicle

A number of prototype mobile robots were constructed during the course of the work described in this report. In the main, their technical evolution progressed from basic locomotion towards increased spatial abilities. This section discusses the technical

development, starting from a summary of the basic components that typically comprise autonomous mobile robots, and continuing through the principal mechanical details to descriptions of the particular sensory devices implemented to sustain locomotor and spatial skills. A concise summary of the technical details of the robot can be found in Section 3.3.

3.2.1 Point of Departure

Most roboticists would agree that the following components, together with some guiding assumptions, form the essential basis for an autonomous mobile robot, as distinct from other robotic mechanisms:

- a central structural element, onto which all remaining components are mounted, and which therefore binds the parts of the system into a coherent whole that moves as a single entity;
- some means of propulsion; and
- a device or devices able to regulate the propulsion component such that the vehicle can be made to move in space in a controlled way, possibly requiring
- transducers capable of measuring physical quantities such as light intensity, from which might be deduced the relationship between the vehicle and the environment it is moving around in.

Typically, a mobile robot is a platform or chassis with wheels either driven (actuated by motors), or passive (casters), whose motion is controlled by one or more digital computers, and which is equipped with sensors that make it possible to generate the kinds of information used by programs running on the controlling computers to decide what motor actuations will achieve some desired behaviour of the system as a whole. The robot becomes "autonomous" if these sensing, actuation, and control components are entirely contained within it, that is, if the sensory information generated is processed on-board into appropriate motor commands, without the intervention of an external controlling influence such as a human operator. These assumptions and this configuration of components have been the point of departure for all of the robots that were constructed and experimented with in the work described in this document.

3.2.2 Basic vehicle configuration

First Steps

The first prototype devices were constructed of Lego Technic™, as the large variety of structural components available offered a basis for fast assembly and reconfiguration of chassis designs, and the motors and gears of the Lego Technic family were a compatible and convenient means of propulsion for prototype vehicles. I designed a simple control processor with easily configurable input and output channels that allowed for some latitude in choosing the types and combinations of sensors and actuators used. The modularity of the Lego permitted the mechanical arrangement of these components to be altered quite easily. The electro-mechanical facilities thus provided formed the basis for a larger research and teaching enterprise (see Donnett and Smithers, 1990).

A large number of student projects have been, and continue to be, carried out with this technology. Its greatest virtue is the ease with which mechanical configurations and sensor arrangements can be adjusted. The variety of vehicles constructed to date, and attempts to control them, have taught that the physical form of vehicles bears very heavily on their control. This relationship between morphology and control remains fertile ground for experimentation. However, the work here concentrates instead on questions of locomotor and spatial competence; principally for the latter, problems of sensor information uncertainty required the augmentation of sensory and motor capabilities of the first prototypes quite considerably. Therefore, a particular prototype mechanical configuration was selected that was found to be easy to control (see next section) in the Lego experiments; an identically configured version was constructed of solid pieces rather than Lego bricks, and with more powerful motors, as the Lego vehicles were mechanically somewhat fragile. A photograph of the final prototype is given in Figure 3.1.

Chassis and Mechanics

The chassis of the robot consists primarily of a tin box housing the electrically noisiest components in order to shield the processing electronics from them. The main processor is mounted on top of the tin box, and the robot's motors are attached on the bottom. The robot's body is about 40 cm long by 25 cm wide.

The mechanical arrangement that was found to be the easiest to control consists of two driven wheels with collinear axles, and a third passive wheel (caster) mounted on a

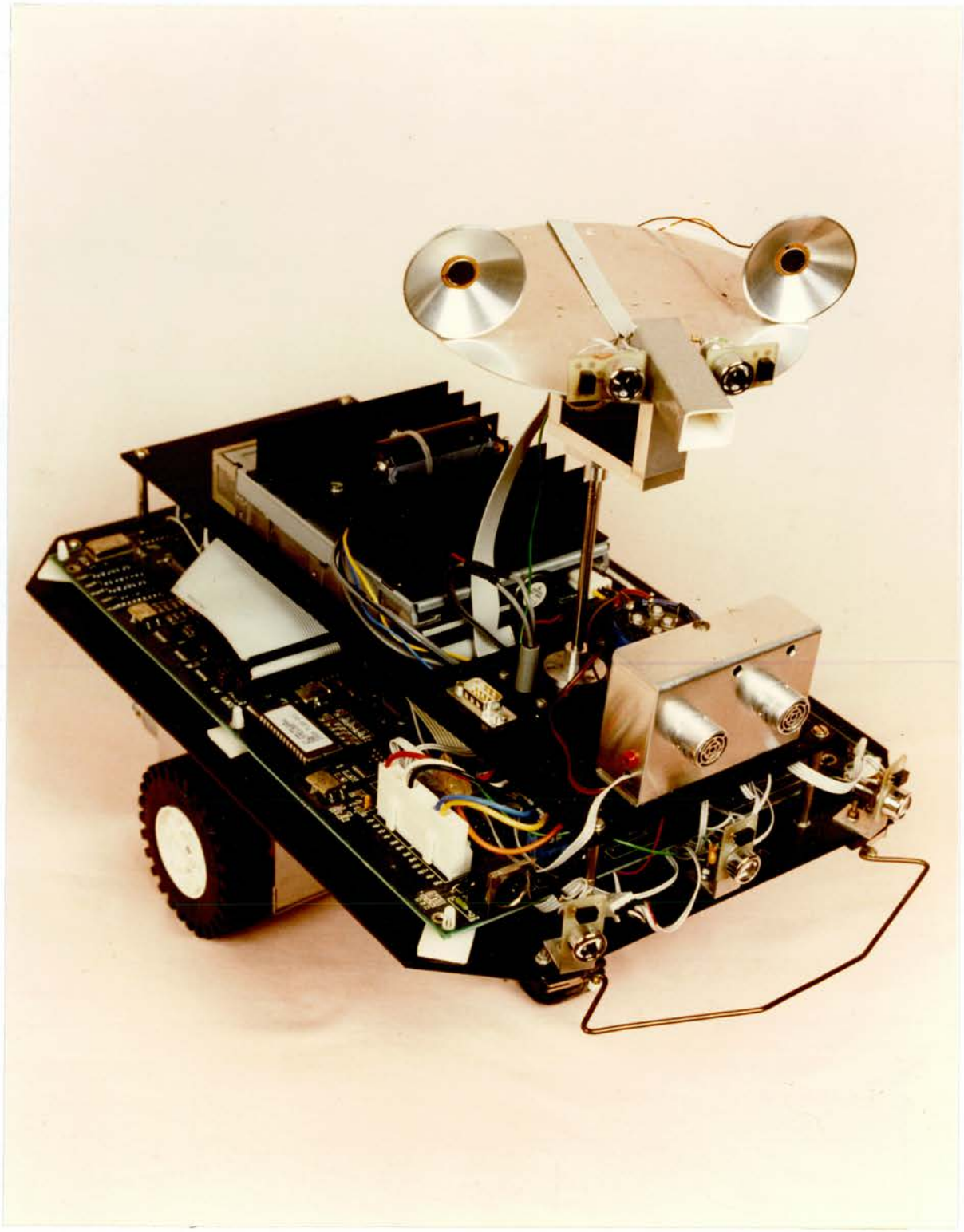

Figure 3.1(a). Photograph of prototype robot vehicle (courtesy of Lawrence Winram)

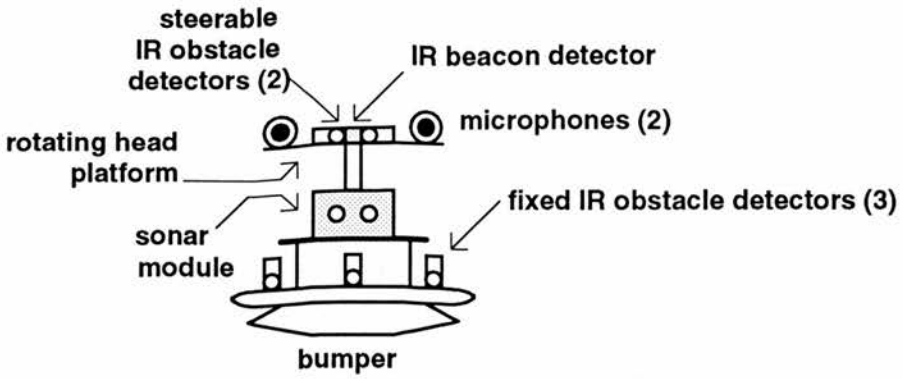

Figure 3.1(b). Legend for photograph, indicating visible sensors from a frontal viewpoint; these devices will be explained in the next several pages of text

line bisecting and perpendicular to the axle-line of the driven wheels (see Figure 3.2). The ease of control obtains for the following reasons: by rotating the driven wheels at the same speed forwards or backwards, the robot moves in a straight line forwards or backwards, respectively; by rotating the wheels at different speeds, the robot turns along a curve; and by rotating the wheels at the same speed but in opposite directions, the robot can (in principle) rotate on the spot (about a vertical axis bisecting the driven wheel axle-line). The robot can reach any (unoccupied) position in its environment simply by executing a sequence of rotations-on-the-spot and straight-line forward displacements. Other mechanical arrangements such as the typical automobile-type four wheel design (two on a driven axle and two on a steered axle) require sophisticated movement strategies for getting into and out of physically restricted areas (e.g., parallel parking) which would be directly accessible to robots configured as in Figure 3.2.

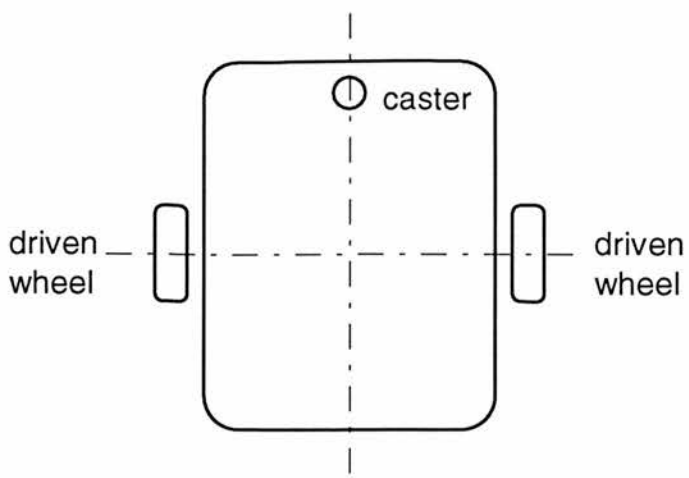

Figure 3.2. Basic mechanical arrangement

To achieve the essential decoupling between the two driven wheels as easily as

possible, each is rotated by its own DC motor and gearbox. The two motors are controlled separately: the polarity of their power supply is reversible, allowing them to turn clockwise or counterclockwise; and the amplitude of the supply voltage is specified by an 8-bit number, so that each motor effectively has 256 forward and backwards speeds, with speed zero set to be "off". The robot's top speed is about 1 m/s, although in practice it was never run at speeds greater than 40 cm/s.

In order to allow some of the sensing devices to be pointed in different directions, the chassis is fitted with a rotating "head" platform. This platform is an aluminium disk 12 cm in diameter, able to rotate from 100° counter-clockwise to 100° clockwise of straight ahead. The head is directly driven by a model aircraft servomotor, so that head position is continuously adjustable in principle. However, the head orientation controller is implemented such that an 8-bit value specifies desired direction, so that resulting angular positioning resolution is about 0.78° (200° divided by 256).

Control Electronics

Although initially based only on a network of small 8-bit 6502 processors, the later experiments in spatial mapping required additional memory and processing speed. Accordingly, a standard IBM PC/AT motherboard was used as a "master" controller, with additional processors dedicated to specific functions like positioning the head platform, manipulating sensor input, and controlling motor speed. Experimental robot control programs can either be developed on-board, or downline-loaded via a serial link from some external program development facility. In general, the latter approach was preferable, with programs written, compiled, and tested initially on a separate IBM PC/AT, and finally transferred across the serial link to the robot for execution.

It is the sensory systems installed in the robot that have undergone the most evolution, mainly in response to requirements generated in progressing from locomotion to localisation, but also for technical reasons. The operational characteristics of the various sensors are described below, categorised by the competences they were primarily intended to support.

3.2.3 Sensory systems for robust locomotion

The principal role of sensors in robust locomotion is to detect the imminent approach of obstacles, so that the robot can stop before it collides with them. Furthermore, since all sensors are liable to fail under some conditions, it is useful also to be able to sense obstacles by detecting actual collisions.

Actual Physical Contact

Actual collisions are detected by a pair of contact switches mounted at the left and right ends of a bumper protecting the front of the robot. The mechanical configuration (see Figure 3.3a) of the bumper is such that if the bumper touches an obstacle to the left of its centre only, the left contact switch connects; conversely, touches on the right close the right contact switch. Head-on collisions result in both switches closing at once. The vehicle's processors can therefore distinguish between three different kinds of front-end collisions (Figure 3.3b).

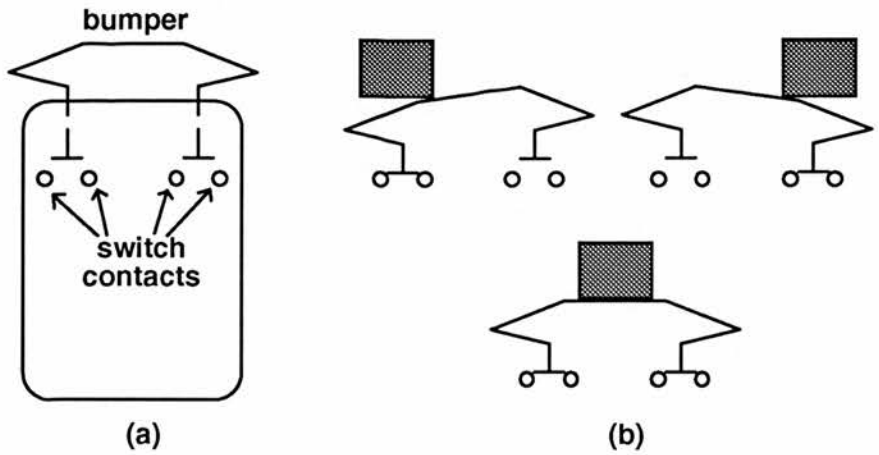

Figure 3.3. (a) bumper and switch arrangement, and (b) bumper and switch contacts for three types of collision

Imminent Physical Contact

Simple contact switches or whiskers can be used as the sole basis for collision detection and avoidance. However, this requires that the robot be moving very slowly so that collisions cause no damage to either robot or obstacle. The vehicle was therefore equipped with six pairs of infrared light emitters and detectors, able to detect objects without physical contact. The emitters are operated at relatively high power, so that their beams can be reflected by objects in the robot's environment and detected up to 2 m away (provided the objects have a surface that reflects IR light, and are oriented appropriately with respect to the emitter/detector pair). The arrangement of a pair is given in Figure 3.4.

For each pair, the emitter is based on a light-emitting diode provided with focusing optics that shape its light into a 6° cone. The emitter is pulse-modulated by a 1 kHz oscillator, with a 10% duty cycle. This low duty cycle allows the diode to be operated at high power; the 1 kHz modulating frequency itself serves to distinguish the robot's own infrared emissions from ambient infrared energy.

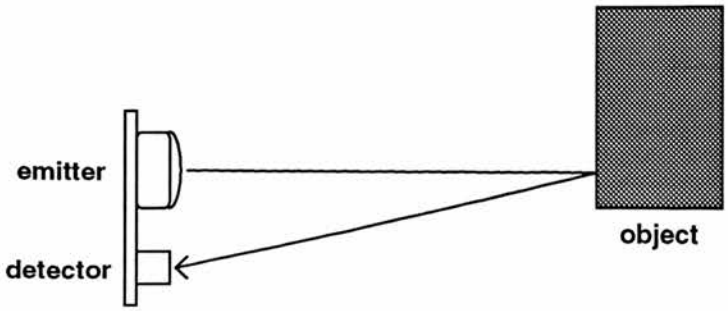

Figure 3.4. IR emitter/detector pair (approximately actual size)

The detector consists of a further diode, operated photoconductively. The signal detected by the diode is amplified, and subjected to high-pass filtering. The filter is designed to pass the 1 kHz modulated signal of the paired emitter, while blocking 50 Hz modulated infrared energy emanating from indoor lighting such as fluorescent tubes. The detected signal is thresholded by a comparator whose reference level is set manually by a potentiometer. This setting essentially adjusts the sensitivity of the device. The output of the comparator is available to the vehicle's processors as a binary detect/no-detect condition, and provides a basis for collision avoidance: with the IR pairs facing in the direction of vehicle travel, obstacles to the robot's progress in that direction will reflect the emitted light, and detecting these reflections signals a condition in which the robot should execute some action to avoid a potential collision.

It will be shown in Chapter 4 that simple obstacle detectors and appropriate avoidance manoeuvres can lead to robust locomotion (i.e., in which the robot never causes damage to itself or any other object in its vicinity), but not necessarily efficient locomotion when it is required to be moving towards a specified goal location. The system's locomotor performance was shown to be improved by the longer-range sensing abilities of the ultrasonic module, but as this system was primarily introduced to support spatial competences, its presentation is deferred until Section 3.2.5.

3.2.4 Sensory systems for tropistic control

The simplest way to increment a mobile robot from random walking to purposeful locomotion is to provide it with some external directional reference, and an associated directionally-sensitive form of transduction. It can then be programmed to operate tropistically, that is, to keep orienting itself so as to point towards (or away from) the external reference.

Acoustic system

An early proposal was to install an acoustic system that would allow the direction and distance of sound sources to be estimated. The vehicle was therefore provided with two electret condenser microphones mounted 12 cm apart on the rotating "head" platform. In principle, the phase and intensity disparities between the two microphone signals can be the basis for directional information, and measured intensity at one microphone or the other is a function of the distance of the source. Furthermore, other characteristics of the sound signal can be used to identify its source (such as the sound's spectral signature). A circuit was constructed in which the signal from each of the microphones is band-pass filtered and amplified (with passband extending from approximately 1 kHz to 2 kHz), and then fed to three parallel stages of processing, structured as in Figure 3.5.

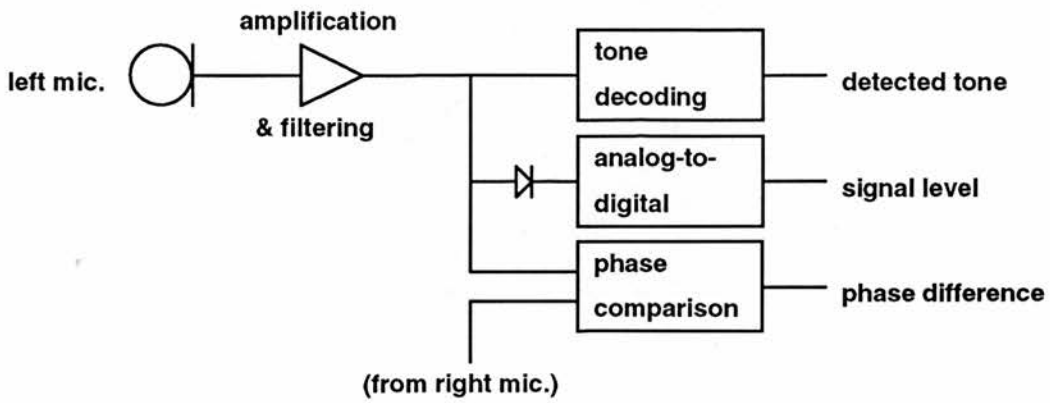

Figure 3.5. Acoustic subsystems

Tone decoders

The first subsystem is essentially a DTMF (Dual-Tone Multiple Frequency) tone decoder. It is provided so that acoustic beacons that might be used to guide the robot can be distinguished easily from other sounds. The principle of the DTMF system is as follows: eight tones, ranging from approximately 800 Hz to 1700 Hz are grouped into two sets of four (one set comprising the lower four tones, the other set the higher four tones). By defining a "DTMF-tone" to be a pair of tones such that one is drawn from the lower tone group, and one from the higher, sixteen distinct DTMF pairs are produced. As "recognising" a DTMF pair requires that both of its tones be distinguished at once, the recognition process is thus less prone to error than one based on listening for a tone at a single frequency. While the single-tone scheme could be made equally robust by maintaining a large separation between beacon identification tones, the DTMF system makes it possible to have sixteen different signatures in a

relatively small piece of the audio spectrum; thus, the passband of input filters can be kept quite narrow. The DTMF approach forms the basis for tone dialling in most modern telephone systems.

Signal Amplitude Detection

The second path of the filtered ear signal is to a diode detector and analogue-to-digital converter. The digitised version of the signal is offered to the control processor, which can operate on it in any desired manner. For example, the amplitude of a sound being received should be inversely proportional to the square of the distance of its source. Thus, given an installed or learned reference level, it should be possible to estimate the distance of a sound's source in absolute terms. However, this subsystem is not directly relevant to tropistic control, which essentially requires only directional sensitivity.

Binaural phase difference measurement

The third path of the signal from each microphone is through a comparator that squares the signal; the two squarewaves are subsequently exclusive-ORed, and the output of the XOR gate is integrated and passed to an analogue-to-digital converter. The effect of this sequence of transformations is as follows: the XOR of two squared signals in phase is constantly logic 0; the XOR of two signals 180 degrees out of phase is constantly logic 1; in between, the XOR of the waves is a signal whose duty cycle is proportional to the phase difference between the input signals. The integral of the XOR gate output is therefore a voltage proportional to the phase difference between the signals at the two microphones, and the digitised version of this is provided to the control processor. It should therefore be possible to estimate the bearing of a sound source from the phase difference between the signals arriving from it at the two ear microphones.

Problems with phase comparison as a directional indicator

While such an acoustic sensor satisfies, in principle, the need for a direction estimator, it is inadequate in practice. There are both conceptual and technological barriers to the implementation of an effective acoustic sensor.

To determine the direction of a single-tone sound source, the robot looks for a difference of zero between the phases of beacon signals arriving at its two ear microphones. This happens if the sound source is directly ahead or directly behind, but also if it is at any other position where it is any multiple n of λ farther away from one microphone than the other (where λ is the wavelength of the tone heard). In effect, the

"lines of zero phase difference" in the plane around the microphones are hyperbolas. The equations of these hyperbolas can be found by noticing that they describe the lines of intersection of pairs of circles, one centred on the left ear, the other on the right ear, and differing in diameter by $n\lambda$.

Assuming that the baseline of the ears lies on the x-axis of a 2-dimensional Cartesian coordinate system, that the midpoint of the baseline is at the origin of the system, and that the distance between the ears is $2h$, the circles are

$$(x+h)^2 + y^2 = r^2 \quad \text{and}$$
$$(x-h)^2 + y^2 = (r+n\lambda)^2.$$

The intersection of these circles gives

$$(x-h)^2 - (x+h)^2 = (r+n\lambda)^2 - r^2,$$

so

$$r = -\frac{2h}{n\lambda}x - \frac{n\lambda}{2},$$

which is substituted in to $(x+h)^2 + y^2 = r^2$ to get

$$\frac{x^2}{n^2\lambda^2} - \frac{y^2}{(4h^2 - n^2\lambda^2)} = \frac{1}{4}.$$

This equation defines a family of hyperbolas that mark the places in the plane where the sound source might be, given that it is arriving in phase at the two ears. The range of values for n which are physically meaningful is determined by the relationship between the inter-microphone distance and the wavelength of the sound, namely

$$n = 0,\ldots,\left\lfloor \frac{2h}{\lambda} \right\rfloor.$$

Since the hyperbola equation given above is not defined at the ends of this range, a more convenient formulation is

$$(n^2\lambda^2 - 4h^2)x^2 + n^2\lambda^2 y^2 = \frac{n^2\lambda^2}{4}(n^2\lambda^2 - 4h^2).$$

Then, at $n = 0$ the hyperbola collapses into the line $x = 0$, corresponding to the possibility that the sound source is straight ahead or behind. If the wavelength is bigger than the distance between the microphones, this is the only case where there will be no phase difference between the signals at the ears. Also, if

$$\left\lfloor \frac{2h}{\lambda} \right\rfloor = \frac{2h}{\lambda},$$

then the last hyperbola is $y = 0$, meaning that if the sound source is on the ear baseline,

it will show a phase difference of zero at the ears. Other values define zero-phase-difference hyperbolas between straight ahead and due left or right (see Figure 3.6 for a typical hyperbola family for a particular combination of inter-ear distance and sound wavelength). Note that substituting

$$m+\tfrac{1}{2},\ m=0,\ldots,\left\lfloor\frac{2h}{\lambda}\right\rfloor-1$$

in for n determines the lines of antiphase.

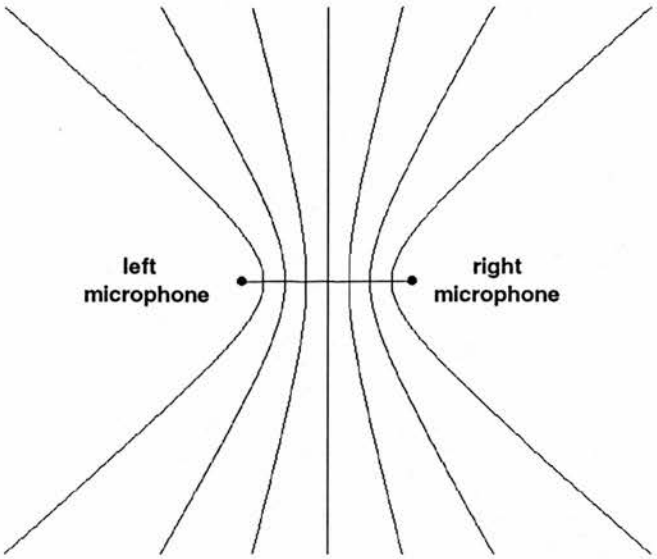

Figure 3.6. Lines of zero-phase-difference: signals from a point source anywhere on these lines will arrive in phase at the two microphones; the presence of four hyperbolas implies that the wavelength of the sound is between two thirds h and one half h $(h/2 < \lambda \leq 2h/3)$

The most constrained situation arises when the inter-microphone distance and sound wavelength are chosen so that

$$\left\lfloor\frac{2h}{\lambda}\right\rfloor = 0,$$

since in this case the sound source is either directly ahead of or behind the robot. With $2h$ = 120 mm, the critical frequency is about 2766 Hz, assuming a wavelength of sound of 332 m/s. This interaural distance was originally chosen since, $\lambda > 2h$ for all of the tones in the DTMF system (although only one tone of a pair may be active at a time for phase comparison).

However, two observations may be made from the foregoing analysis: first, the amount of ambiguity in direction estimation by phase difference measurement at the two microphones depends critically on the relationship between the inter-microphone distance and the frequency of the sound being received; and second, even when $\lambda > 2h$, a binaural system cannot determine whether a sound source is ahead of it or to

the rear. More importantly, however, it is intuitively clear that phase comparisons in this way are only meaningful if the signals at the two microphones are arriving *directly* from a single, spectrally pure, "point" sound source. While mounting the microphones on a rotating head platform makes it possible in principle to eliminate the directional ambiguity, constructing point sound sources producing a spectrally clean signal is very difficult in practice.

Furthermore, it is unreasonable to expect that the microphones will not receive both direct and multipath (reflected) versions of the same signal (except, perhaps, if the robot were operating in a highly contrived environment such as an anechoic chamber). With $\lambda > 2h$, experiments with the acoustic sensor on the robot and a tone generator showed that noise and interference from multiple reflections and harmonics eliminated any usable correlation between direction and phase difference. Nevertheless, the tone decoders and signal amplitude subsystems retain some usefulness, as will be discussed below and in Chapters 5 to 7.

Infrared directional sensor

A usable alternative scheme was to equip the robot with an infrared light beacon detector. This module, based on an infrared photodiode, is designed to detect emissions from infrared beacons in the vehicle's environment, and to estimate the bearing of a beacon relative to the sensor. The beacons generate an infrared light carrier wave (wavelength approximately 450 nm), modulated by a square wave signal. The detector is organised as in Figure 3.7.

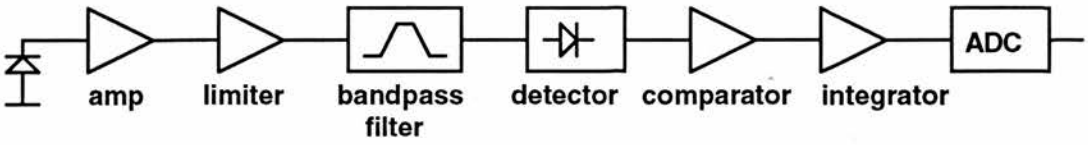

Figure 3.7. Infrared beacon detector block diagram

The receiver consists of a diode detector that separates the modulating signal from the IR carrier wave, a binary thresholding comparator that is off when an appropriate signal is being detected (and on otherwise), a low-pass filter that essentially integrates the comparator output over time, and an analogue-to-digital converter allows the filter output amplitude at any instant to be digitised for computer processing.

The photodiode is housed within a rectangular box 15 mm wide, by 12 mm high, by 30 mm long. The unconstrained field of view in the horizontal plane is thus 28°. When a beacon is located within this field, the modulating signal is constantly detected,

the comparator output is constantly off, and the integrator output has a small value. As the relative bearing of a beacon increases beyond 14° from the major axis of the housing, its emissions reflect off of the walls of the mouth of the housing, and continue to be sensed by the photodiode; however, the scattering of the light by the material of the housing, coupled with diffraction and interference effects, results in intermittent signal detection. The effect of this intermittence is that the thresholding comparator turns on in the absence of a detectable carrier, and the integrator output level rises in proportion.

As the bearing of the beacon increases towards 90° to the housing axis, the surface area of the housing mouth exposed to the beacon decreases, the signal is detected less reliably, and the comparator is nearly always on. The integrator output approaches its maximum value under these conditions. In practice, this "infinity" is reached when the beacon bears approximately 65° clockwise or counterclockwise from the long axis of the detector housing. Therefore, between 14° and 65° relative bearing, the integrator output level varies in proportion to the direction of the beacon.

It should be noted that this effect was observed experimentally, but was not anticipated. The sensor was originally intended only to be used in a binary fashion: mounted on the rotating head, it would be turned until a beacon was detected; the bearing of the beacon would be estimated from the angle of the head when the detector was approximately centred in the beacon's beamwidth. The additional quantitative relationship between bearing and digitised integrator output (given in Figure 3.8) was not expected. This relationship is non-linear, and is a function of the changing surface area available for reflection at the mouth of the detector housing.

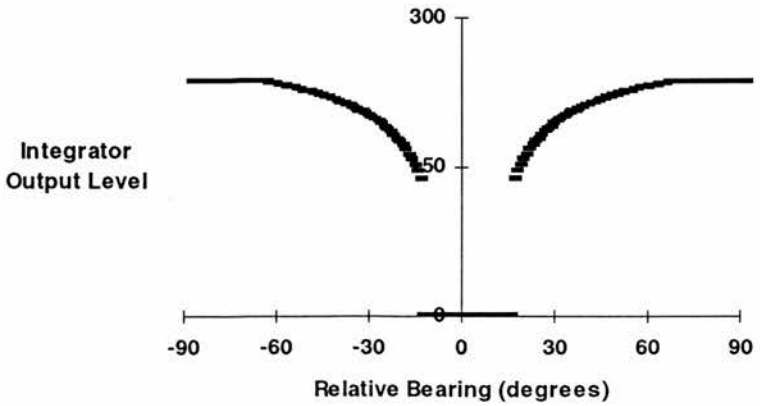

Figure 3.8. Integrator output (digitised) against bearing

Of course, as the infrared beacon detector is mounted on the "head" platform,

provided the vehicle is within view of the beacon, it is possible to rotate the detector until it is pointing directly at the beacon, and thereby to determine the beacon's bearing to within the 0.78° resolution of the head angle. However, in the qualitative mapping experiments described in Chapters 5 to 7, the head was only permitted to rotate in steps of size 22.5°, so that the additional quantitative information was useful in determining the bearing of a beacon even when the detector was not able to point directly towards it.

Unlike the acoustic phase difference subsystem, the infrared beacon detector was found to be an effective directionally-sensitive device. However, as no facility is provided for distinguishing one beacon from another, infrared beacons were attached to acoustic beacons, and the tone decoding subsystem described previously was used as a cross-modal mechanism for identifying a particular IR light source as follows: the set of IR beacons are strobed, so that only one is active at a time; when active, the loudspeaker onto which the beacon is mounted generates one of the DTMF tones described above; when the robot detects an IR beacon, during the periods that the beacon is on the robot attempts to determine which tone it is hearing in order to identify the IR source. The set of beacons can naturally be strobed relatively quickly (so that each is active for a few hundred milliseconds, say), as the tone decoding system is able to identify a tone in a very short time (as the basis for touch-tone telephone dialling hardware, it must be able to cope with fast button-pushing); this gives the effect of all of the beacons being on almost at once.

3.2.5 Sensory systems for localisation

Given a directionally-sensitive sensor like the infrared beacon detector, the robot can be made to move in a tropistic fashion relative to a reference beacon mounted in its environment. However, as will be argued in Chapter 5, tropistic behaviour is restrictive in that it does not allow the robot to visit an arbitrary place in its operating environment. If the vehicle is required to move to places that do not coincide spatially with beacons or landmarks, then the robot would have to be provided with some mechanism for finding its way relative to these features.

For instance, if the robot could detect the bearings of three beacons or landmarks, then it could triangulate and estimate its position in space relative to the references. If only one or two beacons were available, however, it would not be able to determine its position uniquely unless it could determine the distance to one of the beacons. It could also identify its position uniquely if it were provided with an additional orientation

reference such as a compass, or by tracking changes in its orientation relative to some initial value. The latter means of estimating orientation can also be used to determine the robot's position in some world-centred coordinate system, and is often called dead reckoning. Any of these options requires additional sensory devices capable of generating the required kind of information (direction or orientation).

Sound level as a distance indicator

As mentioned earlier, there is a inverse-square relationship between sound level and distance. Figure 3.9 shows, for the three equally powerful acoustic transmitters in the laboratory, sound levels measured at various distances in the room. The data for this plot were obtained as described in Chapter 5. Two important characteristics are apparent: the first is the level section of all three plots between zero and 1.5 metres from the source; the second is that, while amplitude and distance are clearly inversely proportional, the data are increasingly unreliable as source proximity decreases. The first observation is easily explained: the knee at 1.5 m represents the point at which the 8-bit digitiser saturates; although the sound level reaches a maximum immediately adjacent to the source, all levels measured at distances closer than 1.5 m are also converted to the digitiser's maximum value of 255. Irregularities beyond this distance are due to fluctuations in the output amplitude of the transmitters, asymmetries in the shape of their radiation patterns, or standing wave patterns in the room.

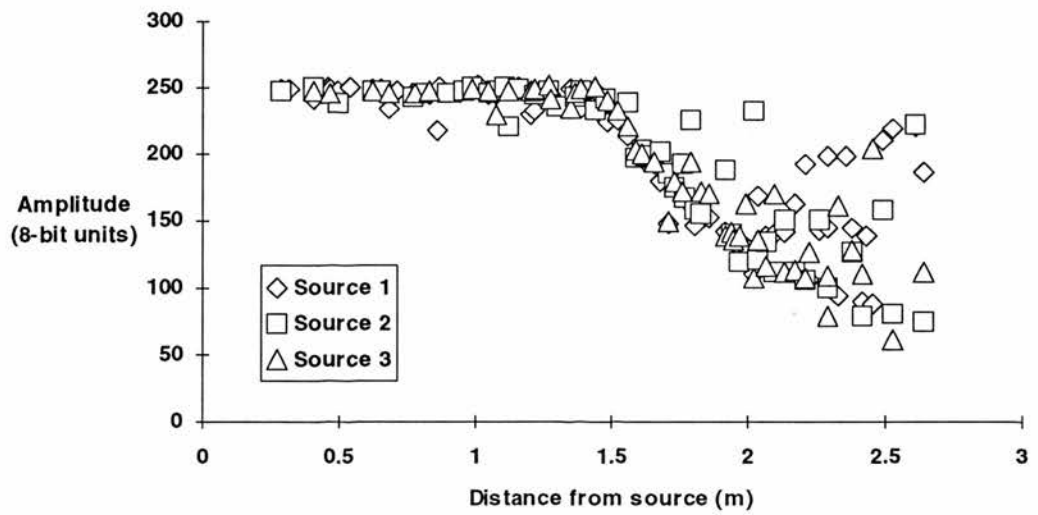

Figure 3.9. Relationship between the (8-bit) digitised amplitude of a sound and the distance of its source for the three loudspeakers in the laboratory

With high quality sound sources, an environment with few acoustic reflectors (and no spurious sounds), high resolution analogue-to-digital converters, and perhaps

microphone signal amplifiers with adjustable gain control, the performance of the system could be improved. Still, it is difficult to imagine such a system working under anything but highly contrived circumstances. Analysing the nature of received signals computationally might help compensate for some of these problems, but this kind of analysis is non-trivial. Empirical results with the simple acoustic sensor discussed in Chapters 5 to 7 suggest that it must be supplemented by other means if localisation is to be reasonably accurate.

Ultrasonic module

A popular procedure for generating distance information is to emit a pulse of sound, and measure the time that elapses before the sound's echo is heard. The sound's time of flight is a function of how far it travelled. The wavelength of the sound must be smaller than the distances to be measured[3]; in practice, this means that ultrasonic (above the range of human hearing) signals are used. This approach to distance measurement is used by ships, submarines, and some aquatic mammals, bats, and even perhaps certain insect species (e.g., Suga, 1990); it is often called *sonar*, short for *so*und *na*vigation and *r*anging.

The robot is equipped with such a device, based on a transmitter/receiver pair of ultrasonic transducers operating at a nominal frequency of 40 kHz. The sensor is organised as in Figure 3.10.

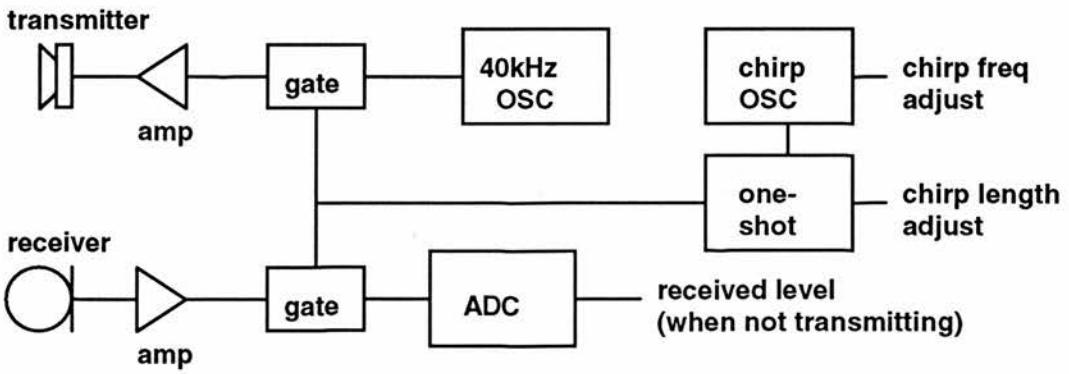

Figure 3.10. Ultrasonic sensor block diagram

In the transmitter half of the circuit, a 40 kHz oscillator is pulse-modulated by a gating oscillator, normally producing a 1 ms burst of carrier every 35 ms. Assuming a speed of sound of about 332 m/s, the transmitted pulse is 33.2 cm long, and in 35 ms

[3]Techniques exist for resolving distances smaller than λ, but are generally complex to engineer.

travels 9.5 m. The system is therefore capable of detecting reflections of the pulse from objects up to 4.7 m distant. The chirp length and period of repetition are continuously adjustable. However, in this simple system the length of the pulse limits the distance to the nearest detectable object: the proximity of transmitter to receiver means that, during the entire period of transmission, the outgoing pulse will saturate the receiver and drown out the sound of any potential echoes. Similarly, longer periods between transmissions mean reflections from objects at greater distances may be measured, but the current device's output power and the fact of exponential attenuation of high frequency sound in air suggest that listening for echoes from objects farther than 5 m to 10 m away is pointless.

The receiver consists of a diode detector that separates the amplitude of received signals from the 40 kHz carrier wave, and an analogue-to-digital converter allows the received amplitude at any instant to be digitised for computer processing. During pulse transmissions, the ADC input is grounded, so that a computer reading it has an unambiguous temporal reference. This allows both transmitter and receiver to operate continuously, with the computer connected to the ADC required to synchronise its clock with that of the sensor.

The system is thus used as follows: the ADC is polled continuously until its input is at 0 V, indicating that the transmit pulse has commenced. The polling processor's clock is reset to 0, and the ADC is sampled until its value returns to 0, indicating the start of the next pulse.

By referencing sampled amplitudes to the time since the start of the transmit pulse at which they were obtained, large amplitudes suggesting a reflection from an object can be related to the distance of the putative object. The relationship is $d = 0.5 \times 332t$, where d is the distance of the object in metres, and t the time in seconds between transmission to reception; the factor of 0.5 is included as the reflected pulse has travelled an out and return distance twice as great as the actual distance to the reflecting surface.

In practice, a typical amplitude-time graph for the device looks like Figure 3.11. The initial amplitude of 0 corresponds to the time during which the transmitter is active. The first actual amplitude measurements (starting at about 1 ms) from the receiver show its decaying response to the transmitted pulse, which saturates the receiving transducer as the latter is within 3 cm of the transmitter.

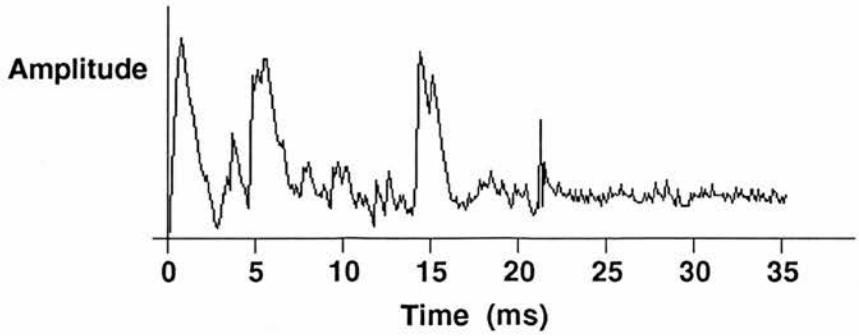

Figure 3.11. Typical scan: time increases from left to right, and the height of the line is proportional to received amplitude

As the reflection peaks are temporally wide, it is convenient in practice to take a simple 3-point numerical derivative of the samples, and compute the distance to objects from the large rates of change associated with the steep leading flanks of echoes. For the typical scan in Figure 3.11, the derivative plot appears as in Figure 3.12.

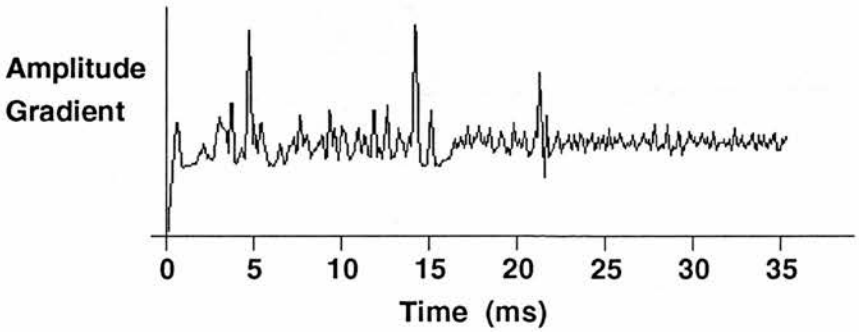

Figure 3.12. Simple -1/0/+1 mask-type derivative of scan in Figure 3.11; the large peaks correspond to the leading edges of prominent reflections seen in Figure 3.11

The spatial resolution of the system is at least constrained by the maximum sampling rate of the ADC. In this circuit, the ADC digitises in 25 μs, during which time sound travels about 8 mm. However, this conversion rate and resolution are only attainable if the sampling computer can process the ADC's output values sufficiently quickly. It has been found empirically that processing speed limitations, coupled with the effects of noise on the reflected signal, have the result that distances may be resolved reliably only to within about 2 cm. Still, it will be shown in Chapter 7 that this system in combination with the infrared beacon detector provides a very effective basis for spatial localisation.

Qualitative spatial sensing

While the sonar sensor described above has the potential for generating metric information, the approach to localisation embraced in this work (Chapters 5 to 7) attempts to sidestep the interpretation of sensory information in this way. Instead, for reasons that will be more fully motivated in Chapter 5, it (and the sonic and infrared beacon systems described in the previous section) will be used as basic space-segmenting devices (i.e., devices which merely associate places in the robot's environment with a certain value or set of values), rather than as a means for explicitly characterising the geometry of the robot's environment. As discussed in Section 2.3, the primary motivation for this is that the quantitative interpretation of sensor data requires robust models of the physical characteristics of the sensors, which are in general difficult to obtain.

Sensing for dead reckoning

From a model of the robot's kinematics and a device which measures the angular displacement of the vehicle's wheels, it is possible to estimate the position and orientation of the robot in some world-based coordinate system. Appendix B offers a simple kinematic model for the robot considered here, and derives the resulting displacement and orientation equations for the robot as a function of time. Briefly, assuming that the robot's wheel encoders are sampled every T seconds, and a Cartesian world coordinate system, it can be shown that the x- and y-coordinates of the robot ($p_{x\mathrm{RB}}$ and $p_{y\mathrm{RB}}$, respectively) after n sampling intervals are given by

$$\begin{bmatrix} p_{x\mathrm{RB}} \\ p_{y\mathrm{RB}} \end{bmatrix}(nT) = \begin{bmatrix} p_{x\mathrm{RB}} \\ p_{y\mathrm{RB}} \end{bmatrix}[(n-1)T] +$$

$$+ \frac{TR}{4}\begin{bmatrix} -\sin\theta_{\mathrm{RB}} & -\sin\theta_{\mathrm{RB}} \\ \cos\theta_{\mathrm{RB}} & \cos\theta_{\mathrm{RB}} \end{bmatrix}[(n-1)T]\left(\begin{bmatrix} \omega_{x1} \\ \omega_{x2} \end{bmatrix}[(n-1)T]+\begin{bmatrix} \omega_{x1} \\ \omega_{x2} \end{bmatrix}(nT)\right).$$

Its orientation, θ_{RB}, is given by

$$\theta_{\mathrm{RB}}(nT) = \frac{R}{2l_a}[\theta_{1x}(nT)-\theta_{2x}(nT)] + \theta_{\mathrm{RB}}(0) - \frac{R}{2l_a}[\theta_{1x}(0)-\theta_{2x}(0)],$$

where θ_{kx} is the total angular (rotational) displacement of wheel k about its axle, and other notation is explained in the Appendix.

The main observations here are that the x- and y-displacement terms of the robot's position after n sampling periods are a function of its displacement, velocity, and

orientation after n-1 sampling periods. Its orientation after n sampling periods is determined directly from the difference between the total displacement of the two wheels about their axles. Therefore, all of these quantities depend on a device for measuring wheel displacement.

The shaft encoders required for this process are implemented as follows: at one end of the axle of each of the two driven wheels is mounted a narrow disc with regularly spaced, equal-sized square perforations encircling its outer edge. Two infrared light emitter/detector pairs are positioned such that the disc passes between emitters and detectors, alternately allowing a detector to see and not see its associated emitter as the perforations and opaque sections between them pass and block the emitter's beam. Using two beams arranged so that as the "leading" (relative to the direction of rotation) edge of one perforation is passing the first beam, the "trailing" edge of the same perforation is passing the other beam (Figure 3.13), it is possible to measure the rotation of the disc to an angular resolution in degrees of 360 divided by four times the number of perforations. The disc used on the vehicle has 96 perforations, so that with the quadrature decoding arrangement described above, it is possible to measure the angular displacement of each axle to a resolution of just under 1°.

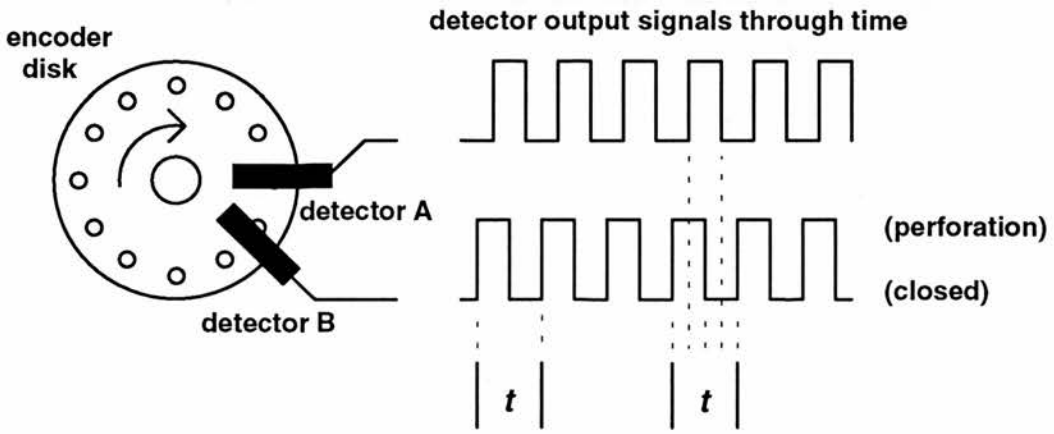

Figure 3.13. Quadrature decoding arrangement on driven axles: a detector is in "perforation" state whenever any part of a perforation intersects with the black rectangle that represents the detector; t is the time it takes for a perforation to pass a detector, so that with the staggered arrangement of two detectors shown here, each of these intervals contains 4 detectable events, thus angular resolution is a function of 4 times the number of perforations

On the vehicle, the detector outputs are provided directly to the control processor. This device samples them continuously (at 10 kHz); when a perforation edge passes a detector (the detector's output changes state), and from its knowledge of the direction of rotation of the axle, the processor either increments or decrements an angular

displacement counter (denoted in the equations above and in Appendix B by θ_{1x} and θ_{2x}, for the left and right wheels, respectively).

Although dead reckoning offers a straightforward basis for estimating position, it suffers from decreasing accuracy as the robot travels around due to slippage between the wheels and the floor. It must occasionally be calibrated with respect to an external position and orientation reference. For straight-line motion, the system implemented here was found to maintain position estimates accurate to less than 1% of distance travelled in both the direction of motion, and in orientation. Despite good contact between the robot's rubber tires and the wooden floor of the laboratory, if the robot turned frequently its orientation estimate drifted by up to 1° per 30° of rotation.

Therefore, the use of the dead reckoning system in this work is limited. In developing a smooth locomotion competence (Chapter 4), the system was used to generate traces of the robot's motion over short distances, involving little rotation, so that drift was not a significant in this application. In the localisation work (Chapter 5 to 7), it was used in obtaining an approximation to the robot's orientation.

Geocentric orientation

As shown in Appendix B and discussed above, given shaft encoders on each axle the robot's orientation (relative to some allocentric frame of reference attached to the surface it is driving around on) can be found at any moment by subtracting the two encoder counts and multiplying the difference by a suitable constant (offset by the difference between the counts when sampling began and the vehicle's orientation at that time). This estimate is subject to errors due to wheel slippage and friction, and unless the counters are occasionally calibrated, these errors grow with time. In order to obtain a less temperamental orientation reference for the localisation experiments (Chapters 5 to 7), the robot was equipped with a device for detecting its orientation relative to the earth's magnetic field.

The basis for this device is a commercial flux-gate electronic compass (Micronta 63-641), intended for use in automobiles. It consists of a small pickup assembly (a pair of inductors), and larger unit containing circuitry that measures the effect of geomagnetism on the coils to determine the orientation of the pickup relative to magnetic north. The output of the device is in the form of a vertically-mounted, rotating circular disk inscribed with the usual markings of a compass rose.

The marked direction currently closest to the top of the display is calibrated to

indicate the orientation of the pickup assembly relative to the earth's magnetic field. Conveniently, the output device's angular position is determined by two voltages which are a function of the sine and cosine of this relative orientation. Therefore, the commercial device was fitted to the robot with the pickup aligned along the robot's longitudinal axis, and the display unit has been modified to provide the two voltages in a digitised form to an external processor. This processor computes the four-quadrant arctangent of these two values in order to establish the robot's current heading with respect to magnetic north.

In practice, this device is quite robust, provided the pickup unit is mounted away from the magnetic fields of the motors, and that the robot's environment is free of strong confounding magnetic influences. In a typical office environment and on the robot described here, it has been found empirically that these conditions are sufficiently met. With 8-bit ADCs sampling the two compass voltages, the angular resolution of the device is not linear, as the plot in Figure 3.14 shows. Where this curve is steepest, the computed angle changes most per change in voltage. This worst case situation happens near 0°, 180°, and 360°, where the angular resolution is only about 8°. However, the resolution everywhere else is almost consistently better than 1°.

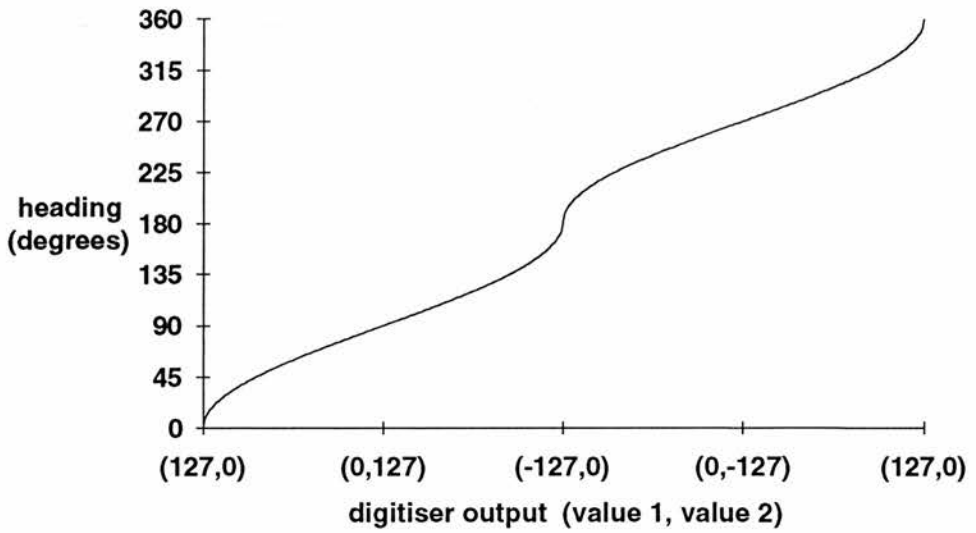

Figure 3.14. computed heading plotted against the combinations of the two digitised voltages (value 1 and value 2) from which it is derived; the steeper the curve, the poorer the resolution

3.3 Summary

This chapter outlined a robot behaviour programming system with components based on features of both analytic and synthetic robotics, supporting the realisation of an autonomous vehicle with locomotor and spatial capabilities. It also documented the technical details of a robot on which various behavioural experiments were performed. In summary, the robot has the following facilities:

- two independently driven wheels and a caster, arranged so that the robot can rotate on the spot, drive forwards or backwards, or trace out smooth curves; it is easy to control because it can reach any place in its environment by executing a sequence of rotations and straight-line forward movements;
- a front bumper and forward- and backward-facing IR light emitter/detector pairs used for detecting or avoiding collisions;
- a rotatable head mechanism fitted with microphones, able to detect the intensity of sound sources, and with an infrared beacon detector, able to determine the bearing of an IR light source;
- a module which emits bursts of ultrasound, and is able to detect returning signals which have reflected from objects in the robot's environment;
- shaft rotation transducers fitted to its driven axles, from which the robot's motion relative to the surface it is moving along can be estimated; and
- a fluxgate compass which provides an approximation to the robot's orientation with respect to the earth's magnetic field.

The remainder of this dissertation discusses experiments with this robot, beginning in the next chapter with locomotion, and continuing in Chapters 5 to 7 with the problems of map-based localisation.

Chapter 4

Smooth Goal-Directed Locomotion

In Chapter 2, we saw that there are significant differences between the "analytic" and "synthetic" approaches to the design of even the simplest enabling competences of a mobile robot, i.e., the very ones that allow it to move around controllably and without causing damage to itself or to objects in its vicinity. The analytical approaches generally start from an explicit representation of the robot's operating space, and then by means of one of a variety of techniques, plan the actions that will take the robot from its current location to where it is required to be next. Any unexpected obstacles to a vehicle's progress along the way are treated as exceptions, often requiring the generation of a new action plan.

The synthetic approaches, on the other hand, reject planning as an article of faith: first, because unforeseen events *en route* are inevitable in most environments, so that it makes little sense to pre-plan an activity only to have to abandon the plan and make a new one if either the environment is perturbed, or the robot fails to move exactly as planned; second, route planning invariably requires some model of the robot's environment, which it seems pointless (or even impossible) to provide wherever this environment is variable. Moving around is therefore implemented as a largely reactive process, based on an "obstacle avoidance" subsystem that is called into play as required.

The appeal of this treatment of obstacle avoidance, as noted in Chapter 1, is that it allows the other aspects of the robot's behaviour to be implemented with the assumption that they are underpinned by an asynchronously operating virtual machine which guarantees collision-free robot movement. However, the impenetrability of the mechanism tends to prevent the kind of interaction between competences that appears to be necessary to turn robust locomotion into efficient locomotion, at least when the robot is required to move towards a goal.

It will be argued in this chapter that there are features of the analytical approaches to locomotor competence design which couple well with the obvious advantages of on-line control typical of the synthetic enterprises. In particular, we will see that it

makes sense to try to "fit" the robot's speed to the environment: beyond just avoiding potentially damaging collisions with objects in its path, it should move smoothly in the world, slowing down where there is a great clutter of potential obstacles, and moving more quickly in open spaces. It will be shown that this can be done reactively, but is more effective with some predictive capability, either by means of longer-range sensing or environment representations conventionally the reserve of analytical robotics. This discussion has also been given restricted coverage elsewhere (Donnett and McGonigle, 1991).

4.1 Basic motion, obstacle detection, and obstacle avoidance

In the absence of an overarching goal or journey destination, a mobile robot is said to be performing a "random walk". Its locomotor competences need only consist of some basic mechanism for activating its drive motors to propel it a specified speed, and some means of detecting and circumnavigating obstructions it encounters. As discussed in the previous chapter, obstacle detection on the robots considered in this work is primarily the task of a collection of infrared emitter/detector pairs mounted on the vehicle's chassis and rotating "head".

A simple obstacle detection/avoidance strategy that would conform to the general synthetic design philosophy is depicted in Figure 4.1. Here, the three IR emitter/detector pairs on the robot's body are used to detect eight different obstacle situations, and obstacle avoidance consists of a set of basic manoeuvres like turning left or right, or reversing and then turning.

For cases B, G, and H, the obstacles are central, so that the robot can choose to turn either left or right. When the robot is walking randomly (bumbling from obstacle to obstacle), this choice is arbitrary. By biasing the choice in favour of one or the other cardinal direction, it is possible to transform the motion of the vehicle into something with a bit more purpose: as long as the geometry of its environment obeys certain rules, the turn bias with these simple avoidance tactics results in wall-following behaviour. Such a basic tropism is a successful (although not usually efficient) tactic for negotiating certain kinds of mazes.

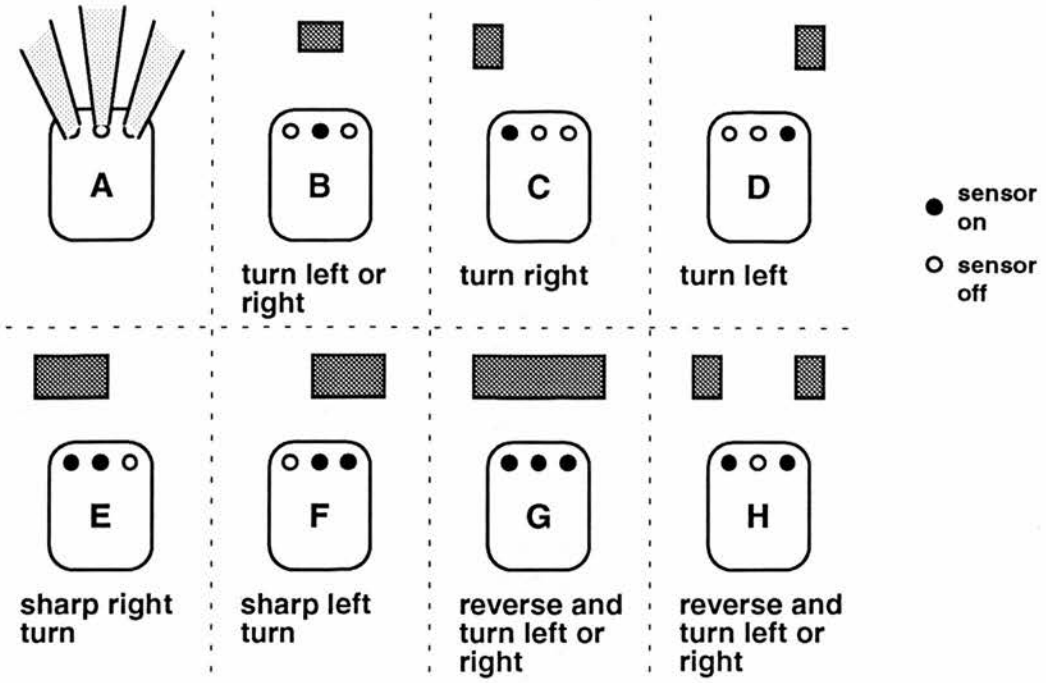

Figure 4.1. Eight basic obstacle configurations and avoidance tactics (top view, robot moving towards the top of the page); the hatched shapes are obstructions; the lettered boxes represent the robot, and the three circles show the activity of its three IR emitter/detector pairs; **A** depicts the approximate sensor fields of view

Furthermore, Nehmzow *et al.* (1989) have built a robot that uses wall-following as the basis for localisation: it learns about corners in the walls it is following in order to construct a representation of its operating space. However, any system whose ability to position itself is anchored to the shape of its operating environment can be inconvenienced, at least temporarily, by perturbations to that environment, such as the arrival of a new object. A common approach to overcoming this potential problem in robot navigation is to equip the robot with some device for orienting itself relative to a locative reference in its environment.

4.2 Goal-direction motion

While mobile robot spatial capabilities are primarily the subject of later chapters of this document, the present argument is that their eventual inclusion in the system bears on the design of locomotor competences; it is therefore necessary to introduce the basis for some kind of spatial sense.

The simplest locative reference is afforded by a guiding beacon or landmark external to the robot, whose direction relative to the robot's current bearing can at least be partitioned into "left", "right", or "centre". This is the essence of *tropistic*

motion: the vehicle can reach the locative reference by always turning so that it is straight ahead, and moving forwards. The robot described in Chapter 3 can achieve this by means of the passive infrared detector mounted on its rotating head, where the guiding beacon is an IR source mounted somewhere in its operating space. The relative direction of the IR source is determined by rotating the head to determine the beam's relative direction, and then turning the robot so that it faces along the beam.

There is at least one way in which equipping a robot with just basic skills like avoiding obstacles and beacon tracking challenges any design approach that would package the avoidance competence away as an impenetrable unit in a distributed control system. It follows directly from the discussion of obstruction avoidance tactics in the previous section. Consider the simple situation in Figure 4.2: suppose that the geometry of the situation is such that when the robot sees obstacle B, it turns left, and then sees obstacle A only with the middle IR sensor; then, according to the tactics in Figure 4.1 it might choose to turn left or right. However, it would obviously be better if the obstacle avoidance subsystem were to bias its choice in favour of the direction of the goal, known to the beacon tracking system.

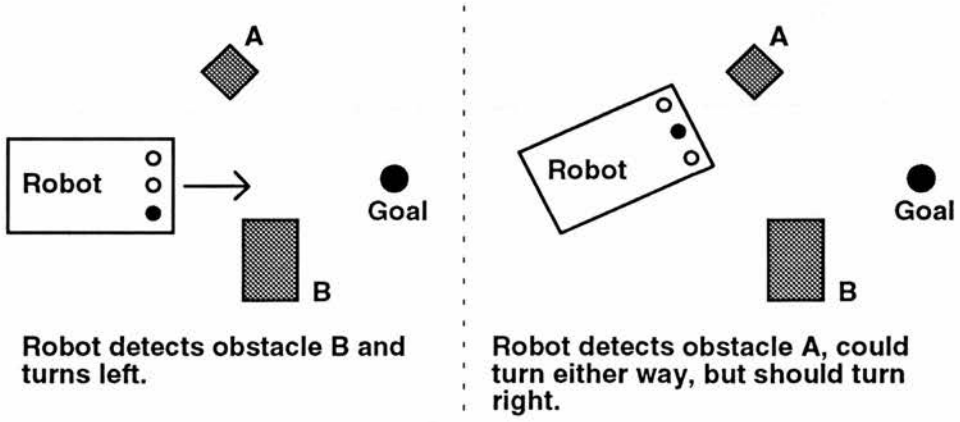

Figure 4.2. Situation where a simple obstacle avoidance strategy might unnecessarily delay arrival at goal if the robot arbitrarily turned left instead of right on encountering obstacle A (the robot is assumed to be able to fit through the gap between A and B)

Obviously this can be achieved by providing some input to obstacle avoidance that biases turn direction when conditions are such that this choice may be made freely, but this implies that this basic competence cannot actually be designed without some forethought as to how it may be affected by other subsystems, which appears to violate one of the fundamental tenets of the synthetic approach.

Furthermore, it will be shown in Section 4.3 that the typical synthetic-robotics implementation of locomotion as a straightforward on-line process is also deficient if

robust locomotion (simply not bumping into things) is to become *efficient* locomotion (being able to get to a target destination as directly as possible). First, however, it is necessary to explore a latent but pivotal characteristic of the relationship between locomotion and beacon tracking, namely that the speed at which the robot is travelling directly affects how its time is apportioned between heading towards its goal, and dealing with obstructions.

4.2.1 The effect of robot speed on goal-directed motion

In obstacle-free space, the robot simply heads towards its target beacon, and eventually reaches it. Suppose that there are obstructions between robot and beacon, however, and that the robot has been constructed according to synthetic principles, i.e., that obstacle avoidance and beacon tracking are separate encapsulated competences, operating on-line (without anything like trajectory planning). In this case, when an obstacle is encountered, the beacon tracking is interrupted and some avoidance manoeuvre is executed.

A simple physical analogy

For the sake of argument, it is reasonable to think of the robot as traversing an environment of evenly distributed objects, roughly equal in size, like a bunch of chairs. To begin with, assume that it does not take any time to negotiate an obstacle: the robot sees it and instantly changes course. In this case, the robot's trajectory can be thought of as being like a two-dimensional equivalent of the (3-D) mean free path of a particle in a gas[1] (see Figure 4.3).

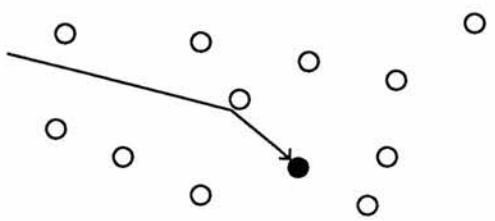

Figure 4.3. For a particle moving in a gas, the mean free path law states that it can expect to travel a distance $d = 1/ns$ m between collisions if the cross-sectional area of the gas particles is n m^2, and their density is s particles/m^3

In this case, if the objects had a diameter of, say, n metres, the robot could sense them at a distance of n metres also, and the mean density were s objects per square

[1]This analogy was suggested by Dr. G. Hayes.

metre, the mean free path expression would show that the robot can expect to travel $d = 1/ns$ metres between collisions, and if it is travelling at v m/s, it can expect a collision every $t = 1/vns$ seconds. In other words, as the vehicle travels faster in an area of constant object density, the mean *distance* between collisions is unchanged, but the mean *time* decreases.

The assumption that it takes no time to negotiate an obstacle is obviously invalid, and time spent avoiding obstacles is time not spent heading towards the goal. Suppose that the mean time between collisions is t_b, and that collisions take t_c seconds to overcome on average. Then, an estimate of the fraction of its time the robot spends navigating is given by

$$\frac{t_b}{t_b + t_c} = \frac{\frac{1}{vns}}{\frac{1}{vns} + t_c} = \frac{1}{1 + t_c vns}.$$

Clearly, the robot always spends some of its time navigating, unless either the time to deal with a collision, its speed, the size of obstacles, or obstacle density is infinite. Similarly, it is only able to spend all of its time navigating if it takes no time to negotiate obstacles, if it is travelling at zero speed, if the obstacles have zero-size, or if their density is zero.

Collision time constancy assumption

Thus, the simple physical analogy leads to reasonable estimates at the extremes of its parameters. Assuming relatively constant obstacle size and density, and that the time the robot takes to deal with a potential collision is also uniform, the model predicts that the faster the robot goes, the less time (proportionally) it spends doing something other than negotiating obstacles. If, however, t_c were to decrease as v increased, then the proportion of time spent navigating might be independent of speed. It might be expected that the latter situation would prevail: as the robot travels more quickly, it may have more obstacle encounters per unit time, but it can negotiate these more rapidly as well.

An initial test for whether or not speed directly affects the ratio of time spent on-course to time spent circumnavigating obstructions is obtained by means of a relatively simple simulation. The robot is modelled as a circle moving within a field of other circles, as depicted in Figure 4.4, where the robot (hatched circle) starts at the top left of the diagram and heads towards a goal situated at the bottom right. A trivial obstacle-detection model is implemented which attempts to mimic the robot's ability to distinguish the eight left-right-central obstacle configurations shown in Figure 4.1, and

the actual avoidance and tracking algorithms used in the real robot dictate how the simulated vehicle should respond to collisions.

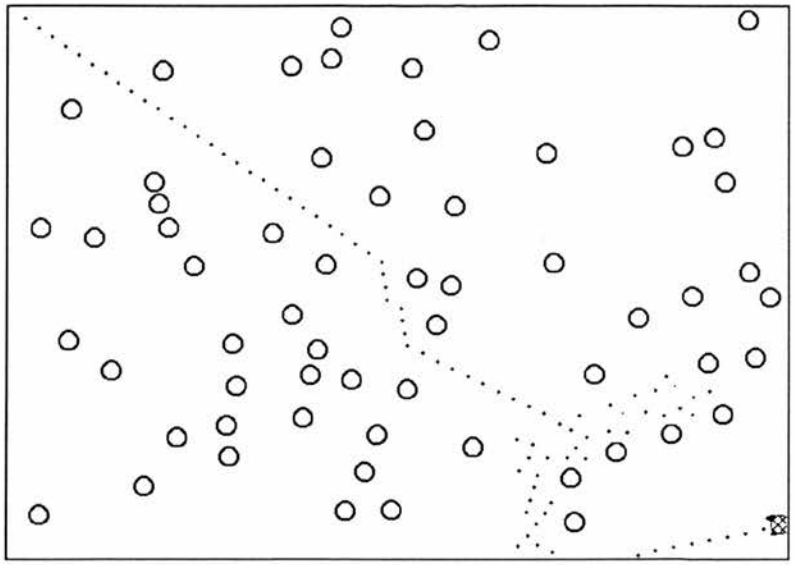

Figure 4.4. Depiction of a simulated "particle" robot moving through a "particle" obstacle field, from top left to bottom right; it should have been able to pass between the line of obstacles near the bottom right, but it was moving so quickly that it had encountered the next before realising that it had cleared the first

Time data were collected across many thousands of simulated runs, with many different random obstacle configurations, as a basic Monte Carlo test for typical behaviour from a random process. The results offer an impression of how the interaction between speed and density affects the balance between navigation and obstacle avoidance. Figure 4.5 shows typical simulation data for a range of densities, and for three different simulated speeds, with the simulated operating space (the bounding box in Figure 4.4) set to be 20 m wide by 20 m long, and the diameter of the circles 0.8 m.

The percentage times spent navigating are averages of 1000 runs at each density, with the configuration of the simulated obstacle space changed on every run. As predicted by the model (and as expected for a system lacking trajectory planning), increasing obstacle density invariably means proportionally less time spent heading towards the goal. Moreover, it is clear that for a given density, increased speed also results in less time spent navigating.

For example, at a density of 2.5 obstacles per square metre, the robot typically spends 10% less of its time navigating at 3 m/s than it would if it were travelling at 1 m/s. This suggests that as v increases, t_c stays relatively constant (for the results in Figure 4.5, t_c is of the order of 0.1 s for all combinations of speed and density). Thus,

travelling more quickly leads to more encounters per unit time, *and* it takes proportionally longer to deal with these. This results from the kind of effect that can be seen in the bottom right corner of Figure 4.4. When the robot is travelling quickly, it is prone to overshoot spaces between objects, meeting the next obstacle as soon as it has cleared the previous one, because it cannot detect and respond to them quickly enough.

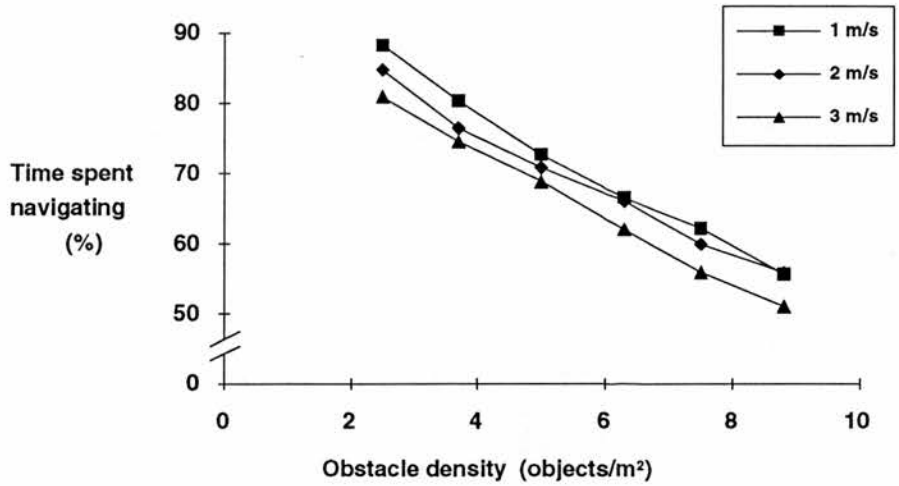

Figure 4.5. Typical simulation results for a range of densities

Quantitative predictions

The mean-free path analogy does make quantitative predictions. For example, if the density of obstacles is 2.5 objects/m², their average diameter 0.8 m, the robot's speed 1 m/s, and the time to detect and steer away from an obstacle 1 s, then the robot should spend about $1/(1+2.5\times0.8\times1\times1) = 1/3$ of its time navigating. The simulation results in Figure 4.5 show proportions of time spent navigating which are much higher, suggesting that the time it takes to avoid a collision is often much less than 1 s. This is reasonable in this particular case, as the obstacle detection system often sees an object off to the robot's right or left, and executes only a slight correcting turn to avoid it; only close to head-on potential collisions will take a long time to negotiate (the time to execute a turn increases monotonically with the angle of turn).

Quantitative predictions from the simple physical model are therefore difficult to make because of non-uniformities in the time it takes to negotiate obstacles. It would be necessary to treat the nature of obstacle encounters more thoroughly (e.g., by incorporating the likelihood that they are head-on versus glancing). However, the primary purpose of this exercise has not been to construct a rigorous, quantitative model of a physical process; rather, the important point was that travelling faster can

lead to less efficient goal-seeking behaviour, in terms of the proportion of the robot's time that is spent looking for the goal versus avoiding obstacles *en route*.

In any case, just as the model is only qualitatively related to the simulations, there are also reasons why the simulations might not accurately predict the behaviour of a real robot: for instance, even though the tactics for obstacle avoidance may be identical for simulation and robot, it is difficult to be certain that the sensor and motor models are sufficiently representative of the actual sensing process and the dynamics of the actual robot's motion, respectively (in any case, the discrete nature of the simulation differs from the continuous nature of the robot); also, the simulations assume a space of uniformly distributed objects of equal size, which does not typify laboratory or office environments. Therefore, the effect of speed on navigation has been tested on the prototype robot described in Chapter 3 under realistic circumstances.

4.2.2 Typical effects of speed on actual performance

Figure 4.6 represents an experimental pen about 3 m wide by 2.5 m long (see also photograph and associated description in Section 5.2.2), in which the actual robot is placed at the position marked "Start". At the box marked with the hollow triangle is an infrared light source, which the vehicle is meant to approach. This box is also a loudspeaker, and by means of the acoustic system described in Section 3.2.4, the robot determines when it has arrived close to the loudspeaker/IR beacon (the intensity of the sound detected exceeds some threshold). The arrowed line traces what looks to be a fairly optimal route between start and goal.

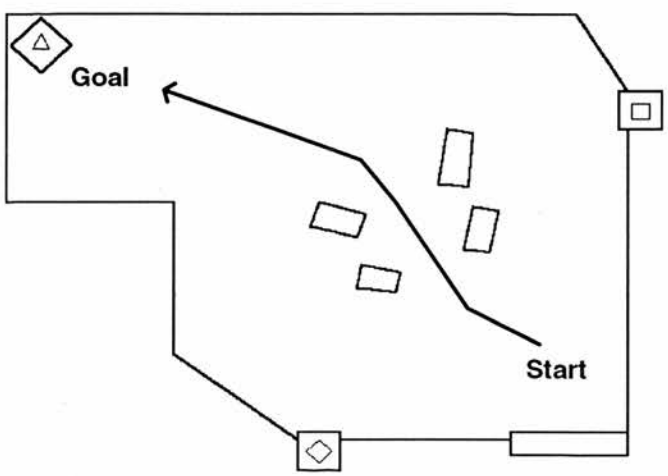

Figure 4.6. Situation where approaching the IR beacon at the Δ means passing through a field of obstacles (in this case, four cardboard boxes) over the top of which the beacon is always visible

A control system to generate this route could consist of two simple competing control modules, one for obstacle avoidance (having prority) and one which orients the robot towards the goal. The robot would then begin its journey by orienting towards the goal, and travelling forward; on encountering the obstacles, the robot's obstacle detection system would cause it to turn; the tracking system would continue to attempt to bias the robot towards the beacon, but this would not succeed until the obstacles had been cleared and the obstacle avoidance system had relinquished control.

Figures 4.7 through 4.11 show the pattern of the robot's motion as it attempts to reach the beacon at a variety of speeds; an image of the robot is plotted showing its position at the end of a 250 ms interval (i.e., four images are plotted per second of travel). These diagrams are based on displacement and orientation data from the robot's wheel encoders. This information is not perfect, and certainly becomes increasingly inaccurate as the robot drives around due to wheel slippage. However, the robot travels at most 3 m in these runs and empirical tests (Section 3.2.5) have shown that dead-reckoned position and orientation estimates do not drift significantly over distances much greater than this. The encoders were always reset at the start position, which was marked on the laboratory floor to ensure consistency between runs.

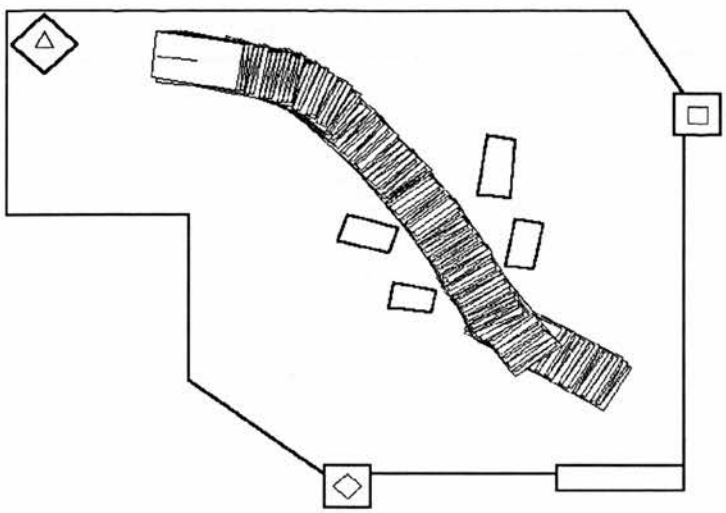

Figure 4.7. Trajectory for a desired robot speed of 4 cm/s

It is clear that for the slower speeds (4 cm/s and 8 cm/s, Figure 4.7 and 4.8 respectively), the robot follows a path that is reasonably optimal. It approaches the group of boxes slowly, detects the lowermost one (in the diagram), turns right slowly, and travels forwards into the gap between the boxes; it continues to detect the beacon on its left, but cannot turn towards it because the left IR sensor is detecting the obstructions. Once in the clear, the beacon tracking system causes it to turn towards

the beacon, continuing until the loudspeaker's signal exceeds a detection amplitude threshold for a sufficiently long period. However, when the speed is increased to 16 cm/s (Figure 4.9), an interesting effect occurs: the robot is now approaching the boxes very quickly, and when it is forced to turn left by the first one, it appears to overshoot (in fact it collides with the box on its right). It then seems to "oscillate" its way between the boxes, before it is sufficiently in the clear to be able to proceed smoothly.

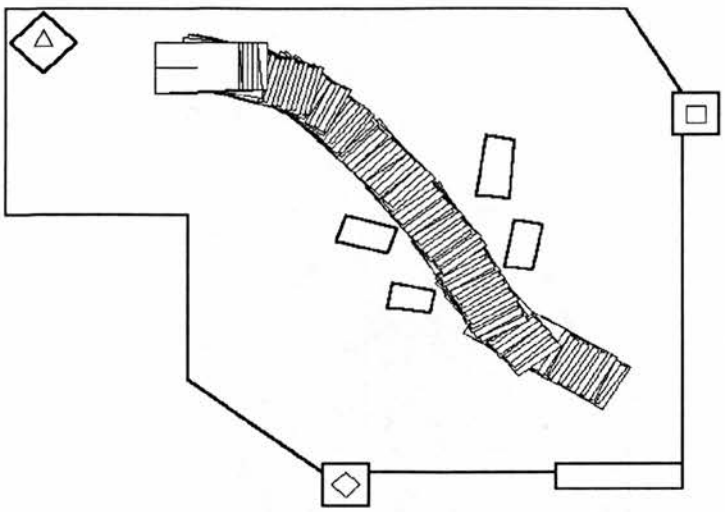

Figure 4.8. Trajectory for a desired robot speed of 8 cm/s

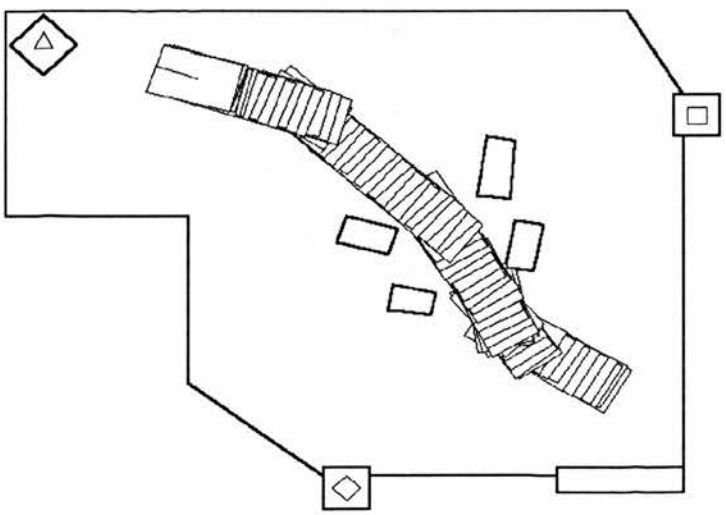

Figure 4.9. Trajectory for a desired robot speed of 16 cm/s

In Figure. 4.10, it can be seen that same effect occurs when the robot attempts to pass through at an even greater speed. In fact, the robot collides with the boxes on both sides of the "mouth" of the obstacle field. The robot occasionally (on perhaps one run in thirty) even behaves as in Figure 4.11, in which it fails to gain access to the

obstacle field because it keeps oscillating left and right between the boxes it first encounters, and eventually reverses (part of its avoidance strategy called into play to enable escape from such a failure to move forwards); on one turn, it turns left sufficiently far to be able to bypass the obstacle field and find an alternative (but much longer) route to the beacon.

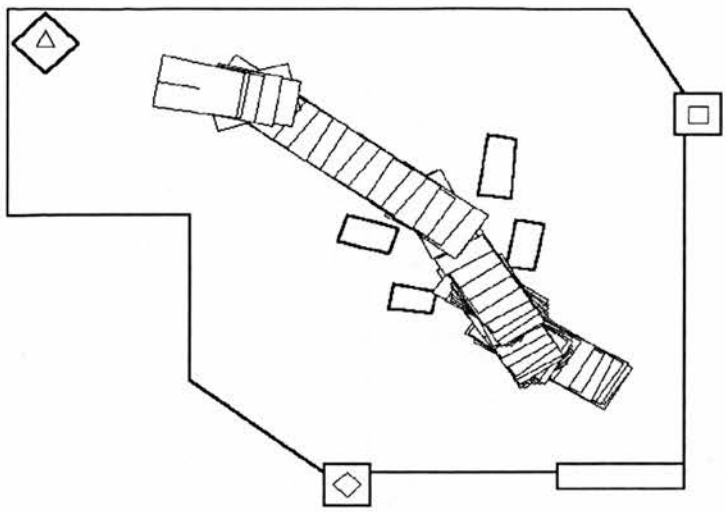

Figure 4.10. Trajectory for a desired robot speed of 32 cm/s

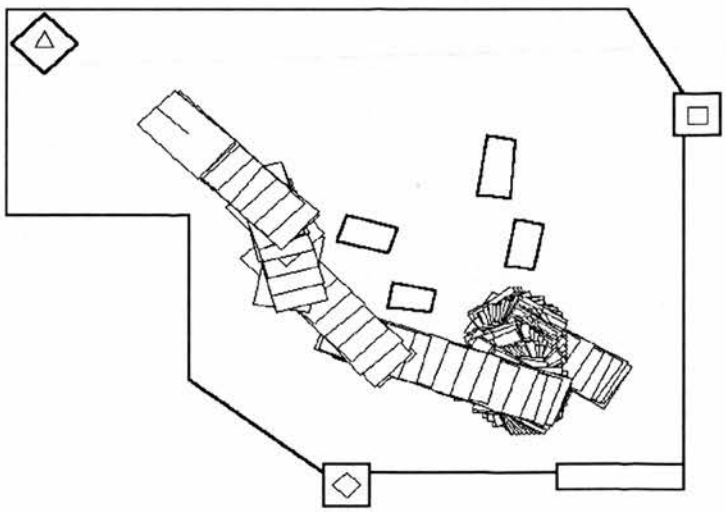

Figure 4.11. Trajectory for a desired robot speed of 32 cm/s; the robot fails to get between the boxes, and instead happens by chance to find another route to the beacon

By profiling the amount of time the robot spends executing the tracking versus avoidance parts of its control program, it is possible to compare the amount of time it spends navigating relative to negotiating obstacles. Figure 4.12 plots results for the typical runs depicted above. At 4 cm/s, the robot typically spends three quarters of its

time aiming for the beacon; but at 32 cm/s this proportion has dropped to half of its time.

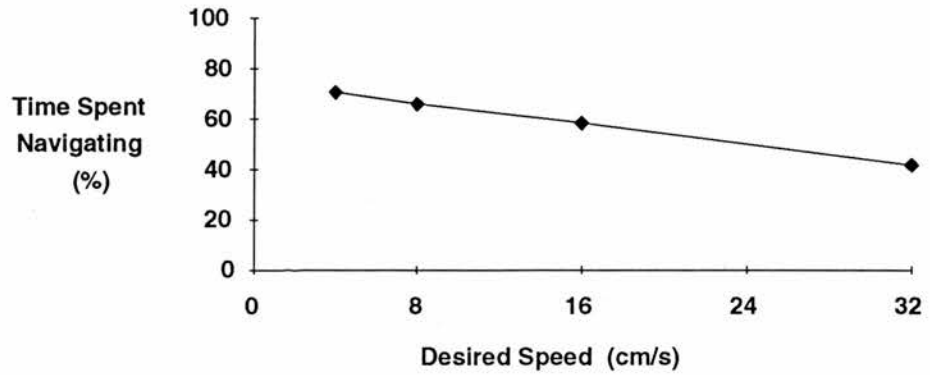

Figure 4.12. Proportion of time spent navigating for typical robot runs

Furthermore, the obstacle avoidance manoeuvres inevitably slow the robot down, as they require it to change directions, and occasionally even reverse. In Figure 4.13, the "desired" robot speed, i.e., the speed the robot was requested to move at, is compared with the speed it effectively did move at (the ratio of the distance from start to goal to the time it took to get there). The robot's effective and desired speeds are close when the latter are programmed to be 4 or 8 cm/s, but at 16 cm/s the robot actually travels only 75% as fast as requested, and at 32 cm/s only about a third as fast.

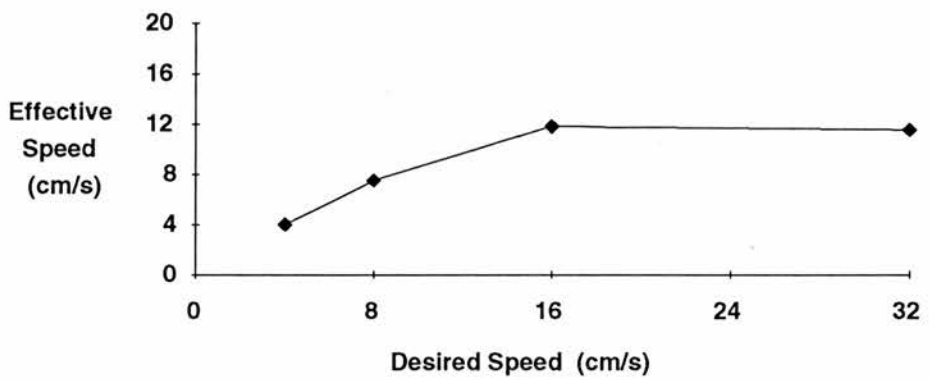

Figure 4.13. Effective speed (distance over time to destination) plotted against desired or "programmed" speed

It is not easy to construct a world in which the obstacles have a given density, so real robot runs do not validate this aspect of the simulation. However, it does seem to be the case that for one particular density (that of the field of cardboard boxes in the experimental pen considered), travelling faster within it means spending less time heading towards the target beacon relative to time spent circumnavigating obstacles. Since the effective versus desired speed curve flattens out, there is no point in trying to

travel faster than a certain speed. In this way, at least, the results of this practical test do agree with the model and simulations in Section 4.2.1.

Clearly we could have the robot travelling very slowly all the time, which would maximise the amount of time it spent navigating versus avoiding obstacles. But setting the speed is really an optimisation across conflicting variables, because it is easy to imagine tasks in which the robot should get to its destination as quickly as possible (at 8 cm/s the robot takes an agonisingly long time to get from one end of the laboratory pen to the other).

It therefore seems that basic locomotor and obstacle detection and avoidance competences which lead to robust, collision-free random walking are left wanting when the robot's motion is required to be goal-directed and fast. Instead, the tactics used in negotiating obstructions should be modulated by the direction of the target in situations like Figure 4.2; also, it appears that a greater proportion of the robot's time will be spent moving towards its ultimate destination, and the ratio of its effective speed to desired speed will be larger, if it moves slowly among obstacles.

4.3 Speed control

Approaches to robot design that have been labelled analytical in this document are typified by locomotor competences based on explicit representations of space, and trajectory planning. In principle at least, this notion of providing a robot with an internal map of the environment in which it operates effectively rules out the problems discussed in the previous section. Many efficient route planning procedures have been devised (see Section 2.2.1), allowing optimal trajectories between start and goal to be computed. Furthermore, this *a priori* knowledge of its path means that a robot can regulate its speed so as to progress smoothly, for example, slowing when executing turns, and accelerating in open spaces.

However, it remains difficult to deny that in reasonably variable environments, up-to-date maps are not an easy commodity to maintain, and without them, the advantages of prescience disappear. Therefore, it would be convenient if we could at least adapt the synthetic locomotor competences so that they are efficient rather than just robust. In this section, a transformation to the basic avoidance-navigation pair will be proposed that causes the robot to decelerate and move carefully in cluttered areas, and accelerate when its path is clear.

It will be shown that controlling speed in this way leads to improved performance in terms of the proportion of time the robot spends navigating versus negotiating obstacles. Nevertheless, an enhancement even to this scheme will suggest that the system would benefit substantially from some capacity for sensing or predicting what lies ahead between it and its goal.

4.3.1 An on-line (reactive) solution to speed control

Since both beacon tracking and obstacle avoidance express themselves through the motion of the robot, only one of these subsystems is in control at a time. If an obstacle is detected while the robot is busy trying to home in on the infrared beacon, the higher priority obstacle avoidance system takes charge and causes the vehicle to execute manoeuvres that will circumnavigate the obstruction. It is reasonable to assume that in areas of greater obstacle density, beacon tracking will be interrupted by obstacle avoidance more frequently than in open space.

Therefore, a simple reactive strategy for controlling speed suggests itself: incorporate some mechanism for monitoring the frequency of transitions between tracking and avoidance; when this frequency is small, the robot goes quickly, but when it is large, it slows down. In particular, define some function $f(t)$, such that

$$f(t) = \begin{cases} 1 & \text{if avoiding obstacles at time } t, \\ 0 & \text{if tracking beacon.} \end{cases}$$

This function is a rectangular pulse train; the pulses are of width t_c, and the period of the train averages $t_b + t_c$, assuming as in Section 4.2.1 that the mean time between collisions is t_b, and that collisions take t_c seconds to overcome. It can be shown that the average value of such a signal over an infinite time interval is[2]

$$\begin{aligned} \langle f(t) \rangle &= \lim_{T_1 \to \infty} \frac{1}{2T_1} \int_{-T_1}^{T_1} f(t)\, dt \\ &= \frac{1}{T} \int_{-T/2}^{T/2} f(t)\, dt, \end{aligned}$$

where T is the period of the pulse train (McGillem and Cooper, 1984). Here, this period is $T = t_b + t_c$, and the integral of the function over a single period is just t_c, so

$$\langle f(t) \rangle = \frac{t_c}{t_b + t_c}.$$

---

[2]McGillem and Cooper (1984) use the $\langle\ \rangle_T$ notation to denote the average value of the function enclosed in the brackets over an interval of length T; the subscript is omitted if $T = \infty$.

Taking the mean time between collisions to be $t_b = 1/vns$, as in the mean free path analogy, would give

$$\langle f(t) \rangle = \frac{t_c}{\frac{1}{vns} + t_c},$$

which may be rearranged to get

$$s = \frac{\langle f(t) \rangle}{t_c vn \left(1 - \langle f(t) \rangle\right)}.$$

In other words, assuming uniformly distributed objects of similar size, their density is proportional to the average value of the interrupt function $f(t)$.

So, the monitoring mechanism added to the obstacle avoidance and beacon tracking systems gets a binary signal indicating when avoidance is taking place; it samples this signal regularly, and determines its average value. From this value, the current object density can be estimated, and the motor controller of the robot is instructed to set a speed that is inversely related to this estimate. The best speed at which to travel for a given density could be determined experimentally.

As noted in Section 4.2.1, taking $t_b = 1/vns$ is only reasonable if the assumptions of uniform object density and size hold. However, it does not really matter whether or not taking $s \propto \langle f(t) \rangle$ is a good estimator of the actual, spatial, density of obstacles. Suppose that there are only two small obstacles in front of the vehicle, such as two of the cardboard boxes used in the runs described earlier, between which the vehicle could pass if it went slowly. If it starts by oscillating between them, s will grow because $f(t)$ is mostly 1; clearly the spatial density of the obstacles is not actually increasing (their size is not increasing relative to the floor surface), but by responding as if it were, i.e., by slowing down, the robot is more likely to find its way between them.

Such a speed control mechanism was implemented on the robot described in Chapter 3 as follows: an asynchronous process was run on the robot's controller which sampled the three forward IR detectors five times per second. If any of the three was detecting, the sample was assigned a value of 1, otherwise it was assigned a value of 0. Every three seconds, the fifteen samples collected were averaged, and the robot's speed was set to be larger the smaller this number, and vice versa, according to an empirically-derived mapping (see Section 4.3.3). The procedure is summarised algorithmically as follows:

1. sample the IR detectors 15 times, once every 0.2 seconds, such that the sampling function is

$$f(i) = \begin{cases} 1 & \text{if any IR detection occurring} \\ 0 & \text{otherwise,} \end{cases} \qquad i = 1,\ldots,15;$$

2. compute the average of the sampling function as an approximation to the density,

$$s \cong \langle f(t) \rangle_{15} = \frac{1}{15} \sum_{i=1}^{15} f(t); \text{ and}$$

3. request that the maximum speed of the robot be

$$v_{max} = \min\{ g(s), v_d \},$$

where v_d is the "desired" robot speed, and g is a function that maps density onto speed-to-travel.

A trace of a typical run with the robot using this simple speed control scheme is depicted in Figure 4.14. It is apparent that slowing down when the group of obstacles is encountered has resulted in a much less traumatic passage, with almost no oscillation between the boxes. On this run, the robot spent 64% of its time tracking the beacon, as compared with only 42% at the same speed but without speed control, and the effective speed (distance from start to goal over total time taken) has improved slightly from 11.6 cm/s to 12.4 cm/s.

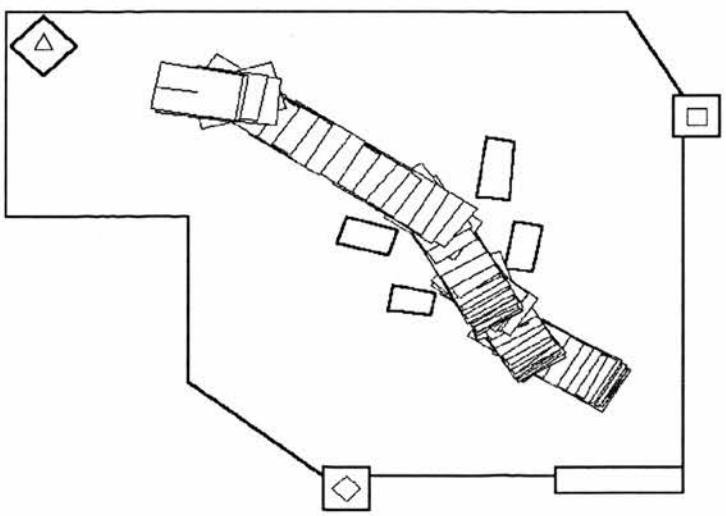

Figure 4.14. Trajectory for a desired robot speed of 32 cm/s, with IR-based speed control

4.3.2 Predictive control by long-range obstacle sensing

The disadvantage of the infrared collision avoidance system used in this way is that it only operates over a short range, so that the robot can only decelerate once it has been within an obstacle-rich area for long enough to have recognised this fact. An obvious improvement that is still effectively "on-line" is to provide it with a means of sensing impending obstacles before it reaches them, so that it can begin earlier to decelerate to a speed more appropriate to efficient obstacle avoidance.

Such an experiment was performed on the robot with its ultrasonic module (see Section 3.2.5) as follows: every 0.2 seconds, an ultrasonic chirp was generated, and the time of arrival of its first echo measured; this indicated the distance to the nearest specular reflector within the field of view of the sonar module. The nearest specular reflection was assumed to be the nearest obstacle. This assumption is not strictly valid, as the sonar signal will reflect from objects anywhere within a fairly large cone extending in front of the robot, even ones that are not in fact obstructions (i.e., do not lie directly in its path). However, it seems reasonable to suppose for safety that any close reflection might be an obstacle, and therefore to slow down in anticipation of it; and conversely, if there are no nearby reflectors, it might be worth trying to accelerate. In any case, the IR collision avoidance system (Section 3.2.3) prevents actual collisions with objects which might, for example, be oriented so that the ultrasonic signals do not reflect back from them to the robot.

In practice, this speed control mechanism was implemented as a monitoring process, as was the interrupt-frequency-integration process discussed in Section 4.3.1. Five times per second, it requests that the motor control constrain the robot's maximum speed according to a simple heuristic, namely

$$v_{\max} = \begin{cases} \min\{h(r), v_d\} & \text{if } r < \text{ threshold} \\ v_d & \text{otherwise} \end{cases}$$

where v_d is the "desired" speed (as before), r is the distance to the first specular reflector, and h is a function that maps distance onto speed-to-travel (see Section 4.3.3). If the robot detects any reflections within a specified minimum distance, it uses the distance to the reflector to determine how fast to travel.

The trace for a typical run with this system is given in Figure 4.15. As with the interrupt-based speed control procedure discussed above, the smoothness of the run benefits from the fact of travelling through the obstacles slowly, and accelerating in the

open space between them and the beacon. In fact, the robot shows almost no oscillation between the boxes, is now able to spend nearly 86% of its time navigating rather than negotiating obstacles, and has an effective speed of 19 cm/s.

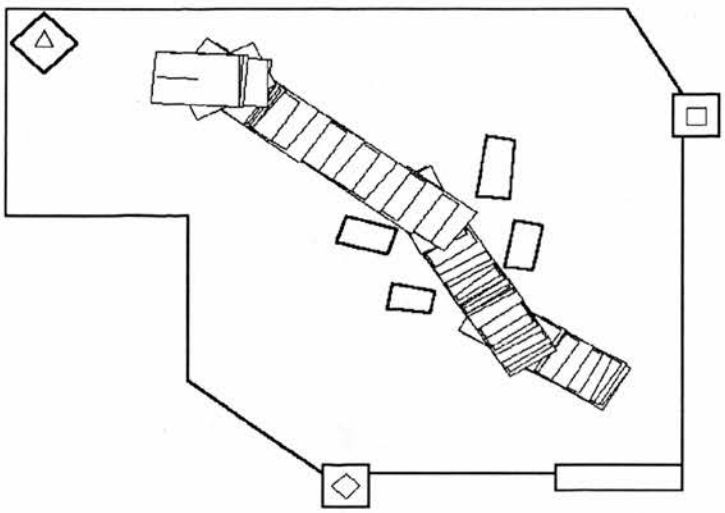

Figure 4.15. Trajectory for a desired robot speed of 32 cm/s, with sonar-based speed control

4.3.3 Comparison of results

Table 4.1 compares typical results for percentage time spent tracking the beacon versus avoiding obstacles and effective speed with and without speed control. It is clear that without speed control, the robot is as well to travel at 16 cm/s as at 32 cm/s, since its erratic behaviour when it is amidst the obstacles in the centre of the pen means that in both cases its effective speed is about 12 cm/s.

	16 cm/s, no speed control	32 cm/s, no speed control	32 cm/s, IR speed control	32 cm/s, sonar speed control
Time spent navigating (%)	58.3	41.9	64.3	85.9
Effective speed (cm/s)	11.8	11.6	12.4	19.0

Table 4.1. Comparison of proportional time spent navigating and effective speed with and without speed control mechanisms.

On the other hand, both simple speed control mechanisms afford improved effective speed at 32 cm/s, and in fact the predictive sonar-based method is over twice as good. The latter also allows the vehicle to spend proportionally more time trying to track the beacon, so that it appears to be a step in the right direction towards reasonably efficient locomotion, while still being basically on-line.

The functions g and h in Sections 4.3.1 and 4.3.2, respectively, map estimated density and distance to nearest reflector onto a robot speed that is suited to particular circumstances. In practice, these mappings were determined entirely empirically, that is, by running the robot many times and constructing a lookup table of speeds that led to the best average times spent navigating. They could, of course, have been learned on-line using connectionist techniques.

For example, the 86% time spent navigating for sonar-based speed control in Table 4.1 was obtained by the linear relationship $h(r) = (r / 45) \times 32$, with the distance threshold set to 45 cm. In other words, if the robot received a reflection at 45 cm or greater, then it set its speed to 32 cm/s. Any reflections closer than 45 cm led to a proportional decrease in speed. An interesting avenue for future research would be to attempt to develop a model, perhaps a refinement of the mean free path analogy, to the point where it became useful for evaluating the effectiveness of control laws which then could be transferred onto the actual robot.

The underlying effects of the speed control techniques are easy to see in Figure 4.16, which shows the pattern of transitions between beacon tracking and obstacle avoidance. When the trace for a particular speed control mechanism lies on the baseline, the vehicle's motion is being dictated by the beacon tracking system; when it rises above the baseline, the robot is avoiding obstacles.

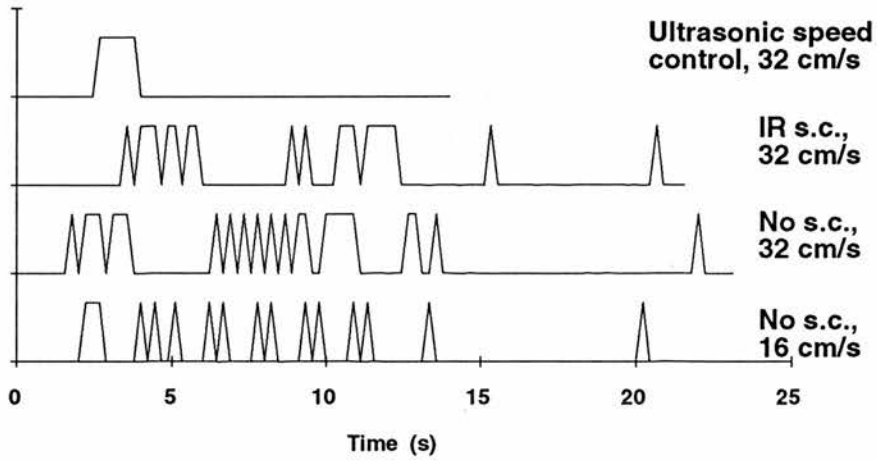

Figure 4.16. Transitions between obstacle avoidance and tracking; periods during which a curve is above the baseline mark when avoidance is in control

With no speed control active, the transition pattern shows the vehicle to be oscillating, as was apparent in the top-view graphics in Figures 4.9 and 4.10. At 32 cm/s desired speed, after the initial contact with the boxes the vehicle spends about 4 seconds turning back and forth without speed control; with IR-based reactive slowing

down (the curve immediately above), it oscillates just a couple of times. Observing the associated speed profile plot in Figure 4.17, it can be seen that the robot with no speed control reaches the obstacles at maximum speed and of course tries to maintain that speed; it fails, and is forced to reverse (the speed curve drops below the zero line). With reactive speed control, on the other hand, the robot decelerates on first contact with the obstructions, and this is sufficient to get it between the first pair of boxes. It then accelerates and slows again as it leaves the first pair and encounters the second pair of boxes, respectively; without speed control, it accelerates too rapidly away from the first pair, is again stuck between the second pair and forced to reverse once more before finding a clear path out and towards the beacon.

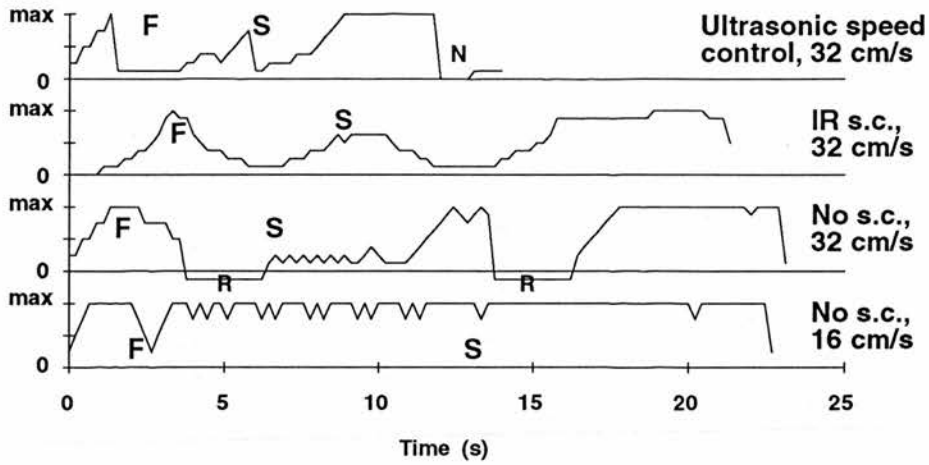

Figure 4.17. Speed profiles for various speed control options; the **F**'s mark the times at which the robot encountered the first set of boxes, the **S**'s mark encounters with the second pair of boxes; the small **R**'s indicate times when the robot was reversing (at **N**, the robot started moving again, triggered erroneously by some change in the beacon's signal)

The best performance of all obtains with the sonar-based predictive slowing down. In Fig 4.16, it can be seen that the contact with the obstacles occupies only one 2 second interval of the robot's time, leading up to which it has decelerated (see Figure 4.17). It does accelerate slightly away from the first pair of boxes, but slows again as it nears the second, with the result that it does not actually end up detecting them as potential obstructions at all.

Clearly, the robot's locomotion becomes much smoother if it uses either mechanism for regulating its speed in response to changes in the proximity of obstacles. Furthermore, being able to decelerate before reaching an obstruction is more effective than slowing down on first contact (sonar-based versus IR-based control in the previous section). However, the usefulness for this purpose of the ultrasonic module is subject to some limitations. For example, as mentioned previously, it is

possible that the orientation of obstacle surfaces will be such that either there is no reflection detectable at the robot, or its distance is incorrectly measured. Figure 4.18 shows two speed profiles for sonar-based speed control as before. On run 1, the robot successfully clears the second set of boxes and accelerates away; on run 2, however, this acceleration is interrupted by the detection of a specularity that was apparently not seen during run 1.

One way around this kind of problem is to use a different long-range obstacle sensor like a camera (Donnett and McGonigle, 1991). It would not be necessary to recognise actual objects: what the system needs to know is bounded by the nature of the locomotor task, namely the relative position of obstacles as it moves towards them on a potential collision course. Since the robot generates parallax information as it moves about, this kind of information can be profitably extracted from its visual input. Its vision system need therefore only consist of relatively simple motion-based depth sensing, successful implementations of which already exist (e.g., Hayes, 1989).

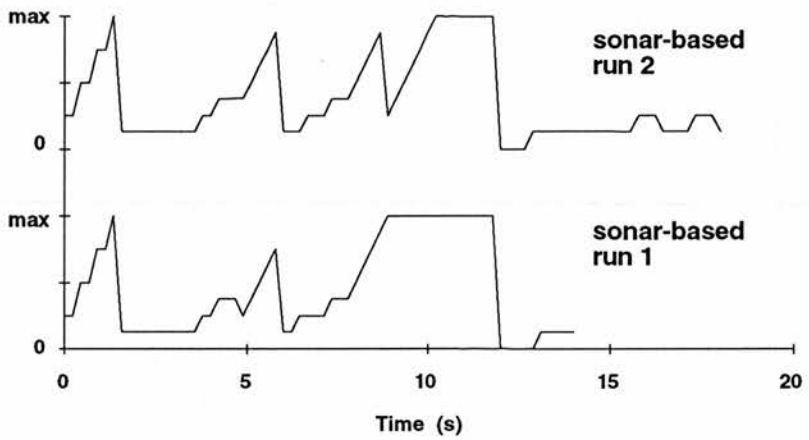

Figure 4.18. How speed profiles can differ between ultrasonic-based speed control runs

4.3.4 Predictive control by spatial means

It is also possible to use spatial strategies for adjusting speed predictively. The procedure is simply that the robot records the densities along its trajectory as it drives along given route; the next time it follows the same path, it can "look ahead" in the recording to determine what the density will soon become, and begin to set a speed appropriate to this anticipated density. Algorithmically, the operation proceeds as follows:

```
1. suppose the robot is following some fixed route
```

2. for every Δd metres of travel, it records its estimate of obstacle density (based on the average of the IR detection function, for instance, as in Section 4.3.1), so that

$$s(d_i) \equiv \langle f(t_i) \rangle,$$

where $s(d_i)$ is the density after $i\Delta d$ metres of travel, t_i is the time taken to travel that distance, and $\langle f(t_i) \rangle$ is the average value of the obstacle detection function over that interval

3. on subsequent journeys along the same route (and from the same starting point), set

$$v(d_i) = \min[G\{s(d_{i+k})\}, v_d],$$

where $v(d_i)$ is the maximum speed the robot should be moving at after having driven $i\Delta d$ metres, G is a function that maps density onto speed to travel, and v_d is the "desired" robot speed, as previously.

By recording the density it experiences in terms of the distance (rather than time) it has travelled along a route as in step 2 above, the robot generates a velocity-independent density "map" of the route. Subsequently (step 3), it can look ahead in this density map and find the obstacle density which lies ahead, $s(d_{i+k})$, where k is assumed to be positive, and set a speed appropriate to what the density will become. In fact, the function G need not operate on only the single density $k\Delta d$ metres ahead of the robot, but can take account of the range of densities $s(d_{i+1})$, $s(d_{i+2})$, etc., and could even be learned.

As an indication of the effectiveness of this kind of scheme, consider the beacon-approach with obstacles run depicted in Figure 4.19. This represents a simple experiment which deviates from the algorithm presented above in that the robot has merely been told to drive as fast as it can for about 2 seconds, then to drive slowly for 14 seconds, and then as fast as it can again. Here, the mapping and prediction are replaced by the robot programmer's knowledge of room layout and intuitions about how the robot should adjust its speed, rather than being performed explicitly by the robot itself.

The interrupt profile in Figure 4.20 shows that the robot spends only a small amount of its time avoiding obstacles (about 79%), and it has an effective speed of just over 14 cm/s.

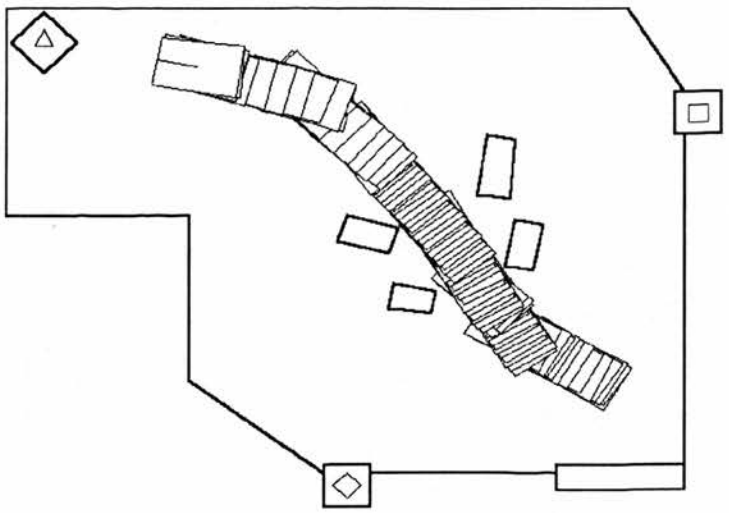

Figure 4.19. Trajectory for a desired robot speed of 32 cm/s, with pre-programmed slowdown for the period during which the robot is likely to be among the obstacles

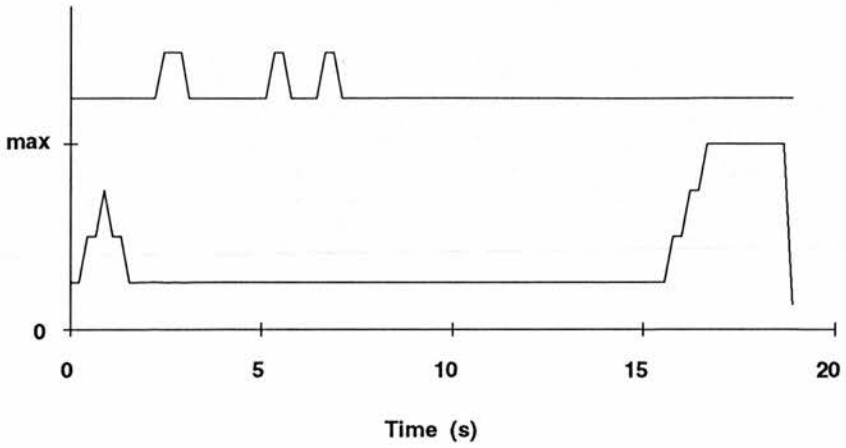

Figure 4.20. Interrupt (top) and speed (bottom) profiles for fixed slowdown run

In other words, even this very uncomplicated predictive spatial mechanism is better than the IR-based reactive scheme. Obviously, this approach is constrained in its usefulness, as it requires both that the robot always know where it is along a fixed route, and that the arrangement of potential obstacles along the way remain relatively fixed. Its density recording could be continuously adjusted to local variation as it drives around, but if objects in the world move around very freely, then the predictive benefit of the internal recording will be lost.

In fact, it could cross-correlate the pattern of densities it has sensed on its current run with the pattern stored from previous runs in order to determine its location along the route, although it would still need to follow more or less the same route from run to run (this kind of correlation-based mapping has been explored by Baader and

Hallam, 1992). However, localisation systems exist in which the robot is able to estimate its position anywhere within its operating space, and it is easy to conceive of techniques for associating estimated location with a best-speed-to-travel there (e.g., if the robot senses that it is in a corridor, it might try to move more quickly than if it senses it is in a crowded storeroom). In this way, at least the fixed-route constraint disappears.

4.4 Summary and conclusions

Ultimately, the only critical requirement of a mobile robot's basic locomotor competences is that they allow it to move around within its operating space without colliding with other objects in its vicinity. Robust locomotion like this can be implemented as an obstacle detection/avoidance subsystem that operates asynchronously, obtaining control of the vehicle's motion only when it is necessary to negotiate an obstruction. This synthetic philosophy of robot design contrasts with analytical approaches that require an explicit representation of the robot's environment and generate *a priori* action plans for moving from place to place.

When the robot is required to move from its current location to some specified goal in as short a time as possible, trajectory planning is advantageous in that optimal routes can be found and the robot's motion can be programmed so that it moves smoothly and efficiently from start to finish. However, pre-planning has the disadvantage that it depends on the layout of the robot's operating space remaining relatively fixed through time, and is adversely affected by unexpected obstacles.

On the other hand, we have seen that reactive systems based on an underpinning but impenetrable obstacle avoidance module may move robustly, but not necessarily efficiently. It seems that in order to achieve minimally deviating motion towards a goal, it is not sufficient simply to equip the robot's control system with an appropriate goal-seeking mechanism that sits atop the obstacle avoidance competence, and is interrupted whenever an obstacle is encountered. Instead, it is convenient if the strategies of obstacle circumnavigation take account of the robot's eventual destination. Furthermore, time-efficient motion towards a target benefits from contouring the robot's speed to fit the density of potential obstacles along its route; this can be accomplished by sensing the approach of obstacles from as far away as possible, or by means of even a very simple model of space, in which the density of

objects on the robot's route was recorded on past runs so that it can anticipate density changes before reaching them and adjust its speed accordingly.

These kinds of improvements appear to be at odds with the two core features of synthetic-based design approaches: the first is that existing, functioning competences (like obstacle avoidance) should not need to be modified as a result of future independent increments of the system's capabilities; second, systems should be engineered without explicit representations of the world. We would argue that it is acceptable to disregard these principles if the result is efficient locomotion with robust obstacle avoidance and smooth speed adjustment to layout, without the need for fixed starting points or routes, and where objects are allowed to migrate freely.

Presumably the most potent approach would be to amalgamate aspects of synthetic and analytic design: the former would furnish on-line control, robust to random changes in the environment; from the latter would be derived the predictive power of having explicit representations of operating space. Therefore, we turn our attention in the next chapter to questions of localisation in space.

Chapter 5

Qualitative Map-Based Spatial Localisation

In the previous chapter, it was argued that fitting a mobile robot's motion to the configuration of objects in its vicinity required either the ability to assess the layout of the robot's operating space from a distance, or some form of layout memory plus a locative sense as a mechanism for aligning remembered layout with current position. However, the usefulness of spatial memory and localisation obviously extends far beyond the limited role it plays as a device for estimating how best to move within a field of potential obstacles. Indeed, a free-ranging robot's principal feature is its mobility, so that most of the tasks it might reasonably be expected to perform will require that it visit particular distinguished places.

Of course, the robot could be allowed to wander randomly, only performing operations relevant to being in a given position if it should reach the position by chance. Alternatively, the locations of importance to the robot could be fitted with landmarks or beacons detectable from anywhere, so that the vehicle might approach a given position by means of a simple tropism (Section 4.2). However, contriving such an arrangement might either be impossible (e.g., where the vehicle's range extends throughout several rooms, separated by sensorially impenetrable barriers like walls, or even within a single room containing large occluding obstacles), or undesirable (e.g., the vehicle might be expected to visit such a large number of positions that marking all of them would be excessively difficult or expensive).

In Chapter 2 it was noted that both analytic- and synthetic-type approaches to robot design support spatial capabilities. Analysis-based systems usually involve geometric maps as models of space and quantitative treatments of the capabilities of sensors, with careful attention paid to the degree of uncertainty in the robot's estimates of where it is. The few synthetic-type efforts towards spatial control are based on qualitative treatments of space, and heuristic models of sensing.

This chapter describes an attempt to treat localisation qualitatively with minimal sensor models. It diverges from existing synthetic-qualitative efforts in two ways: first, the accessible range of the robot is not constrained by the geometry of its environment

(i.e., the robot is not forced to follow walls or other objects); and second, it borrows some mathematical machinery from the traditionally analytic approaches in order to quantify the performance of individual systems and combinations of them.

The chapter begins with a brief recapitulation of the spatial competence discussion in Section 2.3, and outlines the particular view of qualitative localisation embraced here. The second half of the chapter presents the procedures for data collection and analysis, the results of which are given in Chapters 6 and 7.

5.1 Quantitative and qualitative treatments of space

It was noted in Section 2.3 that analytic-type approaches to endowing robots with spatial competences are generally based on metric representations of space, while synthetic approaches use only qualitative spatial maps. This section reviews what localisation means in these two disparate frameworks. It will argue that the qualitative basis is appealing, but that existing implementations suffer in two limiting ways; a method which overcomes these limitations will be proposed.

5.1.1 Quantitative localisation revisited

In metric approaches, the robot's operating environment is usually defined by a two- or three-dimensional Cartesian coordinate system, and the position of the robot and other objects is specified in terms of this reference frame. The system's "map" is a list of the coordinates of some potentially recognisable features. These features might be active beacons (e.g., Durieu *et al.*, 1989), or they might be facets of passive objects detectable by the robot's sensors (e.g., specular-reflection-producing shapes on objects detected ultrasonically; see Hallam, 1985, for instance).

The localisation problem under these circumstances is to determine the positions of such recognisable features relative to the robot, and thereby to determine the location of the robot in terms of the coordinate system attached to its operating space. Attempts to realise localisation in this way acknowledge two major concerns:

1. the computed robot position can only be as accurate as the estimated positions of the sensed locative references; and

2. these position estimates, in turn, can only be as accurate as the sensory system and the process for interpreting its sensations allow.

Robust analytical procedures have been developed for propagating the uncertainties of localisation operations from sensing through final position estimation, such that the uncertainty of the estimate is known provided uncertainties in the sensing process are also known (Durrant-Whyte, 1988a, b; Smith and Cheeseman, 1986; Smith *et al.,* 1988). In fact, optimal position estimates can be computed (i.e., which have minimal ultimate uncertainty for given initial uncertainties).

The lynchpin of these procedures is the robustness of their underlying sensor models. The synthetic school therefore rejects such approaches as an article of faith, as this quantification of sensor characteristics is considered to be both difficult to do reliably, and unnecessary in practice. This argument is easy to support within most of the existing synthetic enterprises: the kinds of sensors that are used tend to be selected for low cost and minimal manufacturing effort, and such devices are probably the most difficult to obtain good models for.

5.1.2 Qualitative localisation and "strip-maps"

The alternative to quantitative techniques is to treat the robot's operating space and sensing systems as qualitative devices. As discussed in Section 2.3, space is represented by things like adjacency relationships (e.g., Mataric, 1990), rather than metrically as a Cartesian coordinate frame. The interpretation of sensor data is based on heuristics (e.g., Flynn, 1988), rather than on quantitative device characteristics. The contention is that although it is easy to imagine large-scale tasks for which quantitative mapping is essential (e.g., aerial or naval navigation), a qualitative representation of space is sufficient to underpin the activities of a robot operating within the small-scale space of a typical laboratory or office building.

This argument is compelling, especially since robust spatial behaviour has been demonstrated in synthetic-type systems like those of Connell (1989), Mataric (1990), and Nehmzow *et al.* (1989). However, these systems require some form of guiding geometric constraint in the environment (see Section 2.3.2). That is, they follow path-creating features like walls, and can localise only along fixed routes. The behaviour that emerges from their interactive architectural components is stereotyped as a result. For instance, they may have no mechanism for exploiting shortcuts like cutting across an area of open space; as with goal-directed locomotion, they are able to perform robustly but not necessarily efficiently.

5.1.3 Qualitative localisation and topological maps

The overwhelming advantage of existing mapping schemes is that they do not (necessarily) restrict the robot's usable domain to geometrically-bounded channels. However, this is not to say that a qualitative spatial sense must be path-based. In essence, all that is required for qualitative localisation is a sensor which is able, by some unspecified means, to partition its operating space into k distinct, unique, reasonably small regions; a robot with such a device can determine precisely which of the k regions it is in, and therefore localise itself uniquely anywhere within its operating domain (to a resolution of k places)[1].

Strictly speaking, such a qualitative map for localisation alone need not even preserve topological relationships, as the robot is able to determine where it is in terms of its k distinguishable regions independently of whether or not it knows anything about adjacency or other relationships between these regions. However, if the ultimate role of localisation is to underpin navigation, and since the latter requires some means for interrelating different parts of space, then at least adjacency relationships should be preserved.

Physical basis

In the context of robots controlled by digital computers, with sensory devices as they are currently engineered, it is relatively straightforward to devise such a qualitative mapping scheme. Almost invariably, "sensors" are electromechanical devices that transduce an extrinsic physical quantity into a form accessible to programs running on digital computers internal to the robot. The last stage of the transduction operation is usually a conversion of the sensed quantity from some analogue form (such as a voltage that varies with changes in the sensed parameter) into a quantised or digital value (a number).

At this level, all sensory systems are qualitative, in that they simply associate position in space with one or more rational numbers. This association partitions the robot's operating space into up to k regions, where k is the number of distinct values into which the sensed quantity may be transformed. In the context of this work, sensors become quantitative only when the numbers they produce are assigned some

---

[1]Qualitative maps can also be constructed from the adjacency relationships between otherwise indistinguishable regions; this is discussed in section 5.2.1 as dynamic mapping.

interpretation with respect to some geometric aspect of the robot's surroundings. In a sense, any attempt to tackle sensor data in such an uninterpreted fashion is "pre-modelling"; that is, insights derived from this process could facilitate the establishment of quantitative sensor models.

Performance quantification

The function of modelling and sensor data interpretation schemes is to extract higher-order attributes of a sensed quantity from the spatio-temporal variations of the digitised values representing the quantity. This suggests that using statistical measures of either minimally processed or raw (completely unprocessed) sensor data values is a reasonable way of characterising, in metric terms, the in-principle limitations on the accuracy which an otherwise qualitative system might hope to achieve in estimating where the robot is. Performance has not been analysed like this in existing qualitative approaches (as Durrant-Whyte, 1988b, argues); rather, systems are subjected to proof by example, and any putative successes or failures are presented as "it works" or "it doesn't work".

Some means of performance evaluation seems particularly appropriate to the problem of how to combine information afforded by multiple sensor systems into a single estimate of where the robot is. Robust quantitative methods have been developed (e.g., Durrant-Whyte, 1988b). However, corresponding qualitative analogues are somewhat limited (e.g., Flynn, 1988). Unless the behaviour of a system is to be assessed entirely by trial and error, it would be beneficial to know, even for a qualitative approach, whether adding a further locative reference beacon (say), enhances the system's ability to localise or confuses it (and is thus detrimental to performance).

5.1.4 Summary and objectives

Qualitative, topological map-based localisation should allow a robot to determine its location almost anywhere within its operating domain (to within an area of space such that all the points in it are indistinguishable), from minimally interpreted sensor data. The objective of the work described in the next section is to develop such an approach, but which in addition to being inherently qualitative, is nevertheless amenable to quantitative assessment. This assay will contribute both to the measurement of performance given only a single sensory modality, as well as when information from multiple modalities is available.

5.2 Qualitative mapping and localisation

The proposal is to treat localisation as a qualitative operation relating position in space to one or more integral values, and to evaluate systems by studying the numerical characteristics of the "fields" produced by aggregating the values associated with different parts of the robot's operating environment. The robot described in Chapter 3 is equipped with three sensory systems which appear to be suitable for a qualitative localisation scheme which correlates position and raw sensor data. These are the sonic signal amplitude subsystem (Section 3.2.4), the infrared beacon detector (also Section 3.2.4), and the ultrasonic module (Section 3.2.5).

These systems will be deployed with the robot running in a realistic environment. Given the particular qualitative mapping and localising strategy to be outlined below, it is hoped that what each system is able to tell the robot about where it is will be discovered. In particular, attempts will be made to establish two things: first, how certain each system is of where it believes the robot to be; and second, how accurate these estimates are. Furthermore, it will be necessary to consider questions of how to combine individual systems, and what effect combination has on the certainty and accuracy of localisation.

5.2.1 Mapping and localising

The mapping activity of the robot essentially consists of driving through its operating environment in some principled manner, and recording the patterns of sensory data it encounters along the way. This recording of sensed data constitutes the robot's spatial "map". On subsequent journeys through the mapped domain, the robot localises by attempting to match the sensory inputs it is currently receiving with stored patterns. This matching operation is intended to establish where the robot is relative to the map it has constructed on previous journeys, not with respect to some arbitrary, external, world-centred coordinate system.

"Static" versus "dynamic" mapping

This kind of qualitative mapping can follow two distinct tracks. As the robot moves through its environment, it can either record absolute sensor values at specific positions, or it can record the positions at which there is a transition between sensor values. The former is a static approach, in that the robot records the sensations it

experiences at certain stationary positions. The latter approach is dynamic, as the robot must by definition be moving in order to detect sensor input transitions.

Dynamic mapping intuitively fits much better into the general qualitative frame of mind, for the following reason: in a qualitative scheme where space is segmented by raw sensor data, any region within which the robot receives the same input everywhere is an indivisible "unit" of space for it; the important event occurs only on entry to or exit from this unit[2]. With static mapping, the robot may record sensory inputs within such a monolithic region several times. Its map will then contain a number of places for which the sensory data are indistinguishable, and if on subsequent localisation attempts it encounters the same sensation, it will discover that it might be at any of several places previously visited. The static map contains redundancy, and is thus less directly coupled to the sensory qualia than the dynamic map.

The dynamic approach has been explored in relation to the construction of environment models from patterns in the structure of visual information experienced by a moving, sighted agent by Koenderink and van Doorn (1977), and is the basis for Gibson's theory of visual perception (Gibson, 1950, 1979).

While compelling, the fact that it possibly represents the ultimate attainment of a purely qualitative spatial sense makes it somewhat unsuitable to the performance assessment aspect of the present work: in the limit, a correctly implemented dynamic system of this type will be able to determine how many sensory "states" there are in the robot's world, and thus how many regions it is able to distinguish; however, it does not characterise the physical extent of the regions. For instance, a particular sensory system may allow the robot to identify 256 different areas; but if one of these zones occupies 99.9% of its operating space, and the other 255 very finely decompose the remaining 0.1% of its space, then this system might be less useful than one which is only able to segment space into 20 parts of equal size.

Of course, the desired performance analysis could have been performed by supplementing qualitative mapping with some mechanism for recording the metric coordinates of the sensory transition points (with the dead reckoning system, for instance); this mark-up information would have allowed the dimensions of sensory zones to be determined and analysed *a posteriori*. However, the work discussed in this

---

[2]For navigation purposes, it is presumably also important to know the specific directions of ingress and egress.

chapter takes the more direct route by espousing the static approach, in which the robot drives through its environment, sampling and recording from its sensory systems at arbitrary, fixed positions, paying no attention to the fact that positions at which sensory data are the same will subsequently be indistinguishable. The next section details the specific procedure adopted.

5.2.2 Experimental Procedure

The robot discussed in the second half of Chapter 3 is driven along a trajectory that covers exhaustively the free space in a pen about 10 square metres in area (see photograph in Figure 5.1). The pen actually consists of a fenced-in area, roughly L-shaped, within a larger room. This room is typical of an office environment, with desks, chairs, filing cabinets, and bookshelves. The robot's pen remains clear of these things, and contains instead four cardboard boxes clustered near its centre. These objects serve to partition the manoeuvrable area in the pen, and are a concession to the limitations of the robot's obstacle avoidance sensors, which occasionally fail to detect narrow features like chair and table legs.

Figure 5.1. Photograph of the experimental environment

No other contrivances are used to simplify the tasks of moving and localising for the sensory systems:

1. sonic localisation is underpinned by three loudspeakers located in a triangle, each driven by a simple tone generator, such that the three beacons produce different, distinguishable tones; no effort has been made to reduce background noise, and the system is thus subject to interference;

2. the infrared beacon system is served by three IR sources mounted on the sides of the loudspeakers, pointing approximately towards the middle of the pen; no effort has been made to measure the precise fields of view of these sources;

3. finally, the ultrasonic system suffers most from the non-toy-world nature of the room, in that it inevitably detects confusing reflections from the many surfaces of objects both within and outside the pen itself.

Figure 5.2 shows the layout of the pen diagrammatically. The lines emanating from each speaker approximately delimit the field of view of the three infrared beacons attached to the sides of the speakers. The dots mark positions at which sensor data was collected; these positions, 166 in total, are about 20 cm apart (23 cm grid diagonals, 16 cm grid sides), and the whole pen is about 3 m long by 2.5 m wide. Four rectangles in the middle of the pen represent cardboard boxes added to subdivide the space.

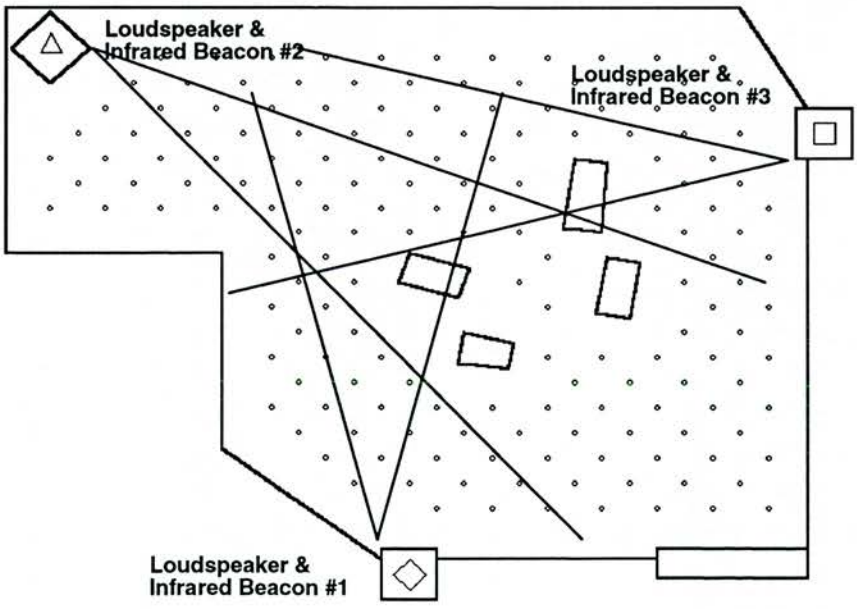

Figure 5.2. Layout of the experimental environment; see text for explanations

Data collection

Sensor data collection proceeded as follows: at each of the dot-marked positions in Figure 5.2, the robot stopped, then rotated through 360°. Every 22.5° (in other words, 16 times per rotation), starting from the same reference direction at each position, and the sonic, infrared, and ultrasonic fields were recorded. The sample data was stored in a list, indexed by the unique integer label assigned to the recording position. With the sonic and infrared systems, data were collected for each of the three reference beacons in succession. This mapping process was repeated three times, with the robot visiting (under operator supervision) each of the 166 distinguished positions.

The position grid coincided with the tiling pattern on the floor of the pen, so that on subsequent visits, it was possible to ensure that the robot was collecting data at nearly the same 166 places. However, there were inevitably slight differences between the exact sampling locations from run to run (see also systematic error discussion in Section 5.2.4). At the end of this acquisition phase, the robot had internalised three qualitative maps of its environment for each of the three sensory modalities. Table 5.1 summarises the form taken by these maps for each modality.

Modality	Data Format
Sonic beacons	for each beacon, the intensity of the received signal for each of the two microphones, at each position, and in each of the orientations
Infrared beacons	for each beacon, the perceived relative bearing and intensity of the IR source, at each position and in each orientation
Ultrasonic module	the intensity of sonar pulse reflections in each of the 16 orientations at each mapping position, for a 30 ms period immediately following pulse emission

Table 5.1. Format of collected sensor data

Motivation

The chosen arrangement of 166 sampling points is motivated in two ways[3]. First, they cover the robot's operating space exhaustively. Second, while the mapping takes place

[3]Actually, the third reason for the choice of these points is that they happen to coincide with corners in the wooden tiling of the laboratory floor; as the tiles were very regular and tightly joined together, this pattern was likely to be much squarer and more precise than a hand-drawn grid would have been.

at fixed positions, subsequent attempts to localise could take place with the robot positioned anywhere; as these points are less than one robot-width apart, if the robot is able to ascertain the nearest of these positions, then it is true to say that at least part of its body is directly over this nearest position. This issue is discussed more fully later.

The decision to record from only 16 orientations per position was motivated by data volume considerations, as the robot's maps were required to be maintained within its on-board memory facilities. The ultrasonic recording time of 30 ms was chosen because sound, in this interval, travels an out-and-return distance of roughly 5 m, which is about the length of the room the robot was operating in.

Data reduction

In fact, to render the analysis more tractable, it was decided to simplify these maps considerably by means of the following data reductions: for each position, the data for the two microphones in the sixteen orientations were found to be quite similar, and therefore these thirty-two numbers were averaged into a single, representative "intensity" value; for the infrared system, the narrow visible field of IR sources meant that if any of these sources was visible, then in it could be seen at only one orientation, therefore only this orientation was recorded, along with the intensity of the perceived signal in that direction; finally, since 30 ms of sonar scanning in each of sixteen directions produces a large amount of data (about 8000 bytes per mapped position in the present implementation), the sonar maps were simplified by computing the distance, in each of the sixteen directions, to the nearest significant reflection. Table 5.2 summarises the form taken by the sensory maps after these simplifications.

Modality	Reduced Data Format
Sonic beacons	for each beacon, a single representative intensity at each position
Infrared beacons	for each beacon, the orientation at which it is perceived to be, and its intensity in that direction, at each position
Ultrasonic module	the distance to the first sharp reflection after pulse emission in each of the 16 orientations, at each mapping position

Table 5.2. Format of reduced sensor data

Summary

In summary, the robot drives around a pen with three loudspeakers, and three infrared

light sources, and records sensor readings from its microphones, infrared beacon detector, and ultrasonic module at 166 positions closely covering its operating domain. These sets of data form qualitative maps per sensory modality. Sonic-signal-intensity field maps consist of a list of 166 values; infrared-beacon-direction field maps comprise 166 pairs of values; and finally, ultrasonic-first-reflection-distance field maps consist of 166 16-tuples of values. Such a triple of maps was produced for each of three robot journeys; henceforth, these three map triples will be called "set 1", "set 2", and "set 3".

5.2.3 Probability-like basis for position estimation

As the robot moves through its environment, qualitative maps of this space are constructed for each sensory modality. Subsequently, wherever the robot happens to be positioned within its operating domain, localisation requires that it perform some matching operation between the data sensed at the current position and the set of stored (map) values.

Map-building

As described previously, the robot's operating space is treated not as a continuum, but rather as a mesh of 166 discrete, labelled regions, centred on the 166 data collection locations. Any two physical points within the same region are said to be in the same named position. The question of the effect of continuity versus discreteness will be returned to in Section 8.1.2, but for the moment it is assumed that this is a valid decomposition as the mesh is reasonably fine, so that performance in discrete sensory space is likely to be representative of performance in continuous space. The robot's position anywhere in its environment can be specified by a unique integer between 1 and 166. Here, the robot's 166 sampling positions (and the regions surrounding them) will be labelled $A_1,\ldots,A_{166}$.

Map-building in this scheme proceeds as outlined earlier: the robot visits the 166 distinct places that comprise its environment, and at each of these records some sensed parameter B. Provided it has a mechanism for associating these values with some position index A_i, it need not in practice store the Cartesian coordinates of its position, although as discussed earlier, if the map is ultimately to be used for navigation, then at least the adjacency relationships between named positions ought to be preserved. For the purposes of the spatial analysis to be presented here, however, it is convenient to retain physical coordinates, as this allows localisation errors to be expressed in concrete units of distance such as centimetres.

For the sake of this discussion, and without loss of generality, the space within which the robot operates is assumed to be a Cartesian plane. Any point in this space can thus be represented by a two-dimensional vector $\mathbf{x} = (x_0, x_1)$. The plane is continuous, and a robot's position is uniquely defined by choosing an arbitrary, fixed reference point on its body and specifying the Cartesian coordinates of this point. With an asymmetric robot, it also necessary to specify its orientation about some axis, with respect to a reference direction (this applies to a symmetric robot as well, of course, unless the reference point is at its centre of rotation). Initially, however, orientation will be ignored, and "place" will be treated as a two-dimensional quantity only (Section 8.1.4 discusses the inclusion of orientation).

Localisation

Once the acquisition or mapping phase is complete, the robot may be positioned at an arbitrary place anywhere within the mapped domain. For the moment, it will be assumed that this place coincides with one of the grid positions, but in Section 8.1.2 it will be shown that the robot is able to localise at off-grid positions almost as well. At the robot's unknown position, suppose that it senses some condition B, and that it has a map $B(A_i)$, $i = 1,\ldots,166$, which contains the sensory data recorded previously at the 166 sampling positions. The problem is to determine which position A_i the robot is currently at, by matching stored data against the current sensor reading.

A very strict matching function would say that if $B = B(A_i)$, then the robot is at position A_i, and is somewhere else otherwise. In other words, the only positions at which the robot could possibly be are ones where there is a perfect match between mapped and sensed data. However, real sensors and the extrinsic physical quantities they measure are inevitably prone to noise, so such a strict matching function would be all but useless: there might either be no position that matches exactly, or there might be many. Instead, it is more appropriate to choose a matching function that somehow produces a graded response.

In fact, the requirements of the task effectively constrain the type of data matching function that is appropriate. The match function must map the difference between the mapped and current sensory states onto a number that quantifies the goodness of match. It must return a large value for close matches, and a monotonically decreasing value for increasingly bad matches. Its strictness should also be adjustable to allow it to compensate for natural variability in the sensor data that might arise even under comparable physical circumstances.

Furthermore, where more than one pair of mapped/current data are to be compared, it would be convenient if the value of the match function as applied to the sum of the differences in each pair were equal to the product of the values of the match function applied to each pair individually. The usefulness of this feature will be justified in Section 5.2.5; for the moment, suffice it to say that the multiplicative effect makes it possible to combine position estimates made independently by individual sensors, rather than having to merge the separate systems into a composite sensor, and then generating a combined estimate. These constraints define an exponential function like

$$f\{B, B(A_i)\} = e^{-\{[B-B(A_i)]/\alpha\}^2},$$

which takes on values between 0 and 1, with 0 indicating worst- and 1 best-possible matching. The parameter α controls the strictness of the matching operation, such that with α large, any mapped value will match any sensed value with $f = 1$, while as α tends towards zero, only perfect matches have non-zero magnitude (in the limit, this is identical to the unrealistically strict matching function alluded to two paragraphs ago). Although the basic form of this function seems the most suitable for the task, Section 8.1.1 discusses the question of whether or not variations on it would have a significant effect on the overall results of the performance assessments to be presented in Chapters 6 and 7.

Quasi-probabilistic view

In a loose sense, the function discussed above is computing the probability of being at a position A_i by matching the data sensed at the current position against stored (map) value for that position. This is not a probability in the *objective* sense of "frequencies of occurrence". That is, if the mapped data for position A_i matches current data B with a value of 0.5, for instance, this does not mean that if the robot senses input B 1000 times, then on around 500 of those times it was actually at A_i. Instead, the match value represents a *subjective* probability (Winkler, 1972), in the bookmaker's odds sense. This notion of probabilities is widely used in circumstances where numbers are probability-like in behaviour, even though they have no frequency interpretation.

Bearing the above caveat clearly in mind, it is possible to think of a localisation attempt as generating a set of (subjective) probabilities of being at each place in the mapped environment as the match function is repeatedly computed for each of the 166 A_i's. The advantage thus conferred is that it becomes possible to apply the analytic methods of probability theory in comparing the performance of individual systems and combinations of them. The results presented in Chapters 6 and 7 will bear witness to the plausibility of this approach.

Bayesian interpretation

Ultimately, what this probability-type stance is expected to deliver is a procedure for deciding where the robot is most likely to be, given the current sensed data, and the record of sensory values at each of the robot's potential locations. It is required to generate the set of probabilities that the robot is at position A_i, $i = 1,\ldots,166$, given that it is sensing some value B. More formally, suppose that random variable $\tilde{A}$ represents the position the robot currently occupies, and random variable $\tilde{B}$ is the value of some sensed parameter; then the notation $P(\tilde{A} = A_i | \tilde{B} = B)$, $i = 1,\ldots,166$, denotes the probabilities that the robot is at each of the 166 positions A_i, given that sensed parameter $\tilde{B}$ currently has value B.

Intuitively, the likelihood that the robot is at some position given what it is currently sensing must be related partly to the likelihood that it would be experiencing the particular sensation at that position, and partly to the likelihood of being there in the first place. In fact, since the set of A_i's is a disjoint, exhaustive partition of the robot's operating space, it is reasonable to compute the desired position probabilities by Bayes' Rule for discrete probability models (Winkler, 1972):

$$P(\tilde{A} = A_i | \tilde{B} = B) = \frac{P(\tilde{B} = B | \tilde{A} = A_i) P(\tilde{A} = A_i)}{\sum_{j=1}^{n} P(\tilde{B} = B | \tilde{A} = A_j) P(\tilde{A} = A_j)}, \qquad i = 1,\ldots,n.$$

This relationship simply states that the probability that some random variable $\tilde{A}$ takes on value A_i, given that another random variable $\tilde{B}$ has value B, is equal to the conditional probability that $\tilde{B} = B$ when $\tilde{A} = A_i$, times the unconditional probability that $\tilde{A}$ takes on value A_i in the first place; the denominator on the right-hand side serves as a normalisation so that $P(A_i | B)$, $i = 1,\ldots,n$ (where A_i abbreviates $\tilde{A} = A_i$ unambiguously, as B does $\tilde{B} = B$), is a probability density function, that is,

$$\sum_{i=1}^{n} P(A_i | B) = 1.$$

In other words, provided its underlying assumptions about the A_i are respected, Bayes' Rule can be used to derive the likelihood of being at each position on the grid given a certain sensed condition B, by computing (and normalising) the probability that condition B would in fact be sensed at that position, times the probability of being in that position in the first place.

The problem is to find these latter two probabilities. Now, $P(B | A_i)$ is the probability that sensed condition B occurs when the robot is at position A_i. This

conditional probability is properly derived from the physical characteristics of the robot's sensing system and the extrinsic quantity it is sensing, since it is the relationship between these two that determines the likelihood of a particular sensation at a given position. However, in recognition of the difficulty of expressing this physical relationship correctly, and in the spirit of qualitative localisation, a different strategy has been employed.

Specifically, it is assumed that the sensory field is stable between robot visits; that is, if the robot returns to the same position several times, it will experience nearly the same sensory inputs. If this assumption is made, then the likelihood of getting a particular input B at position A_i is determined by looking back at the previous inputs sensed in that position, and comparing them. So, given that the robot has in its map some value $\tilde{B} = B(A_i)$ at that position, $P(B|A_i)$ is taken to be a function of the likelihood that B and $B(A_i)$ are the same; and, for reasons discussed earlier the chosen match closeness function is

$$P(B|A_i) = e^{-\{[B-B(A_i)]/\alpha\}^2},$$

where α controls the strictness of the matching operation (again, with large α, any mapped value matches any sensed value with a likelihood near 1, and as α approaches zero, only perfect matches have a non-zero probability). The strictness parameter allows the effect of match tightness on the accuracy of localisation to be tested.

This function is clearly not the conditional probability it is masquerading as, and can only be used if the assumption of reasonably stable sensory fields holds. In turn, this assumption holds for beacon-based localisation systems (here, sonic and infrared) only if the beacons remain fixed relative to the robot's environment, and for sensed-feature localisation systems (here, ultrasonic) only if the locative reference objects in the environment do not move. The implications of this are discussed in Section 8.1.3, where it will it be shown that there are conditions under which it is safe to relax these assumptions.

The distribution $P(A_i)$, $i = 1,\ldots,n$, represents the *a priori* probabilities that the robot finds itself at each of the n (= 166) positions. At worst, it can be assumed that the robot might be anywhere within its operating space, so that $P(A_i) = 1/n$, $i = 1,\ldots,n$. In this case, Bayes' Rule becomes

$$P(A_i|B) = \frac{P(B|A_i)}{\sum_{j=1}^{n} P(B|A_j)}, \qquad i = 1,\ldots,n.$$

This choice of prior probabilities is rather pessimistic, as it views the robot's

position through time as varying unpredictably. In practice, of course, once a localisation decision has been made, any localisation attempt following shortly afterward can exploit the known adjacencies of positions to bias the computation in favour of the likelihood that the new position will lie close to the previous position. However, this issue is ignored for the time being and the analysis assumes the worst-case of no prior knowledge of location. Section 8.1.5 discusses localisation with non-uniform priors.

Deciding on a single estimate of position

The locative competence proposed here consists in the robot sampling the data field of its sensor systems and constructing from the sensory inputs it collects a qualitative map of its environment. On subsequent journeys through its domain, the robot compares sensory inputs it is currently receiving with those in the map. Subject to a number of assumptions outlined above, it attempts to estimate the (subjective) probability that it is at each of the mapping positions; this is done by determining the likelihood that it would be receiving the current data at each position, given what it received at these positions previously, and assuming that the sensory field is relatively unchanging. This set of likelihoods, once normalised, will be called a "position probability distribution", although this is perhaps a loose use of the terminology.

Given a position probability distribution with the robot at location A_i, and receiving some sensor input B, there are at least two obvious ways in which the robot might arrive at a single estimate of its location:

1. the first of these is the Maximum *A posteriori* Probability position, or MAP, which is just the position A_k for which $P(A_i|B)$, $i = 1,\ldots,166$, is maximum (or one of these positions, should there be more than one);

2. the second requires the association, discussed earlier, between position indices A_i, and their spatial coordinates in some reference frame. Suppose that the coordinates of A_i are given by vector $\mathbf{x}_i$. Then the robot can compute the "spatial mean" of the probability distribution; that is, it averages the coordinates of the set of positions at which it might be, weighting the coordinates of each position by the probability of it being there. This expression is called the *conditional expectation*, and is computed here as

$$E[\tilde{A}|B] = \sum_{i=1}^{166} \mathbf{x}_i \, P(A_i|B).$$

Which of these two estimates is the best to use for a particular sensing system

remains to be seen. Note that the first of these specifies the robot's position as the index number of a mapping location, while the second specifies it as coordinates in some arbitrary frame. To compare the two, either the grid position must be mapped to its spatial coordinates in the same frame as the conditional expectation, or the latter must be mapped to the index number of the nearest on-grid position. The former alternative is adopted, as this allows the difference between the two estimates to be quantified metrically; in this case, the coordinates of the MAP position can be thought of as the "spatial mode", while the conditional expectation is the "spatial mean".

An example

At this point, it seems reasonable to apply the procedure proposed above on the robot in order to obtain an initial feeling for whether it is a plausible approach. Suppose the robot described in Chapter 3 operates only with its sonic amplitude subsystem (Section 3.2.4), and that in the experimental pen discussed previously (Section 5.2.2) a single tone generator is active. The robot maps the sonic field of the generator by driving around the room in the manner described in Section 5.2.2, recording at regular intervals the amplitude of the sound it hears. The list of values thus produced constitutes a signal-intensity map of its operating environment.

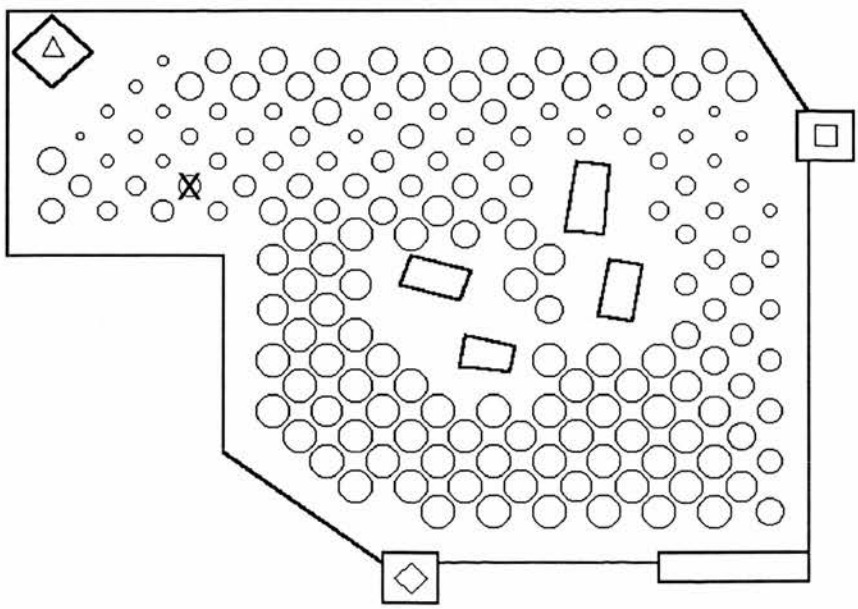

Figure 5.3. Typical intensity map for a loudspeaker at the diamond (bottom-centre)

It is true that the amplitude of the signal must be more or less inversely related to the square of the robot's instantaneous distance from the generator. However, it does not interpret the data in this way, but rather treats the pattern of intensities as a

partitioning of space; that is, it simply constructs a qualitative map of the environment with them. Figure 5.3 depicts a typical intensity pattern recorded by the robot at the 166 sampling locations, with a loudspeaker located at the diamond-marked square at the bottom of the diagram transmitting a tone. The radius of a circle is proportional to the intensity of sound detected at a position near its centre.

Suppose next that the robot is placed near the **X** in Figure 5.3, and has no prior knowledge to influence its guess as to where it now is. In this position its microphone will receive the loudspeaker's signal with some intensity I. For each of the 166 mapping locations A_i, the probability that the robot would have heard the sound with amplitude I at that location can be computed by applying the exponential matching function, namely

$$P(\text{intensity is } I \,|\, \text{position is } A_i) = e^{-[[I - I(A_i)]/\alpha]^2},$$

where $I(A_i)$ is the intensity that was mapped at position A_i previously, and α controls how strict the comparison between past and present is. Given some scheme for digitising intensities, setting $\alpha = 30$ means, for instance, that for any position whose mapped amplitude lies within 5 intensity units of the current signal, the likelihood that such a signal would be heard in that position is greater than 50%. Now, from the assumption that the robot could be anywhere, $P(\text{position is } A_i) = 1/166$, $i = 1,\ldots,166$. For each position, the conditional probability that the robot is at that position given that it is receiving a tone of intensity I can be computed by

$$P(\text{position is } A_j \,|\, \text{intensity is } I) = \frac{P(\text{intensity is } I \,|\, \text{position is } A_j)}{\sum_{i=1}^{166} P(\text{intensity is } I \,|\, \text{position is } A_i)}, \qquad 1 \le j \le 166.$$

When this distribution is computed for the data depicted in Figure 5.3, supposing arbitrarily that I is equal to the actual mapped value at the position near the **X**, and $\alpha = 30$, the result is as in Figure 5.4. In the figure, the size of a circle is proportional to the conditional probability that the robot is at a position near the circle's centre, given a received signal intensity of I. It is clear that with α set so that matches must be very precise (in terms of amplitude units) before they become likely, the robot is able to identify its actual location with a very high probability. Obviously the largest circle is at the exact position of the **X** in Figure 5.3, for I was chosen to be the map value at this position, so that the match probability was equal to 1 there. By choosing the maximum likelihood (MAP) match, the robot would therefore succeed in localising itself accurately.

It is easy to see that tight matching has resulted in a small number of possible

positions, so the robot can be quite certain that it is at one of only a handful of places. However, it is also apparent that some of these positions, while probabilistically plausible, are spatially quite distant from the robot's actual location. This effect presumably results from the fact that signals of the same intensity will be heard at many places in the room (indeed, the likely positions shown in Figure 5.4 lie roughly on a circular arc, which presumably approximates the inverse-squared distance from the loudspeaker proportional to intensity I). A possible next step would be, for example, to compute the dispersion of the distribution of possible locations, for different signal intensities, in an effort to determine how precise localisation based on a system such as this can ever hope to be.

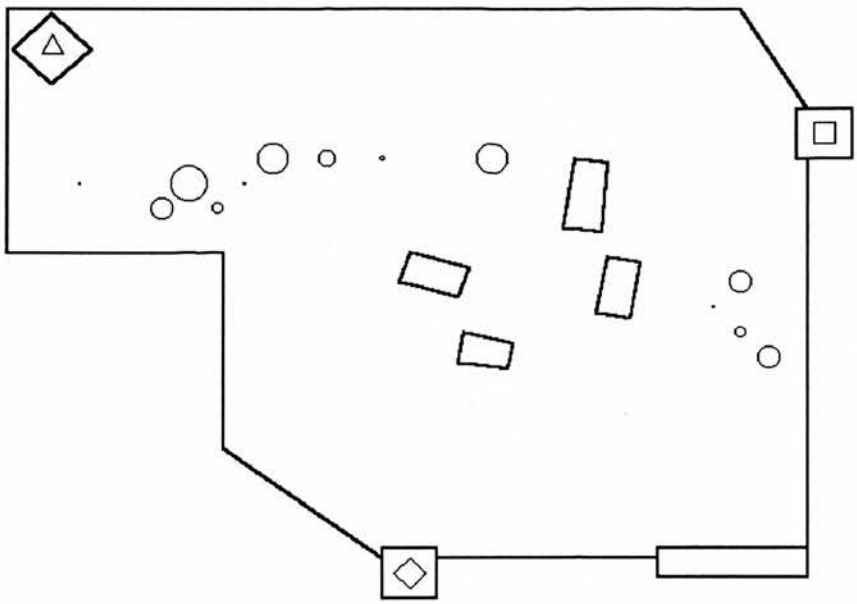

Figure 5.4. Probabilities that robot is at each of the 166 mapping locations

In fact, the advantage of the manageable formalism proposed in this section for mapping and localising by means of a qualitative, probability-like representation, is that it makes available a gamut of analytical tools from probability and information theory; these can facilitate efforts to characterise the limitations on usefulness to localisation of a typical sensing device. Procedures can be established for computing the certainty and accuracy with which a particular device allows position estimates to be made, and the effect of the matching parameter, α, on this process. Furthermore, the common currency of conditional probability distributions potentially allows the performance of disparate systems, and even combinations of such systems, to be compared in a principled way.

5.2.4 Assessing the performance of individual systems

Having constructed a qualitative map of its environment by mapping the data field of a space-segmenting sensor, the robot localises by attempting to match data it is currently sensing with the previously-mapped values. The localisation activity produces what has been called a position probability distribution. From it, the robot might estimate its location by picking the position where the mapped data matches the current data maximally (or one of the positions, if there are several equally good matches), or it might compute a spatial average by weighting the coordinates of positions where it might be by the individual probabilities that it is at each of them.

In order to overcome the lack-of-objective-assessment criticism of synthetic-type approaches, the work described here attempts to go beyond simply presenting the system's successes or failures subjectively. Instead, the proposed qualitative scheme for localisation will be analysed quantitatively to answer two principal questions, namely, "how *certain* can the robot be of its position, for a particular sensing system?", and "how *accurate* is this estimation?". Clearly a good localisation system is one that estimates the robot's position accurately, and is very confident of its estimate.

Localisation certainty

Intuitively, it seems that certainty must somehow be an inverse function of the degree to which space is segmented in a qualititative map. Where, for example, there are only two distinguishable regions, the robot can determine which of these two regions it is situated in, but can not specify its location any more definitely than this. Consider a set of position probabilities computed when the robot is trying to localise, given a previously-constructed qualitative map. The values of the members of this set dictate the amount of "choice" the system has in deciding where it might be. If all but one of these likelihoods is 0, and the remaining one is 1, then it has no choice at all: it must say that it is certain of being at the one position with non-zero likelihood, and cannot be at any other place. If all of the likelihoods are the same, then it has maximum choice: it could be anywhere. Shannon's (1949) objective was to define a quantity, say H, that would measure this "degree of choice", and he showed that a good candidate is

$$H = -K\sum_{i=1}^{n} p_i \log p_i .$$

Here, he assumes that there are n possible "events" having likelihoods $p_1,\ldots,p_n$ and K is a (positive) constant of proportionality. Couched in the terms of robot localisation, this certainty metric has the properties we require: when only one position is likely, and

all the rest are impossible, it has value 0, indicating no choice; when all positions are equally likely, it takes on a maximum value of $\log n$, indicating maximum choice. Furthermore, any shift in the p_i's that makes them more equal increases uncertainty, whereas any shift that tends to make one p_i more likely than the others always decreases uncertainty.

This quantity H is the amount of entropy in the localisation probability distribution. Suppose the robot senses some external condition B, and that it computes the conditional probabilities of being at each of its 166 positions A_i, $P(A_i|B)$, $i = 1,\ldots,166$. The entropy in this distribution (omitting the constant) is defined by (Khinchin, 1957)

$$H = -\sum_{i=1}^{166} P(A_i|B) \log_2 P(A_i|B).$$

When the conditional distribution is flat, that is, given the sensed condition B, the robot has about an even likelihood of being anywhere, or $P(A_i|B) \cong 1/166$, $i = 1,\ldots,166$, then the value of H is $\log_2 166 \cong 7.38$ (the choice of base is arbitrary, but 2 is convenient for reasons to be discussed shortly). This is the maximum value that the entropy of a discrete distribution of 166 values can take. Maximum entropy is equivalent to maximum disorder in the distribution, so that such a distribution tells the robot nothing about position. On the other hand, if $P(A_k|B) = 1$ for some k, with $P(A_i|B) = 0$, $i = 1,\ldots,166$, $i \neq k$, the value of H is 0, its smallest possible value. Minimum entropy, in turn, means that the distribution is minimally disordered, that is, the robot is maximally certain of where it is. For distributions intermediate between maximum and minimum order, H takes on intermediate values between 0 and, in this case, 7.38.

In other words, the entropy function appears to capture the intuitive nature of localisation certainty very well: at the one extreme, where the distribution contains no distinction between potential positions, the robot can not expect to identify its location, and H is as large as it can get; at the other extreme, where the distribution consists of a single, sharp peak, there is a unique point at which the robot can be, and H takes on its minimal value of 0. Entropy is proportional to the degree to which the robot is *uncertain* about where it is. To measure certainty, we need only "invert" entropy by computing

$$N = \log_2 166 - H.$$

The value N is obviously smallest when H is largest, and vice versa, and is therefore proportional to how certain the robot can be about its location. This

difference between entropy and maximum entropy is sometimes called "information"[4]. When it is large, it means that the distribution from which it was derived is very well segmented (i.e., that the robot has a lot of information or certainty about position). It is small when the distribution is flat, and the robot has little information about where it is. By taking logarithms to the base 2, the values N and H are given in "bits". If $N = 3$, for example, the robot has 3 bits of information about position, which means that its qualitative map segments space into about 2 to the power 3, or 8, distinguishable regions. Again, when N is large, the robot can distinguish many different regions of space, and is therefore more certain of its position estimates than when N is small.

Shannon observes other properties of this function that substantiate its use as a certainty measure. For example, suppose that the robot has an equal likelihood of being anywhere; then its uncertainty H is $\log n$, which for $n = 166$ is about 7.38 bits. If the number of positions the robot distinguishes is increased to m, and it is still equally likely to be anywhere, then the absolute amount of uncertainty necessarily increases, since $\log m > \log n$ if $m > n$ (notice that the amount of information, N, defined to be maximum H minus actual H, is of course still zero).

Also, the logarithmic element of the function confers on it very convenient additive features: suppose that (to paraphrase Shannon's own example in terms relevant to this work) from its sensory field map the robot determines that it might be at three possible locations, say A_i, A_j, and A_k with probabilities 1/2, 1/3, and 1/6 respectively; then, the uncertainty in the three possibilities decomposes nicely into a weighted sum of the uncertainty in, say, A_i versus the pair A_j and A_k, and the uncertainty in the A_j and A_k pair itself. That is,

$$H(\tfrac{1}{2},\tfrac{1}{3},\tfrac{1}{6}) = H(\tfrac{1}{2},\tfrac{1}{2}) + \tfrac{1}{2}H(\tfrac{2}{3},\tfrac{1}{3}),$$

where the weighting factor of 1/2 in the second term specifies the amount of the original probability distribution that it accounts for. A choice between several positions is thus equivalent to a succession of choices between subsets of these. Therefore, if there are additional criteria available (perhaps from another sensing system) guiding choice between particular subsets of the possible positions, it is a simple matter to show that these additional constraints always decrease the uncertainty of the unconstrained case, or at worst leave it unchanged.

---

[4]Note that "information" and "entropy" are often used to describe the same quantity H. Indeed, Shannon and Weaver (1949) themselves use the terminology in this way. However, as *more* entropy effectively tells us *less* about a situation, we find it more compelling to consider the words as denoting converse things.

Localisation accuracy

Although a sharp-peaked probability distribution means that the robot has a lot of information about position (is quite certain of where it *thinks* it is), this does not necessarily mean that its guess is correct. In order to estimate the accuracy of a particular sensing system, it is sufficient to compute the distance between the coordinates of the actual position (assumed to be known to the assessor), and the coordinates of the MAP position or the conditional expectation. If the actual robot position is $\mathbf{x}_R$, and the MAP is position A_k, so that its coordinates are $\mathbf{x}_k$, then the two distance errors are given by (assuming column vectors)

$$\text{error}_{\text{MAP}} = \left\|\mathbf{x}_R - \mathbf{x}_k\right\| = \sqrt{(\mathbf{x}_R - \mathbf{x}_k)^{\mathrm{T}}(\mathbf{x}_R - \mathbf{x}_k)}, \quad \text{and}$$

$$\text{error}_{\text{E}[\tilde{\text{A}}|\text{B}]} = \left\|\mathbf{x}_R - \text{E}[\tilde{\text{A}}|\text{B}]\right\| = \sqrt{(\mathbf{x}_R - \text{E}[\tilde{\text{A}}|\text{B}])^{\mathrm{T}}(\mathbf{x}_R - \text{E}[\tilde{\text{A}}|\text{B}])}.$$

Smaller distance errors obviously imply more accurate localisation.

Systematic error

As discussed in Section 5.2.1, the static mapping procedure employed here does not represent the ultimate in qualitative localisation, and in practice it would be preferable for the robot to map its environment dynamically (in terms of transitions in sensor readings, for instance). For the purposes of evaluating the localisation systems quantitatively, however, maps were built by static sampling in a Cartesian frame of reference. Thus, it must be borne in mind that the absolute magnitude of the above distance measures is subject to errors in $\mathbf{x}_R$ (the robot's position as measured by the experimenter), and $\mathbf{x}_k$ (coordinates of grid position A_k).

During mapping and localisation attempts, the experimenter confirmed that the robot had stopped within a few millimetres of a grid position, and that its orientation was within a few degrees of the direction in which it was meant to be facing. The accuracy results in Chapters 6 and 7 have been estimated in centimetres or tens of centimetres, so that errors in the $\mathbf{x}$'s of a few millimetres will not contribute significantly to any putative misinterpretation of these results. In any case, the localisation at off-grid positions discussion in Section 8.1.2 shows that the robot can localise quite well to the nearest grid position (i.e., estimate which grid position is nearest to it) even if it is stationed several centimetres from it.

Variance measures

It remains to be seen which of MAP and conditional expectation results in the best

estimate of position for a particular sensor system. However, it is clear that for the purposes of the analysis, the conditional expectation is a much more telling quantity, as it is contributed to by *all* of the positions at which the robot thinks it might possibly be, not just the single most likely one. Given that the MAP position's coordinates can be thought of as the "spatial mode", and that conditional expectation specifies the "spatial mean", a further and final characterisation of a position probability distribution is the spatial analogue of variability around the mean. The conditional variance that is applicable here is defined as

$$V[\tilde{A}|B] = E[\tilde{A}^2|B] - E[\tilde{A}|B]^2,$$

which for the present purposes it evaluates to

$$V[\tilde{A}|B] = \sum_{i=1}^{166} \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}} P(A_i|B) \; - \; E[\tilde{A}|B] \, E[\tilde{A}|B]^{\mathrm{T}}.$$

These formulations are just the vector versions of variance as it is usually thought of in unidimensional terms, in that they estimate the mean squared-deviation of position estimates (weighted by the likelihood of being at the position) from their mean value (the conditional expectation).

Whereas the conditional expectation is a two-dimensional vector (the coordinates of the weighted mean of the conditional probability distribution), the conditional variance is actually a two by two covariance matrix. It takes the form

$$\begin{bmatrix} \mathrm{var}(x_0) & \mathrm{cov}(x_0, x_1) \\ \mathrm{cov}(x_0, x_1) & \mathrm{var}(x_1) \end{bmatrix},$$

where var(x) denotes the variance of a single variable x, and cov(x,y) denotes the covariance between two variables x and y.

For normally distributed, one-dimensional data, about 68.3% of values can be expected to lie in an interval within one standard deviation of the mean, 95.4% within two standard deviations, 99.73% within three, etc. Even where a distribution is not normal, the above numbers are conventionally used as levels of confidence. In the spatial case, the covariance matrix defines a confidence ellipse, centred on the expectation.

Example revisited

Figure 5.5 shows three confidence regions for the example in Section 5.2.3. It is plain that the conditional expectation (the thick circle on which the ellipses are centred) is

quite a long way away from the robot's actual position (near the left end of the smallest ellipse). The size and orientation of these ellipses capture the skewed shape of the probability distribution, which obviously has much more variability in the left-right direction than up-down. The smallest ellipse marks the region likely to contain 68.3% of the probability density, the next 95.4% of the density, and the largest 99.73%; these are the spatial equivalents of one, two, and three standard deviations from the mean for unidimensional data.

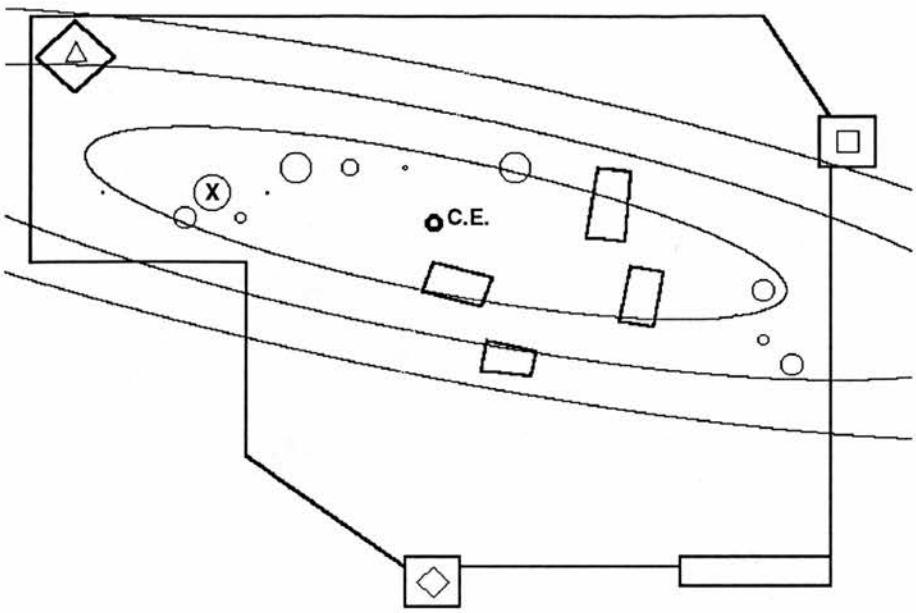

Figure 5.5. The 68.3%, 95.4%, and 99.73% confidence regions for the example discussed in Section 5.2.3; **X** marks the robots actual position, and **C.E.** the conditional expectation

Deriving the shape and orientation of these ellipses from the conditional variance takes a few steps. First of all, by rotating the data through an appropriate angle, it is possible to eliminate the covariance of the x_0 and x_1 directions. Then, the shape of the confidence ellipse is easy to visualise (and plot): the variance of x_0 controls the length of the x_0 axis of the ellipse, and likewise for x_1. The equation of the ellipse (implicitly assumed to have its centre at the spatial mean) is

$$\frac{x_0^{\,2}}{\mathrm{var}_\theta(x_0)} + \frac{x_1^{\,2}}{\mathrm{var}_\theta(x_1)} = R^2,$$

where $\mathrm{var}_\theta(x)$ is the variance of x after the data have been rotated through an angle θ that eliminates any covariance between x_0 and x_1. It can be shown that these new variances are given by

$$\mathrm{var}_\theta(x_0) = \mathrm{var}(x_0)\cos^2\theta + 2\mathrm{cov}(x_0,x_1)\sin\theta\cos\theta + \mathrm{var}(x_1)\sin^2\theta, \text{ and}$$
$$\mathrm{var}_\theta(x_1) = \mathrm{var}(x_0)\sin^2\theta - 2\mathrm{cov}(x_0,x_1)\sin\theta\cos\theta + \mathrm{var}(x_1)\cos^2\theta,$$

where the angle of rotation is

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2\mathrm{cov}(x_0,x_1)}{\mathrm{var}(x_0)+\mathrm{var}(x_1)}\right).$$

(Note that these variances are directly related to the eigensolution of $V[\tilde{A}|B]$, such that $\mathrm{var}_\theta(x_0)$ is the maximum eigenvalue of the matrix, and $\mathrm{var}_\theta(x_1)$ the minimum eigenvalue.)

The radius R of a confidence ellipse is determined from the χ^2 probability distribution for two degrees of freedom. From this distribution it is established that for $R^2 = 2.3$, the ellipse is expected to contain 68.3% of data points, for $R^2 = 6.17$ it contains 95.4% of the data, and $R^2 = 11.8$ is the 99.73% confidence region. These values were used as the radii of the ellipses plotted in Figure 5.5.

Summary

Given a position probability distribution generated from a qualitative sensory map, the various measures introduced above provide the following basic analytic facilities:

- the entropy of the distribution is inversely proportional to how certain the robot can be of a localisation attempt, and the information in the distribution is directly proportional to this certainty;
- the difference between the conditional expectation or maximum likelihood estimate of the distribution and the robot's actual position establishes how accurate a localisation attempt is; and
- confidence ellipses computed from the conditional variance of the distribution can be used to visualise the variability of possible robot locations around the conditional variance estimate of its position.

Evaluation procedure

The individual sensory modalities under consideration (namely sonic, infrared, and ultrasonic), will be evaluated as follows:

1. First, the robot constructs a qualitative map of the sensor's data field at the 166 sampling locations, as discussed in Section 5.2.2.

2. Next, the robot revisits all 166 sampling locations. From the previously stored map and the data it is receiving at the current test location, it computes the distribution

of (subjective) probabilities that it is at each of the 166 sampling positions.

3. Thus, at the end of two trips around its environment (one for mapping, and one for attempting to localise), the robot has a set of 166 position probability distributions, one per localisation attempt at each grid position.

4. The robot is then able to compute, per location, how certain it was of where it estimated itself to be. Also, given the actual coordinates of each grid position (by the experimenter), it is able to compute how accurate its localisation attempt was at each location.

5. For the sensor system under consideration, therefore, the robot has computed how certain and accurate at localising the system allows it to be throughout its operating space. For evaluation purposes, it is possible to determine the worst-, best-, or average-case effectiveness of each sensor system.

6. Furthermore, the above tests can be repeated with different values for the match strictness parameter, α, in order to determine its effect on performance.

A detailed example and discussion of the application of this procedure forms the first section of Chapter 6. The certainty and accuracy measures described in this section will also be employed in evaluating combinations of systems, the additional basis for which will be described next.

5.2.5 Assessing the performance of combinations of systems

The problems of combining information furnished by more than one sensory system have been approached in several ways, usually under the rubric of "sensor fusion". Perhaps the best developed work in this field is that of Durrant-Whyte (1988b). In essence, the central tenet of sensor fusion is that effective models of the capabilities of individual sensors allow the *a priori* construction of optimal strategies for combining and using them. A sensor model is an abstraction of the physical process of sensing that describes the kind of information the sensor provides (the "observation model"), information the sensor must receive from other sources in order for it to operate (the "dependency model"), and how the sensor's operation is affected by things like where it is and what it is doing (the "state model"). The information a sensor is able to extract is defined geometrically, but it is described in probabilistic terms similar to those adopted here, as this allows disparate sensors to communicate by means of probability distributions as a common currency.

The localisation scheme suggested here differs in that it takes something of a

"reversed" approach to the problem: rather than proceeding from sensor abstractions to designing strategies for optimal multiple sensing, this work starts with arbitrary qualitative sensing strategies for several devices, in an effort to expose, from the results of the analysis, (perhaps optimal) abstractions of the sensing process. Furthermore, in moving from data to model, this approach circumvents both the difficult task of attempting to model the complex physical devices that sensory transducers doubtless are, as well as the risk of fundamentally invalidating any putative application of these abstracted sensors by modelling them incorrectly. Nevertheless, some of the general procedures for combining probabilistic descriptions of sensory information are common to both approaches.

Once the question of how to combine sensory data has been resolved, the measures of certainty and accuracy introduced in the foregoing section may be applied. These basic measures will be supplemented by methods for expressing the degree of similarity between the results of separate localisation attempts.

Estimating and combining, or combining and estimating?

If the robot has two sensory systems, it constructs two qualitative sensory maps as it travels around its environment initially, recording at each sampling position the data received at each sensor. When on a subsequent journey it attempts to localise, it will now have two pieces of current data, which must be matched against the stored data in their respective maps. In fact, this is the situation even for a single sensory system in the case of the sonic and infrared beacon detectors: the robot stores the sensory fields of each individual beacon as a separate qualitative map.

The localisation exercise will generate two position probability distributions (either from two different sensory systems, or from one of the beacon-based systems but for two different beacons). Then, the crucial question is, "should a location estimate be established for each distribution, and then these estimates combined, or should the distributions themselves be combined, and then a position estimate established from their composition?"

It is easy to see that these two approaches are not equivalent. Consider the distribution of points depicted in Figure 5.6. Suppose that the position estimates generated by each of the sensors have equal likelihood. Then, the expectation of sensor 1's estimates is probably somewhere near the middle of the four by four grid of positions on the left, while the expectation of sensor 2's estimates presumably lies near the middle of the four by four grid on the right. Therefore, averaging the expectations

for the two sensors (assuming neither is more believable than the other) gives a combined expectation probably somewhere near the **X**. On the other hand, if the two sets of position estimates themselves are combined, then the resulting distribution comprises mainly those points at the bottom centre of the diagram that are common to the two systems. The expectation of this combined distribution would lie somewhere near the **H**, which quite obviously differs from **X**.

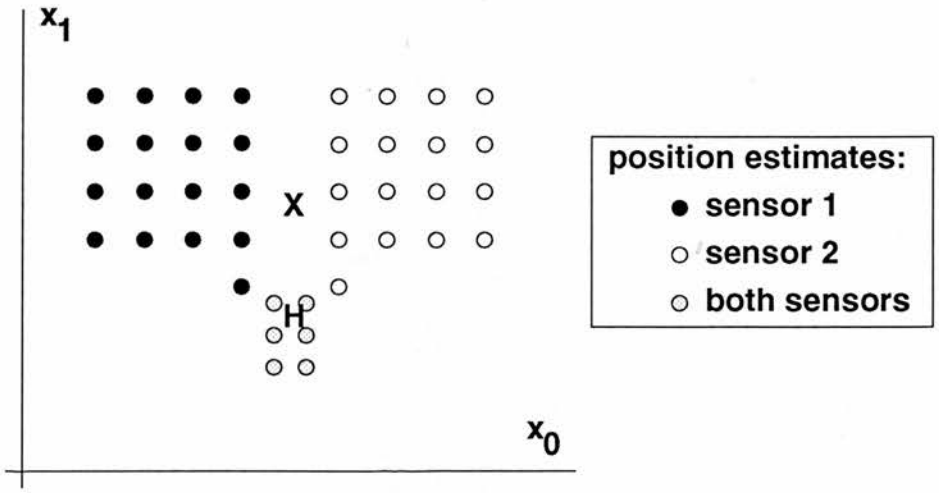

Figure 5.6. Distribution of points where the "combined expectation" of the two sets (near **X**) differs from the "expectation of the combined set" (near **H**)

Implicit in the example above was the assumption that if a point was not included in the set of likely locations specified by one of the sensors, then the sensor believed that it definitely was *not* at that point. That is, if sensor 1 believed that it could only be at one of the 23 positions in the diagram, and nowhere else, and if sensor 2 was equally certain of its estimates, then the most appropriate combination scheme must be the one which finds the intersection between the distributions, and then computes the expectation of this new distribution (**H**). If, on the other hand, sensor 1 believed it was at one of these 23 positions, but had *no opinion* on whether it was anywhere else, then the situation is somewhat different, and choosing to combine the individual expectations (**X**) might be an appropriate action to take.

Therefore, in comparing the position probability distributions of different systems, the following assumption is made: a position has a conditional probability of zero if and only if the system is certain that it is *not* at that position. In this case, it is appropriate to combine distributions, and then compute the expectation of the composite distribution.

Combining conditional probability distributions

Having decided to combine and then estimate, it is necessary to decide whether just "combining" distributions is sufficient, or if it will be necessary to combine the qualitative map information itself, and then compute a probability distribution for the new "composite" map. That is, suppose sensor 1 builds a qualitative map $B_1(A_i)$, $i = 1,\ldots,166$, that consists of stored sense data for each position A_i, $i = 1,\ldots,166$, and that sensor 2 has a corresponding map $B_2(A_i)$, $i = 1,\ldots,166$. From these two maps can be constructed a new map $B_{1\circ 2}(A_i) = [B_1(A_i), B_2(A_i)]$, $i = 1,\ldots,166$. The question is, "how do the distributions calculated for the two separate maps compare to the distribution that would obtain from the composite map?"

So, given some sensed condition B_1, the robot computes conditional probability distribution $P_1(A_i|B_1)$, $i = 1,\ldots,166$. Similarly, for sensor 2 and its sensed condition B_2, the distribution $P_2(A_i|B_2)$, $i = 1,\ldots,166$, is derived. Then, the *combined* distribution is

$$P_{1\cap 2}(A_j|B_1,B_2) = \frac{P_1(A_j|B_1)P_2(A_j|B_2)}{\sum_{i=1}^{166} P_1(A_i|B_1)P_2(A_i|B_2)}, \quad j = 1,\ldots,166.$$

Recalling from previous sections that

$$P(A_j|B) = \frac{P(B|A_j)}{\sum_{i=1}^{166} P(B|A_i)}, \qquad j = 1,\ldots,166,$$

and having defined $P(B|A_i)$ as a function that matches currently sensed conditions against mapped values

$$P_1(B_1|A_i) = e^{-\{[B_1 - B_1(A_i)]/\alpha\}^2} = f_1(A_i, B_1), \quad \text{and}$$
$$P_2(B_2|A_i) = e^{-\{[B_2 - B_2(A_i)]/\alpha\}^2} = f_2(A_i, B_2).$$

Then the combined distribution can be expressed as

$$P_{1\cap 2}(A_m|B_1,B_2) = \frac{\dfrac{f_1(A_m,B_1)}{\sum_i f_1(A_i,B_1)} \dfrac{f_2(A_m,B_2)}{\sum_i f_2(A_i,B_2)}}{\sum_k \left\{ \dfrac{f_1(A_k,B_1)}{\sum_j f_1(A_j,B_1)} \dfrac{f_2(A_k,B_2)}{\sum_j f_2(A_j,B_2)} \right\}}, \quad m = 1,\ldots,166,$$

which simplifies to

$$P_{1\cap 2}(A_m \mid B_1, B_2) = \frac{f_1(A_m, B_1) f_2(A_m, B_2)}{\sum_k \{f_1(A_k, B_1) f_2(A_k, B_2)\}}, \quad m = 1,\ldots,166.$$

The question is how this combined distribution is related to the distribution that would result from applying a *composite* matching function $f_{1\circ 2}(A_i, B_1, B_2)$, which composes the matching functions f_1 and f_2, and acts on the composite sensory map $B_{1\circ 2}(A_i)$, $i = 1,\ldots,166$. The composite distribution is therefore

$$\begin{aligned} P_{1\circ 2}(A_m \mid B_1, B_2) &= \frac{f_{1\circ 2}(A_m, B_1, B_2)}{\sum_k f_{1\circ 2}(A_k, B_1, B_2)} \\ &= \frac{f_1(A_m, B_1) \circ f_2(A_m, B_2)}{\sum_k f_1(A_k, B_1) \circ f_2(A_k, B_2)}, \quad m = 1,\ldots,166. \end{aligned}$$

In other words, provided the composition operator is chosen to be multiplicative, the combined distribution $P_{1\cap 2}$ is equal to the composed distribution $P_{1\circ 2}$. From this equivalence, it follows that once the localisation probability distributions have been computed for two sensory systems, it is appropriate to combine them by multiplying and renormalising the pairs of probabilities for each position. The entropy, MAP or conditional expectation, and conditional variance of this new distribution will then characterise the localisation certainty, accuracy, and accuracy variability, respectively, that obtain when the robot has both systems available to it.

Conflict measures

Given two sensor systems (or a beacon-based system with two active beacons), it would be interesting to know how well the two agree on where they believe the robot is located. The problem is basically to compare the position probability distributions generated by the two systems. There are various ways in which this comparison could be made. Point by point comparison between the two distributions could be performed by means of a standard test such as the linear correlation coefficient, defined as

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}$$

which gives the correlation between two vectors **x** and **y**, and which becomes

$$r = \frac{\sum_{i=1}^{166} P_1(A_i, B_1) P_2(A_i, B_2) \; - \; \frac{1}{166}}{\left[\left(\sum_{i=1}^{166} P_1(A_i, B_1) P_1(A_i, B_1) - \frac{1}{166} \right) \left(\sum_{i=1}^{166} P_2(A_i, B_2) P_2(A_i, B_2) - \frac{1}{166} \right) \right]^{1/2}},$$

for the present purpose of comparing 166-point discrete probability distributions. However, attention must be paid to the typical shape of the distributions: since the match function is exponential, it is possible to have a situation like that depicted in Figure 5.7. Here, two exponential curves are close together, but have hardly any overlap. The r measure would show only a very small correlation between them. Since this situation often obtains in practice, pointwise comparison is not particularly useful.

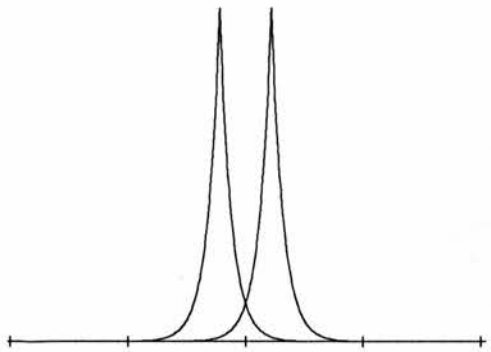

Figure 5.7. Two exponential curves which are close together, but which hardly overlap

Other possible comparisons between position probability distributions are to find either the magnitude of the largest point of agreement, or to find the total area of agreement (actually, since the distributions are two-dimensional, the total volume of agreement is computed). The former is given by

$$\max[\, \min\{P_1(A_i, B_1), P_2(A_i, B_2)\}\,], \quad i = 1, \ldots, 166,$$

and the latter is given by

$$\sum_{i=1}^{166} \min\{P_1(A_i, B_1), P_2(A_i, B_2)\}.$$

Figure 5.8 interprets these two quantities graphically for the curves shown in Figure 5.7. The situation is analogous for two-dimensional position probability distributions. On the left is the first of the two measures, which has been labelled "minimax". The overlap metric is shown on the right.

In fact, there is a further consideration which constrains the choice of conflict measures. In the evaluation procedure discussion in the previous section, it was noted that one aspect of the assessment was to determine the effects of match strictness on localisation success. Therefore, the robot's localisation systems are tested over a wide range of the match control parameter, α. As noted earlier (Section 5.2.3), when α is

very large any current data should match any mapped data. As the matching operations for the two systems become increasingly lax, the two position probability distributions produced from them effectively tend towards "flatness": the systems have no basis for distinguishing one part of space from another (any sensed data matches the stored data for all 166 positions equally well).

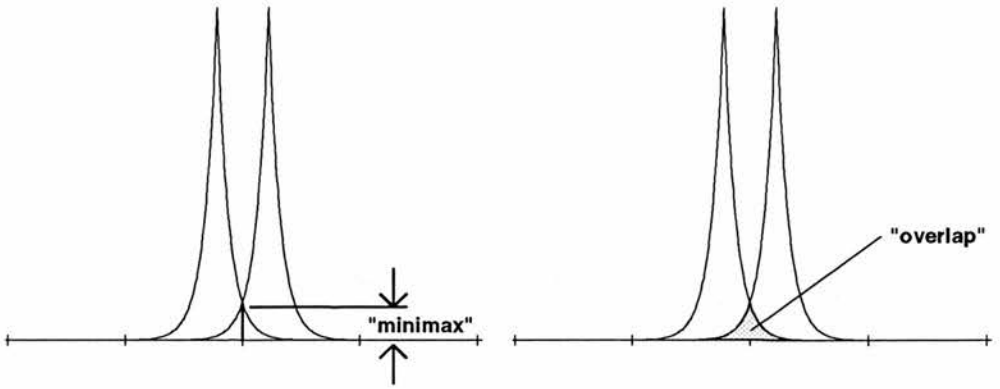

Figure 5.8. Diagrammatic representation of conflict measures in one dimension; at the left, the solid vertical line indicates the magnitude (and position) of the maximum overlap between the curves; at right, the shaded area indicates the total area of overlap

Under these circumstances, the conflict measure used should indicate less conflict. However, the absolute nature of the "minimax" metric is such that it shows the opposite effect: as the distributions become flatter with increasing match freedom, the magnitude of the point of greatest agreement decreases in proportion. Therefore, only the overlap results will be presented in Chapters 6 and 7, as these remain a revealing measure of agreement for both strict and loose matching.

Summary

Two principal points have emerged in this discussion of analysing localisation performance when more than one sensory system is available:

- position probability distributions should be combined first, and the joint location estimation should be computed from the combined distribution; and
- it is not necessary to form a "composite" map from separate ones and then to construct the composite conditional probability distribution; it is sufficient to multiply and renormalise the distributions themselves as long as the composition operation is defined to be multiplicative.

Furthermore, once distributions have been combined, the entropy, MAP position, conditional expectation, and conditional variance measures discussed earlier can be used to characterise the combined distribution just as they were used for individual

distributions. The degree of conflict between two systems (or two beacons of one beacon-based system) can be usefully measured by finding the total "probability volume" lying under both distributions at once.

5.3 Summary

Qualitative, topological map-based localisation as conceptualised here should allow a robot to determine its location virtually anywhere within its operating domain, from minimally interpreted sensor data, rather than just along object-geometry-based fixed routes. In order to explore the development of such a scheme, the robot detailed in Chapter 3 was driven around a pen equipped with sonic and infrared beacons; it was programmed to construct sensory data field maps from readings of its microphones, infrared beacon detector, and ultrasonic module at 166 positions covering its operating domain. A simple, probability-like interpretation of location estimation was established.

Provided its underlying assumptions are respected, the proposed formalism affords a number of useful analytical tools from probability and information theory, which facilitate efforts to characterise the performance of a typical sensing device. Procedures were established for computing the certainty and accuracy with which a particular device allows position estimates to be made, and the effect of the strictness of the matching operation on this process. Moreover, the common currency of conditional probability distributions potentially allows the performance of disparate systems or combinations of them to be compared. Table 5.3 (next page) collects the various methods discussed in the foregoing sections, describing what is to be established with their aid, and the assumptions on which their usage is based.

By making two passes through the environment, mapping it during the first pass and subsequently trying to localise within it, the robot can measure the certainty and accuracy of localisation per sensor system everywhere in its operating space. This enables the assessment of worst-, best- and average-case performance of systems. Chapter 6 presents the results of this analysis for individual sensor systems, and Chapter 7 repeats it for combinations of systems.

Method	Intended use	Assumptions, etc.
Entropy or "information" in distribution	tests how certain a sensor system is of its estimates	requires only a qualitative map, i.e., sensed data from past visits can be abstractly indexed
Difference between actual position and MAP or conditional expectation of distribution	determines how close the system's estimates are to where the robot actually is (spatial mode and mean)	requires that sensed data be associated with their coordinates in space
Conditional variance of distribution	determines how variable the system's guesses are about their spatial mean	as above
Renormalised product of individual distributions	combines localisation estimates from separate systems into a single distribution	composition of matching operations of separate systems must be defined to be multiplicative
Probability volume lying under both distributions	basis for testing how well sets of estimates agree	

Table 5.3. Analytical methods, what they are intended to establish, and assumptions or comments about their use

Chapter 6

Localisation Analysis for Individual Systems

The three locative sensing devices to be considered here are the sonic signal amplitude subsystem, the infrared beacon detector, and the ultrasonic module. By programming a mobile robot to collect an extensive amount of sensor data at a large number of positions within an experimental pen, detailed data-field maps were constructed for each of these sensory modalities. These sensory fields offer a unique insight into how a non-trivial operating environment appears to the robot's sonic, infrared, and ultrasonic sensors as it moves around within it. Furthermore, by computing various statistical quantities of the data fields, it is possible to characterise what each sensor can tell the robot about where it is; these characterisations will hopefully suggest feasible sensors to use for localisation, and effective strategies for using them.

Each of the systems will be considered in turn, beginning with sonic, proceeding to infrared, and ending with ultrasonic. In each case, the system's performance will be evaluated by calculating its positioning certainty and accuracy as proposed in the foregoing chapter. For the sonic and infrared systems, there is an additional dimension to consider, in that both of these depend on external reference beacons, of which three each are provided in the experimental pen; it will therefore be necessary to consider localisation performance per beacon and set of beacons. As the performance of each system is readily expressible graphically, this method of presentation will be preferred.

The chapter concludes with summaries and comparisons of the three individual systems, setting the stage for the analysis of combinations of them. It should be noted that in order to explain the data presentation conventions that will be followed throughout the remainder of this chapter, the sonic system analysis will be described in considerable detail; the analysis of subsequent systems and combinations of systems will be treated in decreasing detail, for the sake of brevity.

6.1 Sonic signal amplitude subsystem

As discussed in Section 5.2.2, a qualitative acoustic map of the experimental pen

consists of a list of signal intensities measured at each of 166 positions distributed evenly across the floor surface. One such map is constructed for each of the three beacons. In this section, localisation certainty and accuracy will be considered for beacon 1 alone, then for beacons 1 and 2, and finally for all three beacons. A brief comparison between the results for individual beacons 2 and 3, and for beacon pairs 1 and 3, and 2 and 3, will confirm that the analyses presented in detail are indeed representative of the other possible cases.

6.1.1 Localisation by beacon 1 alone

Suppose that data set 1 (collected on the robot's first journey around the 166 named positions in its environment) specifies the "current" data, and set 2 specifies the "map" data, where "DataSet*n*(*pos*,*beac*)" will denote the signal intensity of beacon *beac* that was received at position *pos* on mapping visit *n*. The localisation analysis then proceeds as follows:

for each of the 166 positions A_j:

1. let the intensity of the signal that would be heard in that position be B, such that $B = \text{DataSet1}(A_j, 1)$; and
2. as discussed in Section 5.2.3, compute

$$P(A_j|B) = \frac{P(B|A_j)}{\sum_{i=1}^{166} P(B|A_i)}, \qquad j = 1,\ldots,166,$$

where

$$P(B|A_i) = e^{-\{[B-B(A_i)]/\alpha\}^2},$$

with $B(A_i) = \text{DataSet2}(A_i, 1)$, and α a parameter that controls match strictness.

In other words, assume that the robot visits each marked position in the pen, and receives the same signal there as it did when data set 1 was collected (step 1 above), but that it does not itself know where it is; this "current" signal is therefore matched against the "stored" values for each of the 166 positions collected as data set 2 (step 2); given that it is receiving a signal of intensity B at the moment, its likelihood of being at each of the sampling positions is computed. It thus has 166 position estimate distributions (taken represent the probabilities that the robot is at each possible position), which may be expressed as

$$P\{A_j|\text{DataSet1}(A_i, 1)\}, \quad j,i = 1,\ldots,166.$$

As discussed in Section 5.2.4, this set of distributions can be used to characterise the average-, best- and worst-case performance of the robot. That is, the certainty and accuracy of localising with the sonic system and beacon 1 can be computed for each position on the sampling grid. The best- and worst-case estimates are found simply by comparing the measures for each position in the pen. The average-case estimate is found by averaging the measures obtained across the room.

The 166 distributions representing the robot's localisation efforts as it revisits the 166 positions it mapped at might have any of a number of different forms, of which these are the extremes:

1. if $P\{A_j|\text{DataSet1}(A_i,1)\} = 1$ for all $i = j$, and $P\{A_j|\text{DataSet1}(A_i)\} = 0$ for all $i \neq j$, then clearly the matching is so tight that, if the system senses what it did when data set 1 was collected, and uses data set 2 as a map, it can ascertain its current position correctly at any of the marked positions in its operating space; and

2. if $P\{A_j|\text{DataSet1}(A_i,1)\} = 1/166$ for all i and j, then given the "current" and "map" data as assumed above, the robot can never distinguish between any two places.

Obviously case 1 is perfect localisation (in each of the 166 distributions, entropy is 0, information is maximum at $\log_2 166$, or about 7.38 bits), and case 2 is complete inability to localise (entropy is maximum, information = 0). We would expect, therefore, that distributions computed from the collected data lie somewhere between these extremes.

Note that the stored and current intensity values being compared span the range 0 to 255 (they are 8-bit numbers). If $\alpha = 2.55$, for example, then the probability of matching mapped and current values differing by 0 units is 1, by 1 unit is about 0.86, by 2 units is 0.54, by 3 units is 0.25, and by 4 units is only 0.085. Thus, matches must be quite precise if they are to be at all probable. If $\alpha = 2550$, then even if values differ by the maximum of 255 units, they have a match probability of greater than 99%. It therefore seems sensible to consider the effects on localisation of match freedoms between values such as these. For neatness, the match parameter α will be specified as ranging from 1 to 1000, rather than from 2.55 to 2550; i.e., the α in the exponential match functions is actually multiplied by a constant value of 2.55.

The certainty of the sound system for a given value of the match parameter, α, is evaluated by computing each of the 166 distributions as described algorithmically above, and averaging their entropies. Figure 6.1 shows the results of this exercise for the signal of beacon 1. Each data point is plotted conventionally with one-standard-

error bars computed from the standard deviations of the 166 averaged entropies, although these bars are in most cases almost invisible. The standard error of n values is just their standard deviation divided by $\sqrt{n}$; standard error estimates the standard deviation of a sample mean as an estimator of the true mean, so that when two means are plotted with standard error bars, differences between them are easily recognisable as significant if they are separated by a few standard errors.

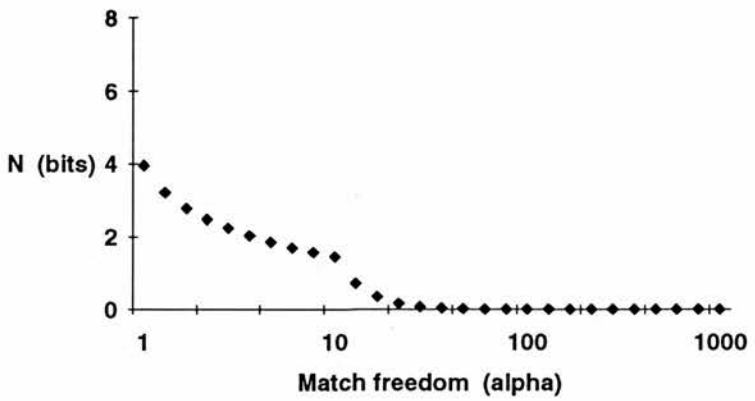

Figure 6.1. Information in the acoustic field of beacon 1

It is easy to see that, for very strict matches, the sonic field of beacon 1 has about 4 bits of information in it. This means that a robot relying on just this beacon, and assuming that the relationship between the signals it detects as it moves about its space is as consistent as the relationship between data sets 1 and 2 appears to be, can distinguish about 2 to the power 4 or 16 different places in its environment. Furthermore, as match freedom is increased, the amount of information drops quickly to 0; in other words, with α greater than about 10, the robot loses its ability to distinguish between positions. The significance of this α will be discussed later.

Localisation accuracy can also be interpreted for the two distribution extremes discussed above. In case 1 (single, correct position has likelihood 1, all other positions have likelihood 0), both the conditional expectation (spatial mean) and the MAP (spatial mode) are equal to the robot's actual position, so that its estimation error is 0. In case 2 (all positions equally likely), the conditional expectation is the centroid of the 166 positions: if the actual position is close to the centroid, therefore, the estimation error will be small; at worst, it is the distance between the centroid and the furthest position from it. For such a flat distribution, there is no unique MAP, so here too, the centroid is as valid an estimator of position as any other.

Again, the actual distributions should lie between these extremes. The evaluation

thus proceeds by computing the conditional expectation and MAP position for each of the 166 distributions; for a given distribution $P\{A_j|\text{DataSet1}(A_i,1)\}$, $j = 1,\ldots 166$, it is known that the robot's *actual* position is A_i, whose coordinates are $\mathbf{x}_i$. Figure 6.2 shows the differences between conditional expectation and $\mathbf{x}_i$, and MAP and $\mathbf{x}_i$, averaged across the 166 distributions, and plotted against the match strictness parameter. One-standard-errors are indicated.

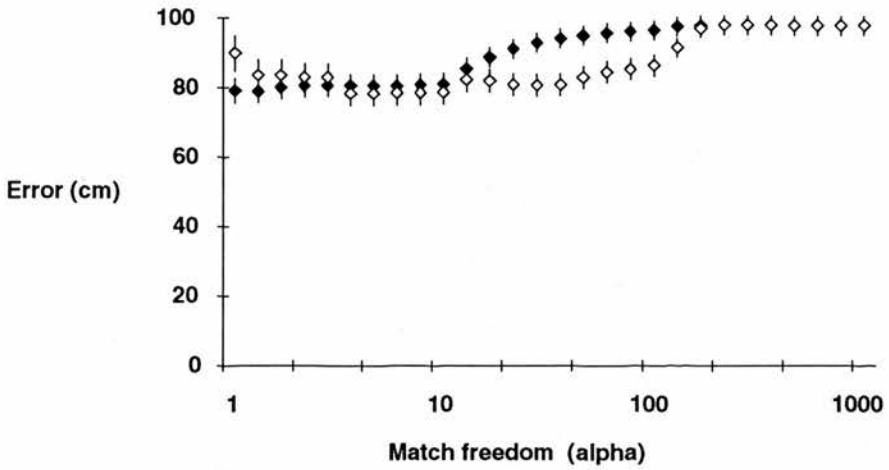

Figure 6.2. In-principle accuracy of localisation by beacon 1; ♦ marks conditional expectation as an estimator, and ◊ denotes the MAP estimator

Where α is stepped through a three-decade range, there is no significant difference between the MAP and conditional expectation as estimators of the robot's position in the first and third decades. In the second decade, the conditional expectation is slightly worse than the MAP estimate. In any case, the localisation error is at least 73 cm for any value of α. Since the distance between the 166 grid points is 23 cm, the sonic signal amplitude subsystem, operating only with loudspeaker 1, can at best permit the robot to estimate its position to within about 1.5 grid positions of where it actually is. It could therefore actually lie anywhere within an area of about $\pi(0.96)^2$ or 2.9 square metres; as its operating space only spans 8.7m^2, this positioning error represents a significant percentage (33%) of the floor surface.

The standard errors plotted in Figures 6.1 and 6.2 indicate the "extrinsic" variance of the set of 166 information or error estimates that were averaged to produce the summary statistics plotted in the figures. The relatively small size of the error bars in Figure 6.2 shows that the accuracy of the conditional expectation as an estimator of position is very *consistent* across the set of positions. Therefore, a succinct method for characterising the "intrinsic" variability (conditional variance) of position estimates about their spatial mean suggests itself, namely to determine, for each distribution, the

confidence region into which the robot's actual position falls.

An example

Suppose the robot is at the position marked **X** in Figure 6.3. At this position, $\mathbf{x} = [15\ 15]^T$. The conditional expectation is determined to be $[76.5\ 20]^T$, and the covariance matrix turns out to be[1]

$$\begin{bmatrix} 2966 & 214.6 \\ 214.6 & 70.87 \end{bmatrix}.$$

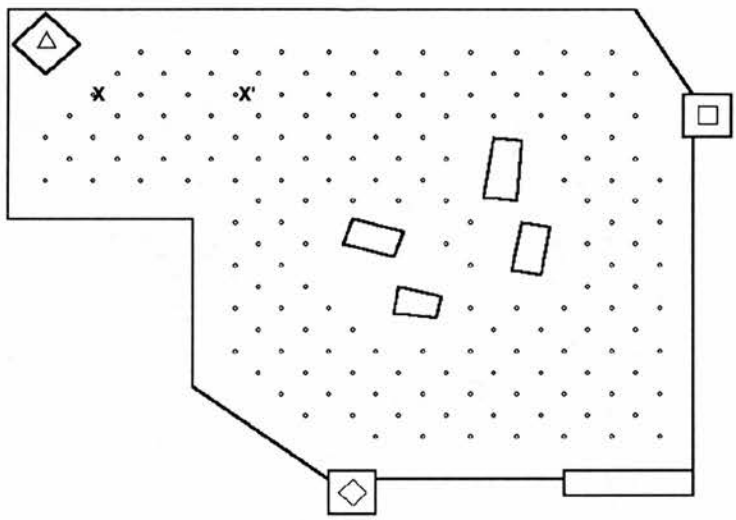

Figure 6.3. Locations of positions **X** and **X'**

Now, the equation of a confidence ellipse determined by this matrix is computed as in Section 5.2.4 as

$$\begin{aligned} \mathrm{var}_\theta(x_0) &= \mathrm{var}(x_0)\cos^2\theta + 2\mathrm{cov}(x_0,x_1)\sin\theta\cos\theta + \mathrm{var}(x_1)\sin^2\theta \\ &\cong 2966\cos^2\theta + 2\times 214.6\sin\theta\cos\theta + 70.87\sin^2\theta \\ &\cong 2982, \quad \text{and} \\ \mathrm{var}_\theta(x_1) &= \mathrm{var}(x_0)\sin^2\theta - 2\mathrm{cov}(x_0,x_1)\sin\theta\cos\theta + \mathrm{var}(x_1)\cos^2\theta \\ &\cong 2966\sin^2\theta - 2\times 214.6\sin\theta\cos\theta + 70.87\cos^2\theta \\ &\cong 55.08, \end{aligned}$$

where θ is computed as

---

[1]The values in this example are given in "floor-grid units", which are the width of the wooden strips out of which the floor surface is constructed. One such unit is approximately 2.29 cm. The choice of distance unit has no effect on the confidence region radii that are computed.

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2\times 214.6}{2966+70.87}\right) = 4.02°,$$

which is the angle the x_0-variance axis of the confidence ellipse makes with the x_0 axis itself. Since $\mathrm{var}(x_0)$ is so much larger than $\mathrm{cov}(x_0, x_1)$, the major axis of the ellipse deviates only slightly from the x_0-axis (about 4 degrees). If the conditional expectation is abbreviated $[\mu_0\ \mu_1]^T$, a confidence region is specified by the equation

$$\frac{(x_0-\mu_0)^2}{\mathrm{var}_\theta(x_0)} + \frac{(x_1-\mu_1)^2}{\mathrm{var}_\theta(x_1)} = R^2, \quad \text{so that}$$

$$\frac{(x_0-76.5)^2}{2982} + \frac{(x_1-20)^2}{55.08} = R^2.$$

Thus, by substituting the coordinates of the actual position ($\mathbf{x} = [15\ 15]^T$) into this equation, we find that this position lies within an ellipse with square radius $R^2 = 1.72$. Since the radius-squared of a "one-standard-deviation" ellipse is 2.3, the actual position of the robot falls well within this confidence region.

On the other hand, consider the position marked **X'** (coordinates $[45\ 15]^T$) in Figure 6.3. Locations **X** and **X'** are relatively close together, so that we might expect little

difference in their position estimates. From the analysis we get that the expectation is about $[111\ 32.3]^T$, $\mathrm{var}(x_0)$ is 441.0, $\mathrm{var}(x_1)$ is 124.0, and $\mathrm{cov}(x_0, x_1)$ is 186.2; computing as above, this gives $R^2 = 14.6$. In the more familiar terms of one-dimensional statistics, a radius this big corresponds to about 3.4 "standard-deviations". Finally, we can compute the distance between the conditional expectation (spatial mean) and the robot's actual coordinates, and summarise:

Position	Coords.	"Mean"	Error	"Deviations"
X	$[15\ 15]^T$	$[76.5\ 20.0]^T$	63.2	<1
X'	$[45\ 15]^T$	$[111\ 32.3]^T$	68.0	3.4

In other words, the conditional expectation differs in physical distance from the actual position by about the same amount for both locations, so that the sensing system's performance is consistent. However, in the case of position **X'**, the variance of the estimates is much smaller than for position **X**, as it results in the actual position being interpreted as more "standard deviations" from the spatial mean.

In this way, it is possible to characterise how the variance of the system's position

estimates varies as it moves about the environment. Figure 6.4 represents the 166 variances by means of a histogram of the number of distributions in which the actual position falls within a given confidence radius R^2, where the match control parameter α is equal to 1000. Note that the histogram bins are labelled "chi-square", as if errors are assumed to be Gaussian R^2 has a χ^2 distribution. Bin 0 contains positions with chi-square less than 1, bin 1 contains positions with chi-square greater than or equal to 1, but less that 2, and so on to bin 12, which contains all positions with chi-square greater than or equal to 12 (since the "3-σ" region has $R^2 = 11.8$, essentially all positions where the actual position is interpreted as more than three standard-deviations from the mean are counted in this bin).

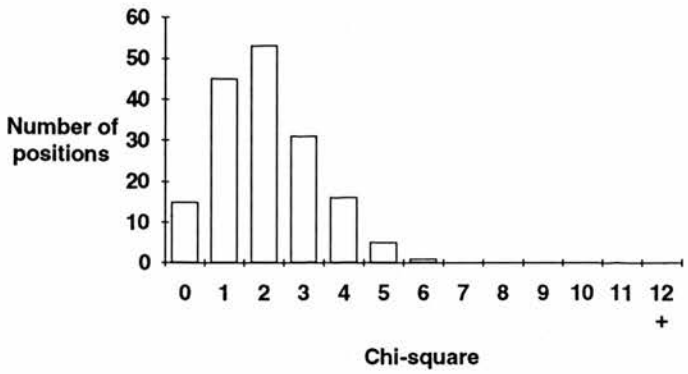

Figure 6.4. Histogram of conditional variances per position for $\alpha = 1000$

With the match control parameter α equal to 1000, we know that by the nature of the exponential matching function any intensity will match any other intensity, so the localisation likelihood distributions are in all cases flat, and their conditional expectations and conditional variances are equal. It follows that the shape of the histogram for $\alpha = 1000$ is a function of the configuration of sampling positions in space: points that happen to be close to the spatial mean contribute to small-numbered chi-square bins, and more distance points contribute to large-numbered bins. This histogram is thus a baseline for comparison with histograms generated for other α's.

If the histogram is computed for $\alpha = 1$, the result is as in Figure 6.5. The example discussed above with positions **X** and **X'** was computed with this α, so that in Figure 6.4, position **X** contributes a count to the column marked 1, and **X'** to the 12+ column.

It is clear that the variances are in general similar to the $\alpha = 1000$ baseline. However, it can be seen that in about 15 (9%) of the distributions, the true position actually falls outside of the "3-standard-deviation" ellipse ($R^2 = 11.8$); this suggests

that for these distributions, the variance is unjustifiably small. That is, where there is little intrinsic variability in its position estimates, the system might expect to be more confident of these estimates than when they have greater variance; but from the extrinsic variance (proportional to the size of the error bars on the accuracy plot), we know that this greater confidence is not warranted. This is a rather important observation, as the *intrinsic* variance of the position estimate distribution is the only means at the system's disposal for judging its own reliability; this issue will be considered in greater detail later.

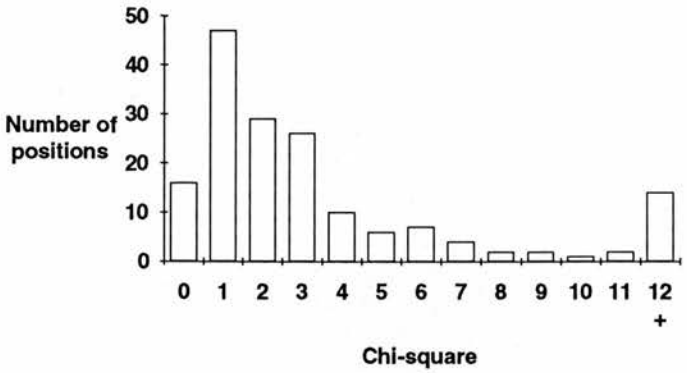

Figure 6.5. Histogram of conditional variances per position for $\alpha = 1$

Such histograms can be computed for any value of α. For Figure 6.6, the results for the three decades of α considered have been computed and sandwiched together into a single plot. It can be seen that, other than for very tight matching, the variances tend to be quite like the baseline.

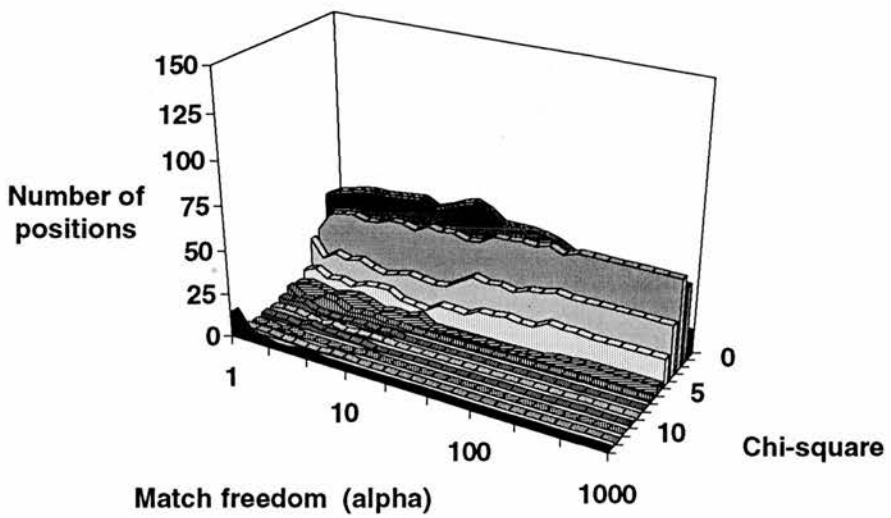

Figure 6.6. Histogram of conditional variances per position for acoustic beacon 1

6.1.2 Preliminary discussion

Having now presented the first analytic results, it seems appropriate to interrupt the progression towards more beacons, different sensors, and ultimately combinations of sensors, and discuss some of general considerations pertaining to this system and the analytic techniques as they have been applied to it.

System particulars

First, let us reconsider the sonic signal amplitude subsystem. At each of the 166 distinguished places in the environment, the robot has stored a signal intensity that represents what it received there on a past visit. Figure 6.7 depicts a typical pattern of intensities in the pen. Subjective examination of this pattern leads to several observations: for example, intensities in a fairly large area surrounding the loudspeaker are approximately equal (in fact, it was pointed out in Section 3.2.5 that this is due to the linear design of the microphone amplifiers, where the common value at these positions is the full-scale reading of the analogue-to-digital converter); it will not be possible for the robot to localise itself accurately within this region. Furthermore, there is another distinguishable "zone", aligned along the top of the diagram. This is an area with an overhanging bookshelf; as the wavelength of the sound is of the order of 50 cm, and the bookshelf sits at about this height above the floor, the sound presumably tends to reflect and concentrate in this area; elsewhere, the sound appears to dissipate more readily, and this forms a third "intermediate" zone.

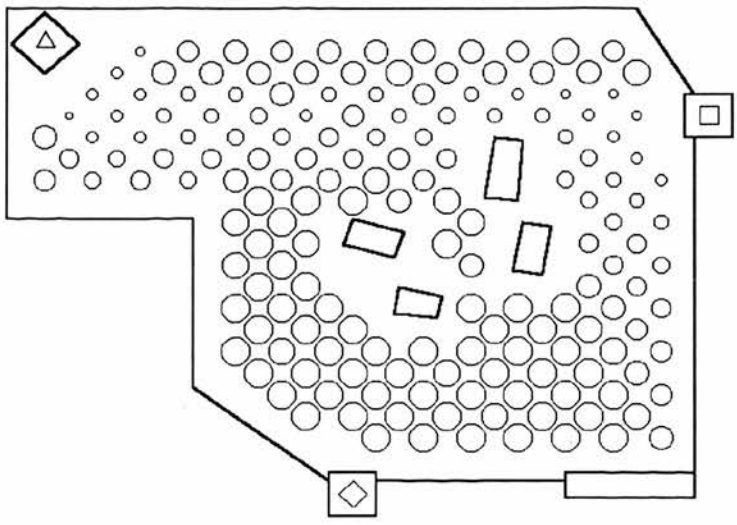

Figure 6.7. Typical intensity "map" for beacon 1 (at the diamond)

If the robot is at any position in the near-beacon region, therefore, it is likely to be

receiving a signal of maximum intensity. If it computes the probability that it is at each of the 166 marked positions, it will find that it has an equal chance of being at any near-beacon location, is less likely to be near the wall, and is even less likely to be in the intermediate zone. Similarly, if it is in the intermediate zone, it will probably receive a low-intensity tone; when it computes the likelihood of being at each position it will find that it is unlikely to be at any location near the beacon, is more likely to be in the wall zone, but is most likely to be somewhere intermediate.

Furthermore, if the robot can only specify its location as "in the near-beacon zone", then presumably it estimates itself to be somewhere near the centre of this region. If, by chance, it *is* near the centre, then it will have localised accurately; however, at worst it will be wrong by an amount equal to the distance between the centre and the furthest position from it (of the order of 1 m for the near-beacon zone in Figure 6.7).

The purpose of the localisation analysis is to translate vague, subjective notions such as these into some quantitative form that will allow the limitations of sensor systems and combinations of them to be evaluated objectively. In particular, it was suggested in Section 5.2.4 that a system should be judged on how certain it is was of the robot's location, and how accurate its position estimates turned out to be.

Localisation certainty

At any position, the sensor receives the beacon's tone at some intensity. It computes the likelihood that it is at each of the 166 positions in the pen, given the intensities it heard at these positions previously. It was argued in Section 5.2.4 that computing the entropy/information in this distribution is a measure of the localisation system's certainty of where it thinks the robot is. But do the results (see Figure 6.1) confirm our subjective impression of how certain about "place" it is possible to be, given a set of intensities like the ones mapped in Figure 6.7? It seems so. When the matching criterion is such that mapped and current intensities must be very similar before a match becomes reasonably probable, there are about 4 bits of information in this intensity field, so that the robot could in principle distinguish between 16 regions in the environment; these presumably correspond to the high-intensity area near the beacon, the area along the wall, and perhaps a finer decomposition of the intermediate-intensity zone.

As α is increased, less precise matches become probable; when $\alpha = 10$, the information has decreased to less than 2 bits, which coincides directly with the

subjective impression of about three or four distinguishable areas. In terms of position estimates and the exponential matching function employed, an α of 10 means that if the intensity presently being received is I, then the robot assumes it has a 50% chance of being at any position where the mapped intensity was about $I \pm 20$ intensity units, or a 90% likelihood of being somewhere with a mapped intensity of $I \pm 8$. It is easy to accept that such a match freedom will wash out any differences between positions within the near-beacon zone, the wall-zone, and the intermediate zone.

When α enters its second decade, the difference between intensities rapidly disappears, and all positions become indistinguishable. In Figure 6.1 this is seen as a drop in information to zero (maximum entropy). In other words, the entropy or information measures proposed seem to capture, in a quantitative way, our intuitive notion of how certain the robot can be of where it is, given only a single loudspeaker as a locative reference.

Localisation accuracy

Again, with the robot at any position, the sound sensor receives the beacon's tone at some intensity, and the likelihood that the robot is at each of the 166 positions in the pen can be computed, given the intensities it heard at these positions in the past. From this distribution, two methods for estimating the robot's actual position have been conjured with (see also Section 5.2.4). The first weights the coordinates of each of the possible positions by the probability that the robot is there; the result is the conditional expectation, or spatial mean. The second simply picks the most likely position (the one with maximum probability), and assumes the robot is there; this is the MAP or "maximum *a posteriori* probability" position, or spatial mode.

It is easy to imagine pathological conditions under which one of these is to be preferred over the other. For instance, suppose that the match criterion is very strict (α is small), and that the possible locations for the robot are clustered in a small area, except for one possible location where mapped and current intensity match ever so slightly better than anywhere else, even though this location is actually far away from the principal cluster (such conditions might easily prevail in the intermediate zone of Figure 6.7). Then, MAP estimation grabs this absolute maximum position, ignoring the group of almost-as-likely locations. The conditional expectation, on the other hand, takes the likelihoods of the other positions into account, and if there are several of them, the influence of the distant point will be diminished. Of course, if the robot actually *is* at the distant position, then the MAP estimator was best; however, in the cluster scenario as described here, it does seem more sensible to choose the

expectation estimator and assume the actual location is near the cluster. The accuracy results in Figure 6.2 seem to show exactly this effect. With the match parameter α equal to 1, the MAP estimator is about 12% less accurate than the conditional expectation.

However, suppose that the match freedom is increased. Then, an increasing number of positions become plausible candidates for the robot's location, and the conditional expectation presumably tends towards the centre of the set of positions. The MAP estimator, assuming it is at all reasonable, will tend to be immune to the addition of new positions, as these will not be more probable than the current MAP. Again, this effect is visible in Figure 6.2. In the second decade of α, the conditional expectation estimator becomes monotonically worse, while the MAP estimator remains steady. As α is increased further still, of course, the differences in match likelihood between positions diminish, so that neither conditional expectation nor MAP do better than to estimate the robot's position as being at the centre of the set of possible positions.

These two metrics seem to behave qualitatively as we would hope them to, judging by the results in Figure 6.2, and it could be concluded that for strict matches, the conditional expectation ought to be used, but that the MAP estimator is more appropriate where there is greater match freedom. But are the *quantitative* results of these analyses consistent with intuition? Again, it seems so. Our subjective impression, sustained by the certainty analysis, suggests that a robot operating with the sound sensor provided and only beacon 1 can distinguish about three places in its domain. In Section 6.1.1 it was suggested that, with the accuracy results as they are, the actual position of the robot could actually lie on average anywhere within an area of about 2.9 square metres centred on its estimate. This area represents about one third of the floor surface, and is in agreement with a subjective impression of three regions not vastly dissimilar in size.

When α is such that any mapped intensity is likely to match almost any current intensity ($\alpha > 100$), both estimators produce positioning errors of the order of 1 m, which is what would be expected if the sensing system were always to estimate that the robot is near the centre of the pen, independently of where it actually is.

Estimate variance

With the beacon's tone arriving at some intensity, a distribution of likelihoods that the robot is at each of the 166 positions in the pen is computed, given the intensities it

heard at these positions in the past. The conditional expectation averages the coordinates of the set of positions where the robot might be, weighted by its likelihood of being at each of them. In a spatial sense, this is the mean of the distribution; the conditional variance is meant to quantify the dispersion of the estimates around this spatial mean.

For every position in the pen, there is a different distribution of likelihoods of being at each of the 166 distinguished spots. For each of these, the conditional expectation can be computed as an estimator of the robot's actual position, and the conditional variance as a measure of the shape of the distribution. In the accuracy analysis, it was possible to compute the average difference between the robot's actual position and the conditional expectation as an error metric. Conditional variance actually specifies the variability in position estimates in the two spatial dimensions of the experimental pen's floor, as well as the covariability between these dimensions. In Section 6.1.1, it was suggested that the three dimensions of conditional variance could be usefully collapsed into a single number, which was the confidence region that the robot's actual position fell in.

From Figure 6.6, it is apparent that, other than for very strict matching, the variability of the system's position estimates is consistent. Almost everywhere, the variance is such that the actual position lies between the 40% and 78% confidence regions (corresponding to the range between $\chi^2 = 1$ and $\chi^2 = 3$, or histogram bins 1 and 2). When α is quite small, however, there are several positions at which the system estimates its variability to be much smaller than it ought to. As an extreme example, with $\alpha = 1$ there is a position at (floor-grid) coordinates $[35\ 35]^T$ for which the conditional expectation is $[32.7\ 21.4]^T$. The estimate is therefore only in error by 13.8 units (about 30 cm). However, the variance of candidate positions is excessively small: $\mathrm{var}(x_0)$ is 103.4, $\mathrm{var}(x_1)$ is 10.30, and $\mathrm{cov}(x_0, x_1)$ is 14.63, so $\chi^2 = 22.8$. This χ^2 corresponds to the 99.999% confidence interval, or about 4.4 standard deviations in one-dimensional statistical terms.

The final question in this preliminary discussion is, "do these variances seem plausible in light of the mooted subjective interpretation of one-beacon sonic localisation?" Again, it seems so. An intuitive decomposition of space distinguishes the three broad regions near the beacon, near the wall, and everywhere else. Independently of match freedom, we would expect that if the robot is anywhere in the near-beacon zone, it will generate plateau-shaped position estimate distributions, where all near-beacon positions are "on" the plateau, and all others lie on a flat valley below. For

these positions, a fairly consistent amount of variance of candidate positions around the spatial mean is plausible. The same observation is likely to hold for the wall zone, again independently of match freedom.

With loose matching, in fact, a similar plateau is likely even for the intermediate zone. When the matching operation is very strict, however, the varied range of intensities recorded in this zone mean that it is possible for the system to find only a very small set of likely matches; under these circumstances, the variance of its estimates will be very small. Indeed, it is found that the 15 or so positions in the 12+ bin in Figure 6.4 all fall within the subjective intermediate zone.

Summary

It thus seems reasonable at this point to suggest that, at least for the sonic system considered here, the analytical tools proposed do produce plausible results, at least insofar as they agree well with intuitive impressions of how successful localisation is likely to be, considering the nature of the sensor data on which it is based. The next step, therefore, is to consider the effects of adding additional sonic beacons.

6.1.3 Localisation by beacons 1 and 2

Each of the three loudspeakers generates a distinct tone. Therefore, if both beacon 1 and beacon 2 are activated, each position in the pen is identified with a *pair* of intensities, namely the intensity of the signal from beacon 1, and the intensity of beacon 2's tone. In Figure 6.8, two sets of circles are plotted: the solid black ones show the pattern of beacon 1 signal intensities, just as in Figure 6.7; the hollow circles indicate recorded intensities of beacon 2's signal.

Purely subjectively, again, it can be seen that there are recognisably distinct regions in the two-beacon environment. For example, there is a zone near beacon 3 (marked by a □) where both signals are weak; there is a wall zone again, where both signals are quite strong; there are now near-beacon zones for both beacons; and there are numerous other regions such as one in the centre of the pen where both signals are approximately equal and moderately strong. Therefore, it should be the case that the information content of the intensity-pair field is greater than the field for beacon 1 alone, and that localisation by means of a pair of beacons is more precise that localisation by a single one.

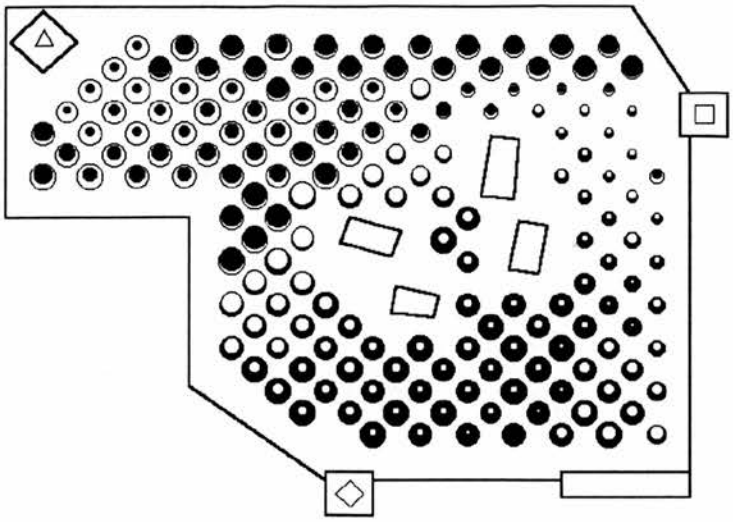

Figure 6.8. Typical intensity-pair pattern for beacons 1 and 2 (◊ and Δ)

In order to analyse the effect of the additional beacon, it is necessary to integrate the two sets of signal intensities. There are essentially two ways in which this can be done. Either a new intensity map can be devised in which the pair of intensities stored for a particular position are assigned a single composite label, or the localisation estimates for each map can be computed independently, and the estimates combined. It was shown in Section 5.2.5 that these operations of composition and combination are effectively equivalent, provided the composition operation is defined to be multiplicative. Suppose that localising with two beacons proceeds as follows, where (as in Section 6.1.1) data set 1 specifies the "current" data, and set 2 specifies the "map" data:

for each of the 166 positions A_j:

1. let the intensity of the beacon 1 signal that would be heard in that position be B_1, such that $B_1 = \text{DataSet1}(A_j, 1)$, and let the intensity of beacon 2's signal heard there be B_2, with $B_2 = \text{DataSet1}(A_j, 2)$; and
2. as discussed in Section 5.2.3, compute

$$P(A_j \mid B) = \frac{P(B \mid A_j)}{\sum_{i=1}^{166} P(B \mid A_i)}, \qquad j = 1,\ldots,166,$$

where

$$P(B \mid A_i) = e^{-\{[B_1 - B_1(A_i)]/\alpha\}^2} \cdot e^{-\{[B_2 - B_2(A_i)]/\alpha\}^2},$$

with $B_1(A_i) = \text{DataSet2}(A_i, 1)$ and $B_2(A_i) = \text{DataSet2}(A_i, 2)$, and α the match strictness parameter.

In other words, assume that the robot visits each marked position in the pen, and receives the same pair of signals there as it did when data set 1 was collected (step 1 above), but that it does not itself know where it is; this "current" signal is therefore matched against the "stored" values for each of the 166 positions collected as data set 2 (step 2); given that it is receiving signals of intensities B_1 and B_2 at the moment, its likelihood of being at each of the 166 positions is computed. Thus we have 166 distributions of match likelihoods for each of the 166 positions (taken to represent the probabilities that the robot is at each possible position), which may be expressed as

$$P\{A_j|\text{DataSet1}(A_i,1),\text{DataSet1}(A_i,2)\},\ \ j,i=1,\ldots,166.$$

The critical step is in the definition of $P(B|A_i)$. In step 2 above, it is computed as the *product* of likelihood of a match between the mapped and current values for beacon 1, and the equivalent match likelihood for beacon 2. This is an entirely reasonable operation: when both beacon intensities match well, the product of the matches is large; when both match poorly, the product is small; and in intermediate cases, for example, if the match for one beacon is similar for two positions, then the product is a function of the quality of the match for the other beacon, which is as we would expect. Therefore, we have a plausible composition operator which is by definition multiplicative; from the relationship in Section 5.2.5 it is thus not necessary to construct and analyse a composite map at all, and steps 1 and 2 above are superfluous. Instead, the analysis for two beacons becomes:

1. retain the 166 distributions of match likelihoods for each of the 166 positions computed in Section 6.1.1, for beacon 1, which were

$$P\{A_j|\text{DataSet1}(A_i,1)\},\ \ j,i=1,\ldots,166;\ \text{and}$$

2. compute equivalent distributions in the same way for beacon 2 signals, giving

$$P\{A_j|\text{DataSet1}(A_i,2)\},\ \ j,i=1,\ldots,166\,.$$

The combined distributions are then simply the renormalised products of these individual distributions, as in Section 5.2.5, that is,

$$P\{A_j|\text{DataSet1}(A_i,1),\text{DataSet1}(A_i,2)\}=\frac{P\{A_j|\text{DataSet1}(A_i,1)\}\,P\{A_j|\text{DataSet1}(A_i,2)\}}{\sum_{k=1}^{166}P\{A_k|\text{DataSet1}(A_i,1)\}\,P\{A_k|\text{DataSet1}(A_i,2)\}},$$

where $j,i=1,\ldots,166$. Thus, the same technique is used for combining the qualitative intensity maps for two beacons as will eventually be used for combining the

localisation estimates of two completely different sensory modalities.

The information in the position likelihood distributions computed in this manner is presented in Figure 6.9. These results are very much as expected: where the match freedom is very small, the system has about 2 more bits of information than it had when there was only one beacon, so that it can distinguish about 4 times as many regions. Again, when $\alpha = 10$, less precise matches become probable, and the information has decreased to about 3 bits, which fits well with a subjective segmentation of the intensity pairs into eight or nine distinguishable areas. Just as with one beacon, when α enters its second decade, the difference between intensities disappears quickly, and all positions become indistinguishable; the information drops to zero.

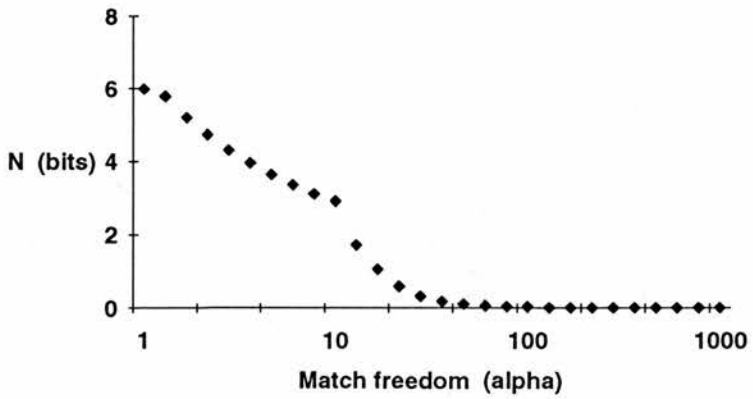

Figure 6.9. Information in the acoustic fields of beacons 1and 2

Similarly, analysing the accuracy of the sonic system with two beacons shows some trends that are shared with the one beacon case. In Figure 6.10 it can be seen that the conditional expectation is on average a better estimator of position than the MAP where α is less than 10, that the MAP tends to find and hold onto good estimates during α's second decade, and that both are about equally poor when the match freedom is high.

The absolute accuracy for matches with α less than 50 is greater than for the one beacon case. In fact, with α less than 10, the average estimate error is approximately half as large as it was for one beacon, so that the second reference signal very obviously improves localisation.

The variabilities in the distributions are compared by the same histogramming procedure as was employed in Section 6.1.1 for a single beacon. The compiled

histograms for the three decades of α considered in this analysis are shown in Figure 6.11. Here, we see some substantial differences with the results for one beacon alone. First of all, it is clear from the $\alpha = 1000$ slice that, as we would expect, with total freedom of matching the variances (and therefore the histogram) are identical to the single beacon case. However, the histogram for $\alpha = 1$ (see Figure 6.12) is much more U-shaped for two beacons. That is, there are now many positions (more than 50, so over one third of all positions) for which the system estimates its variance to be smaller than it ought to, based on the actual errors it is making. Unlike the single-beacon case, this U-shape of the histogram persists for α's up to 5 or 6.

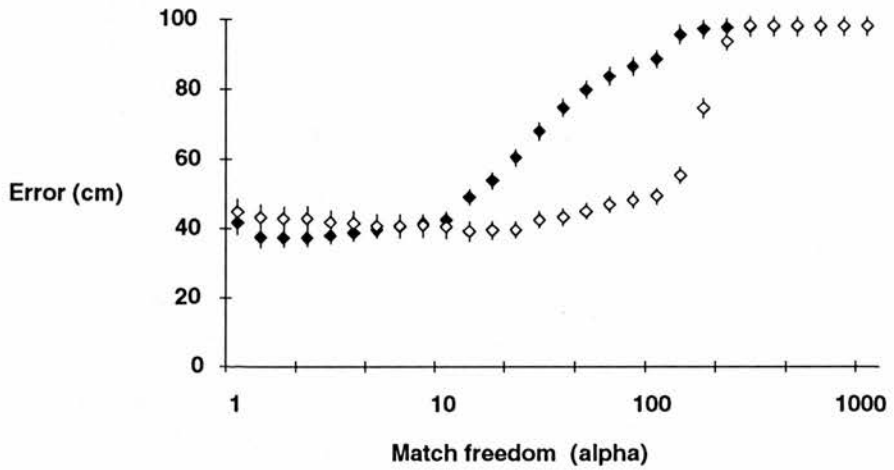

Figure 6.10. Accuracy of localisation by beacons 1 and 2; ♦ marks conditional expectation as an estimator, and ◊ denotes the MAP estimator

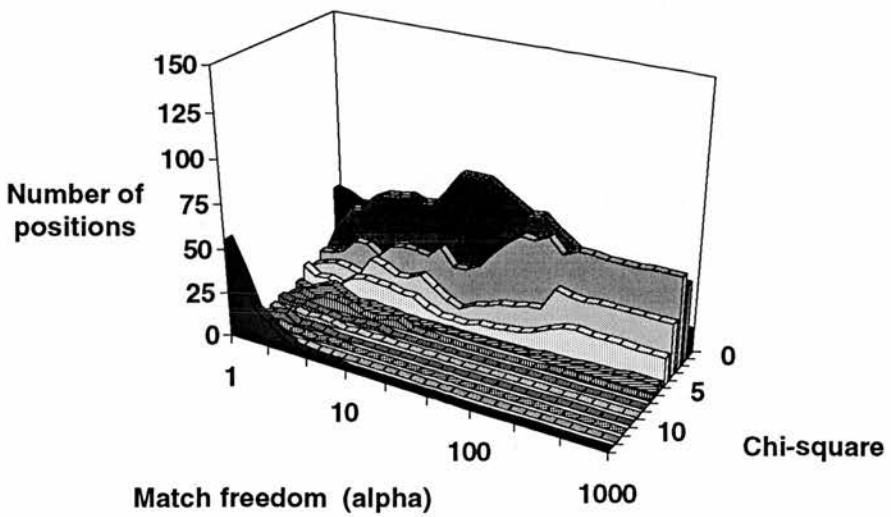

Figure 6.11. Histogram of conditional variances per position for beacons 1 and 2

Furthermore, for $\alpha = 30$ it seems that the variances for many positions are perhaps larger than the actual errors would require them to be (almost all positions are in chi-square bin 1). The variabilities seem to progress from being too small on average in the first decade of α, to unexpectedly large in the second; they are the same as in the third decade as they were for a single beacon, which is consistent with the inevitability of identical, flat position estimate distributions in both cases.

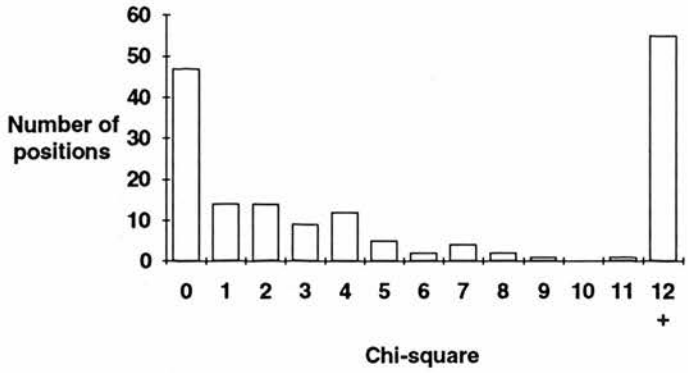

Figure 6.12. Histogram of conditional variances per position for $\alpha = 1$

These variability trends can be partly rationalised as they were for beacon 1 alone. For example, when the matching operation is strict, the increased dimension of variability in the intensity pair in what were called "intermediate" zones, means that the system is more prone to finding only a very small set of likely matches, and under these circumstances, the variance of its estimates will be small; however, there are the wall and beacon 3 zones where both signals have the same intensities everywhere in each zone, so that there are still places where the variability is always be large even though α is small. This would explain the U-shaped histograms for small α (e.g., Figure 6.12).

As α enters its second decade, the differences between intensities are washed out. This has little effect on the zones of perpetually large variance, of course, but it has a significant effect on the intermediate regions which had small variances with $\alpha < 10$. Here, previously unlikely matches in other, possibly distant, intermediate areas can suddenly become likely, so that while conditional expectation might still ultimately be a reasonable estimator of position, the conditional variance could increase disproportionately because of the new, confounding matches. This effect is presumably maximal near $\alpha = 30$.

With larger α, the conditional expectation becomes steadily worse on average as a position estimator, and the variances equalise as all positions become equally likely.

This explains the identical shape of the histograms for the single and double beacon situations with α in its third decade.

Summary

Again, therefore, it seems that the qualitative and quantitative results of the analysis are plausible in comparison with an intuitive account of two-beacon sonic localisation. Where matching between mapped and current data is sufficiently strict, the robot has more information about place, and the average error it makes when using either the conditional expectation or MAP position estimators is significantly lower than it was for a single beacon.

The overlap between the the position estimate distributions for the two beacons is given in Figure 6.13. The ♦ mark the average overlap across the 166 sampling positions in the experimental pen, and the grey lines mark the range between maximum and minimum overlap for each α. It can be seen that when matching is strict, estimates based on the two beacons overlap very little on average, and at best overlap by 30 to 50%. The overlap between estimates from the two beacons increases as matching becomes less strict. This is not surprising, as the effect of relaxing match strictness spreads the distributions, so that more of the combined probability density falls under both distributions. For very loose matching, this effect results in estimates from the two beacons agreeing completely. However, as discussed previously, neither beacon leads to a very accurate estimate of position.

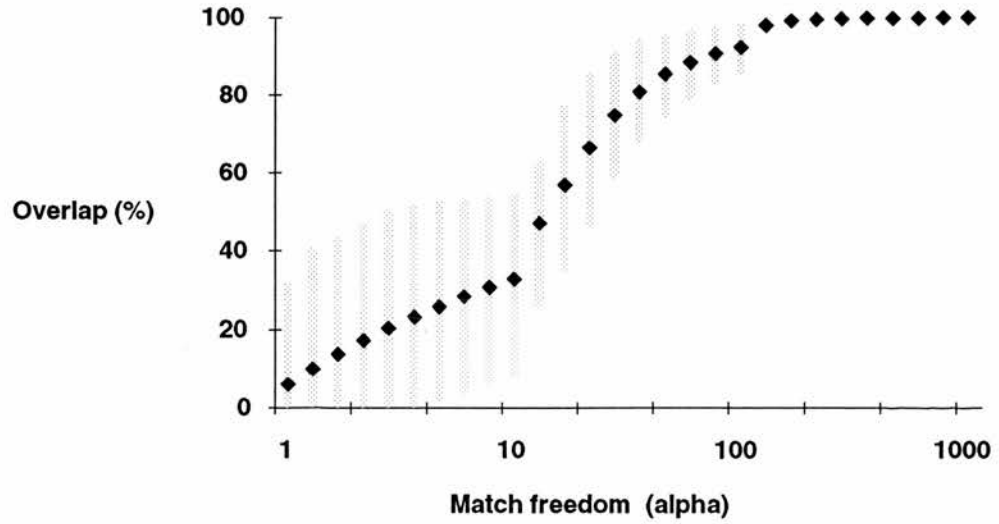

Figure 6.13. Overlap between estimates for beacons 1 and 2; the ♦ marks the average overlap, and the extent of the line denotes the range between maximum and minimum overlap

6.1.4 Localisation by all three beacons

When the third beacon is activated, each position in the experimental environment can be labelled by a triple of intensities, whose components are the individual intensities at that position of each of the three distinguishable beacon tones. From the effects of adding a second beacon to the first, we would expect that adding a third ought to increase the amount of information the robot has about its place in the pen, and decrease the average error in its position estimates.

In performing the analysis for three beacons, it seems sensible to proceed by extending the two-beacon case in the same manner as was extended the one-beacon case, with the same justifications (see Section 6.1.3):

1. retain the 166 distributions of match likelihoods for each of the 166 positions computed in Section 6.1.1, for beacon 1, namely

$$P\{A_j|\text{DataSet1}(A_i,1)\},\ j,i=1,\ldots,166;$$

2. retain the 166 distributions of match likelihoods for each of the 166 positions computed in Section 6.1.3, for beacon 2, namely

$$P\{A_j|\text{DataSet1}(A_i,2)\},\ j,i=1,\ldots,166;\ \text{and}$$

3. compute equivalent distributions in the same way for beacon 3 signals, giving

$$P\{A_j|\text{DataSet1}(A_i,3)\},\ j,i=1,\ldots,166.$$

Again, the new distributions whose information and accuracy will be measured are just the renormalised products of these individual distributions, i.e.,

$$P\{A_j|\text{DataSet1}(A_i,1),\text{DataSet1}(A_i,2),\text{DataSet1}(A_i,3)\}=\frac{\prod_{m=1}^{3}P\{A_j|\text{DataSet1}(A_i,m)\}}{\sum_{k=1}^{166}\left\{\prod_{m=1}^{3}P\{A_k|\text{DataSet1}(A_i,m)\}\right\}},$$

where $j,i=1,\ldots,166$.

In Figure 6.14, the information in the sonic field of the three beacons is plotted as a function of the match control parameter. Compared with one (Figure 6.1) or two beacons (Figure 6.9), it can be seen that the additional beacon has increased the amount of information about place available to the system when matching is strict. On

average, adding beacon 2 to beacon 1 increased *N* by about two bits, whereas adding beacon 3 to these first two adds only one extra bit of information; this is not surprising, as there is no reason to assume that the number of beacons and the degree of segmentation of space are linearly related.

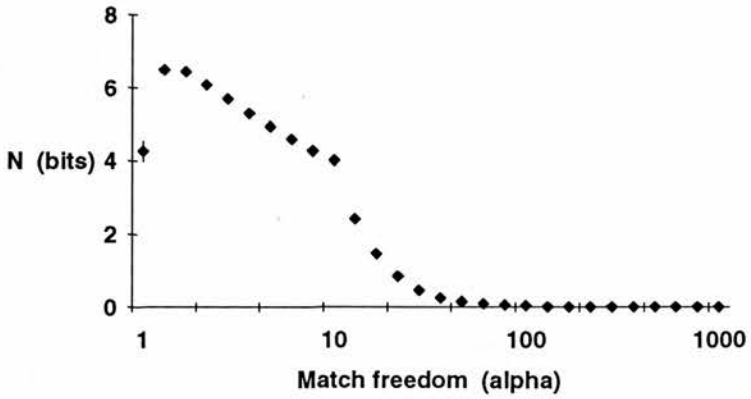

Figure 6.14. Information in the acoustic fields of beacons 1 to 3

However, the new constraint does have a surprising effect on the amount of information in the sonic field when $\alpha = 1$. Here, the average information across the room is less than it was for two beacons, and about the same as it was for one beacon; the variance of the information is also quite large. Scrutiny of the data determines that this is caused by disagreement between the position estimate distributions for the three beacons, within what in previous sections was referred to as the "intermediate zone", where a given combinations of intensities was characteristic of relatively few positions.

With the addition of the third constraint, it is expected that large, mostly indistinguishable regions (e.g., near the beacons) should become more finely partitioned, so that the entire pen eventually takes on an "intermediate zone" diversity of intensities. That this is so can be seen in Figure 6.15, where typical intensities of beacon 1's (◊) signal are plotted as solid black circles, those from beacon 2 (Δ) as hatched circles, and those from beacon 3 (□) as solid white circles.

In the previous cases, position estimate distributions in the intermediate zones were relatively sharp peaks (the number of possible matches is small and hardly variable). Here, a position estimate distribution is computed for each beacon, and these are combined multiplicatively. Where these distributions are in agreement, their product is also sharply peaked, with low entropy (near 0) and high information content. On the other hand, if a peak in the distribution for one beacon does not quite agree with the other two, the product distribution is flattened, with maximum entropy

and no information content. With $\alpha = 1$, this kind of instability in the matching operation apparently arises quite frequently.

As the match freedom is increased even slightly ($\alpha = 2$ or $\alpha = 3$), the peaks of the position estimate distributions for intermediate zone positions are sufficiently widened that the product distributions, while still perhaps mainly flat, do end up with some probability density concentrated in one area; and any tendency away from flatness increases the amount of information in a distribution. Therefore, the additional beacon increases the amount of information available to the system on average, but when matching is set to be very strict, this third constraint can equally be a source of confusion, given that the intensity recorded at the same position for the same beacon on two different visits is rarely identical.

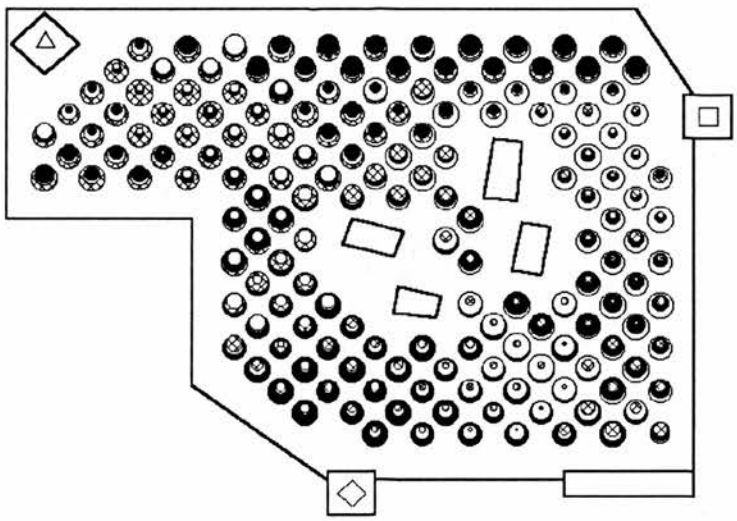

Figure 6.15. Typical intensity-triple pattern for all three beacons

From the certainty results, we would expect that the average positioning accuracy should be affected by this matching instability, but that it should otherwise show similar trends to the accuracy results for one and two beacons. Furthermore, the absolute accuracy of positioning ought to be increased by the third beacon. In Figure 6.16. we see that these intuitions are well founded. For $\alpha = 1$, the accuracy of the conditional expectation and the MAP as position estimators is less than it was for only two beacons. However, both estimators become about 30% more accurate than they were with only two active beacons. Again, the trends are that conditional expectation is as good as MAP for small α, that the MAP estimator tends to stay good over the second decade of α, and that both are equally bad when match freedom is high, as any intensity triple presumably matches any other.

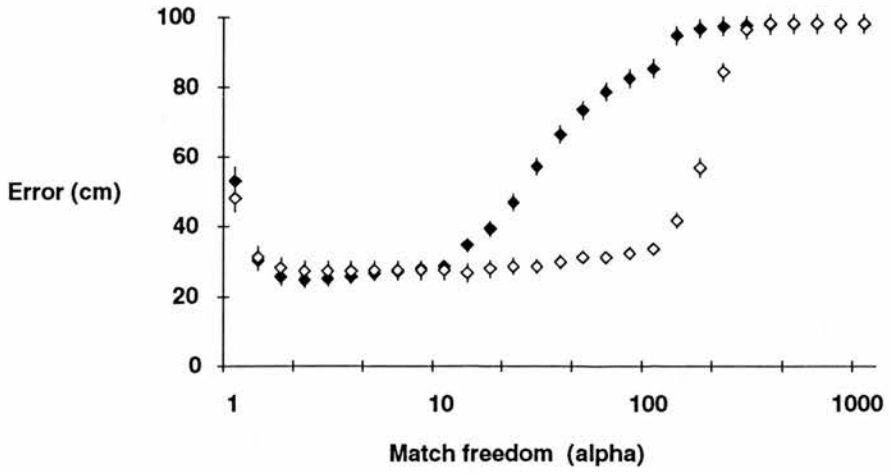

Figure 6.16. Accuracy of localisation by beacons 1 to 3; ♦ marks conditional expectation as an estimator, and ◊ denotes the MAP estimator

It is also unsurprising that the pattern of variances (Fig 6.17) is very much like it was for two beacons. The histogram at $\alpha = 1$ has a pronounced U-shape as a result of the many positions where the estimate variances are lower than they ought to be, and this U-shape persists almost throughout the entire first decade of α. The baseline shape at $\alpha = 1000$ is naturally the same as for the single and double beacon analyses.

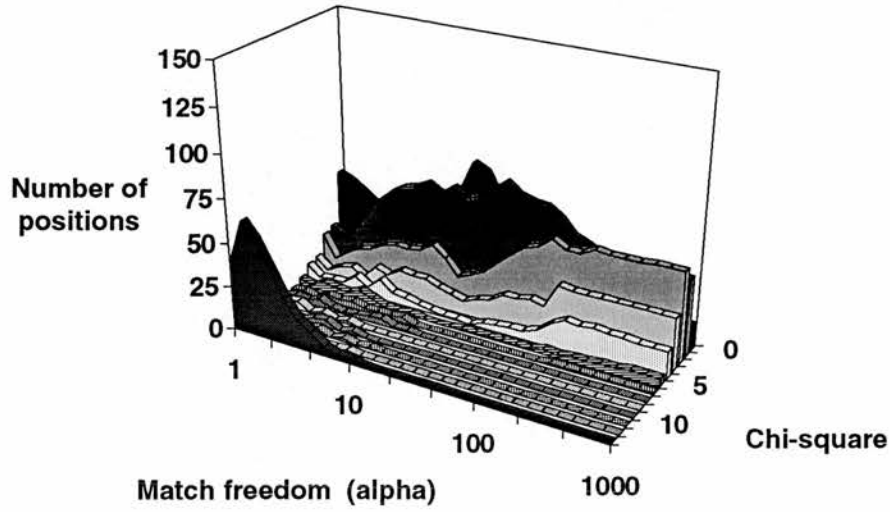

Figure 6.17. Histogram of conditional variances per position for beacons 1 to 3

The overlap between the the position estimate distributions for all three beacons is given in Figure 6.18. When matching is strict, estimates based on the two beacons overlap very little on average, and at even best overlap by less than 20%. As for two beacons, the overlap between estimates from the three beacons increases as matching

relaxes, due to the spreading of the distributions. Again, this effect results for very loose matching in estimates from the two beacons agreeing completely (on an inaccurate estimate).

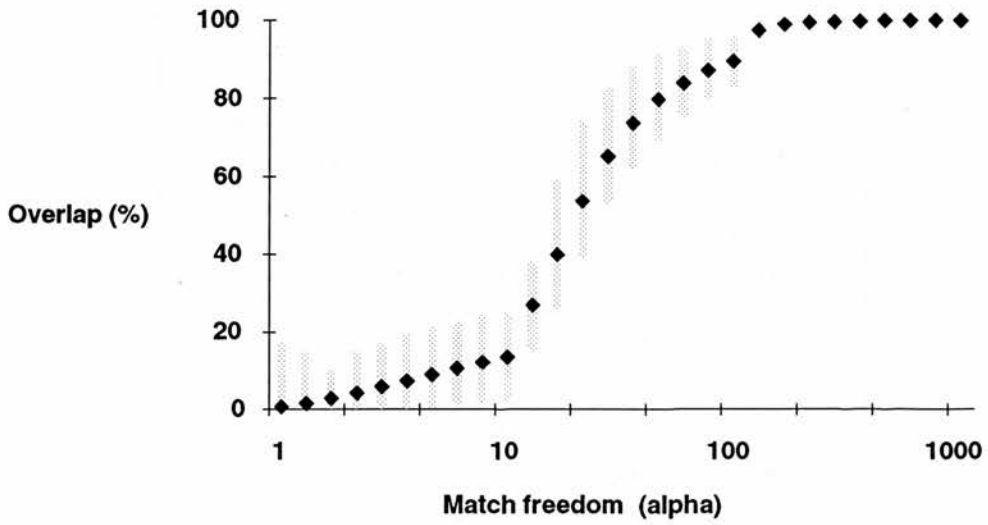

Figure 6.18. Overlap between estimates for all three sonic beacons

6.1.5 Comparison of beacons and beacon combinations

In the foregoing analysis, beacon 1 was considered alone, and then in combination with beacon 2. In order to show that the same arguments could have been made for either of beacon 2 or 3 alone, and for the other possible pairings 1&3 or 2&3, the information and accuracy results for all seven beacon combinations are now briefly summarised.

The information content of the various individual beacon and combination fields is given in Figure 6.19. It is clear that the entropy/information measures for each of the three individual beacons is nearly identical. Similarly, the three possible beacon pairs are also very closely associated in information terms. As previously discussed, for the first decade of α, pairs of beacons give more information about place than single beacons, and the trio gives more information than any of the pairs.

In Figure 6.20 these same seven beacon combinations are compared by the accuracy of the conditional expectation as a position estimator. Here again, it is generally true that less accuracy obtains from single beacons than from pairs of beacons, and that with all three beacons accuracy is maximised. As the match freedom is increased, however, all combinations become increasingly inaccurate, and in the second decade of the α, there is an anomaly where beacons 1 and 3 together are less

accurate on average than beacon 2 alone. This is a consequence of the fact that beacon 2 on its own is significantly more accurate than either of beacons 1 and 3, so that beacon 2 compares favourably even when these two are taken in combination.

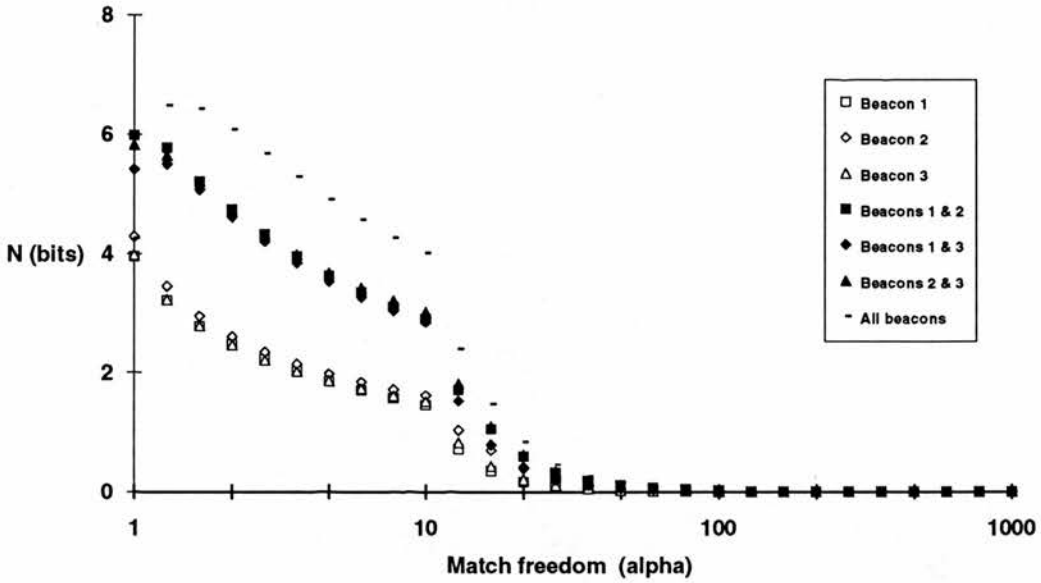

Figure 6.19. Information in the acoustic fields of all seven beacon combinations

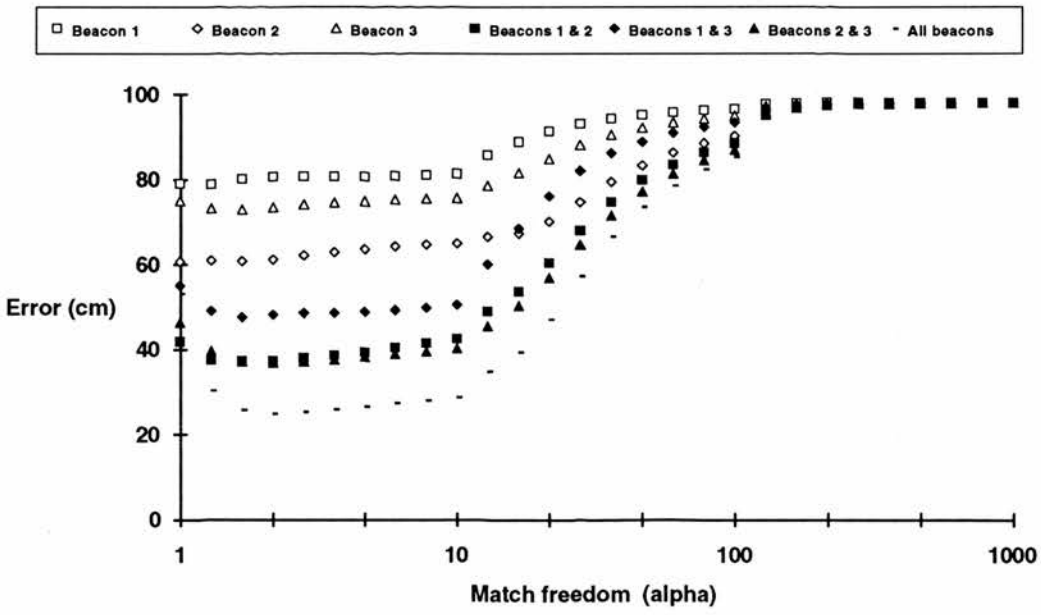

Figure 6.20. Accuracy of localisation for all seven beacon combinations (with conditional expectation as an estimator only)

6.1.6 Summary of sonic amplitude localisation

For each of three tone generators, a qualitative acoustic map of the experimental pen consists of a list of signal intensities measured at each of 166 positions distributed evenly across the floor surface. In this section, these maps were evaluated in terms of how certain a robot using them can be about where it is, and how accurately it is able to position itself with their aid. The results of this analysis are summarised below.

Certainty

When the match strictness parameter α is less than 10, the robot can distinguish about 3 or 4 regions in the pen with one beacon, 8 or 9 regions with two beacons, and perhaps 16 regions with all three beacons, which agrees with subjective decompositions of the signal intensity patterns for the various beacon combinations. For the exponential matching function employed, an α of 10 means that if the intensity presently being received is I, then the robot assumes it has a 50% chance of being at any position where the mapped intensity was about $I \pm 20$ intensity units, or a 90% likelihood of being somewhere with a mapped intensity of $I \pm 8$. As α is increased beyond 10, the localisation certainty of the system drops quickly to zero.

Accuracy

When α is less than 10, the conditional expectation of the position likelihood distributions is the best estimator of position, on average. For one beacon, estimates are in error by about 80 cm on average, for two beacons by about 40 cm, and for all three beacons by 30 cm. As α is increased through an order of magnitude to 100, the conditional expectation steadily worsens as an estimator, while the MAP estimator remains almost as good as the expectation was in α's first decade. With α greater than 100, both the conditional expectation and the MAP estimators degrade to placing the robot somewhere near the centre of the pen independently of where it actually is (positional error on average 100 cm).

Variance

The conditional variance of the system's position estimates is generally consistent, with two exceptions: variance is sometimes smaller than it ought to be in the first decade of α, and is sometimes larger than we would expect in α's second decade. The first exception is caused by the instability of the matching process when α is small, which leads to situations where the number of candidate positions for the robot and the

variability in these estimates is also small; these variances are usually smaller than the actual precision of the estimates would warrant, so that the system is overconfident in these cases. This effect increases with the number of active beacons, as the instability of strict matching is worse when it is the product of several individual strict matches. The second exception arises, with α between 10 and 100, where the differences between intensities disappear; this has significant consequences in the regions which had small variances with $\alpha < 10$. Here, previously unlikely matches in other, possibly distant, areas can become likely, so that while the position estimates might still ultimately be a reasonable, the conditional variance occasionally increases disproportionately because of the new, confounding matches. This effect is found to be maximal near $\alpha = 30$.

Synopsis

The sonic signal intensity localisation system works best when the matching between what it has heard in the past and what it is hearing presently is defined to be quite strict. Naturally this requires that the signal strengths of the sonic beacons remain reasonably constant over time. Under these circumstances, the system benefits in terms of certainty and accuracy from increasing the number of tone generators, at least as they have been arranged in the experimental pen considered here.

6.2 Infrared beacon detector

A qualitative infrared map of the experimental pen consists of a list of pairs of numbers for each of the 166 sampling positions, as described in Section 5.2.2. The first number in the pair is the relative orientation at which the beacon is perceived to be (its bearing), and the second the amplitude of the beacon's signal in that direction. Such a map can be constructed for each of the three beacons, and it is the analysis of the certainty and accuracy of localisation by means of these maps that is the subject of this section. The analytical procedures to be followed have much similarity with those described in the foregoing discussion of the sonic system; this section will therefore begin with a discussion of localisation with a single infrared beacon, and then proceed directly to the case where all three beacons are active.

6.2.1 Localisation by infrared beacon 1

Suppose again that data set 1 (collected on the robot's first journey around the

environment) specifies the "current" data, and set 2 specifies the "map" data, where "Bearing*n*(*pos*,*beac*)" denotes the bearing of beacon *beac* that was perceived at position *pos* on mapping visit *n*, and "Int*n*(*pos*,*beac*)" denotes the intensity of the signal in this direction. The localisation analysis then proceeds as follows:

for each of the 166 positions A_j:

1. let the bearing at which the beacon would be seen in that position be C, such that $C = \text{Bearing1}(A_j,1)$, and let the intensity of the signal in that direction be B, such that $B = \text{Int1}(A_j,1)$; and
2. as discussed in Section 5.2.3, compute

$$P(A_j|B,C) = \frac{P(B,C|A_j)}{\sum_{i=1}^{166} P(B,C|A_i)}, \qquad j = 1,\ldots,166,$$

where

$$P(B,C|A_i) = \begin{cases} 0 & \text{if } C \neq C(A_i) \text{ or } B(A_i) = 0 \\ e^{-[[B-B(A_i)]/\alpha]^2} & \text{otherwise,} \end{cases}$$

with $B(A_i) = \text{Int2}(A_i,1)$, and $C(A_i) = \text{Bearing2}(A_i,1)$, and α a parameter that controls match strictness.

In other words, assume that the robot visits each marked position in the pen, and perceives the infrared beacon to be at the same bearing and intensity there as it did when data set 1 was collected (step 1 above), but that it does not itself know where it is; this "current" data is therefore matched against the "stored" values for each of the 166 positions collected as data set 2 (step 2); given that it is receiving a signal of intensity B in direction C at the moment, its likelihood of being at each of the 166 positions is computed. Just as with the sonic system, we thus have 166 distributions of match likelihoods for each of the 166 positions (taken to represent the probabilities that the robot is at each possible position), which may be expressed as

$$P\{A_j|\text{Int1}(A_i,1), \text{Bearing1}(A_i,1)\}, \quad j,i = 1,\ldots,166.$$

Furthermore, since the infrared signal intensities also range from 0 to 255, it makes sense to compute the certainty and accuracy measures with respect to the same range of the match strictness parameter α, namely from 1 to 1000 (actually from 2.55 to 2550, as discussed in Section 6.1.1).

In fact, the only major difference between the infrared and sonic systems is that the infrared signal has the added dimension of bearing. This quantity comes into play in

the matching operation as defined in step 2 of the algorithm given above. It is incorporated as follows: if the signal currently being received from the beacon has a bearing of x, then the robot cannot be at any position in its qualitative map where the signal was perceived at a bearing other than x; and, if the robot does not currently see the infrared beacon, then it cannot be at any position in the map at which it did see the light previously and vice versa.

A typical qualitative map for the infrared light source attached to the side of loudspeaker 1 is depicted in Figure 6.21. The orientation of the arrows in the figure show the perceived bearing of the infrared beacon. The length of an arrow is proportional to the strength of the signal. The number of possible bearings, as discussed in Section 5.2.2, has been restricted to 16. The direction of bearing 0 is marked in the figure. The bearing index is incremented by 1 for every 22.5° of clockwise rotation, so that straight down in the diagram is bearing 4, straight left is bearing 8, and straight up is bearing 12.

In fact, the rotating platform on which the infrared beacon detector is mounted does in practice permit the bearings of a light source to be determined quite accurately, as described in Section 3.2.4. However, since the analysis to be performed here assumes that the robot's absolute orientation reference is in use, and since neither system available for this purpose (dead reckoning and the digital compass) is extremely precise, it seems reasonable to define direction as a 16-valued quantity.

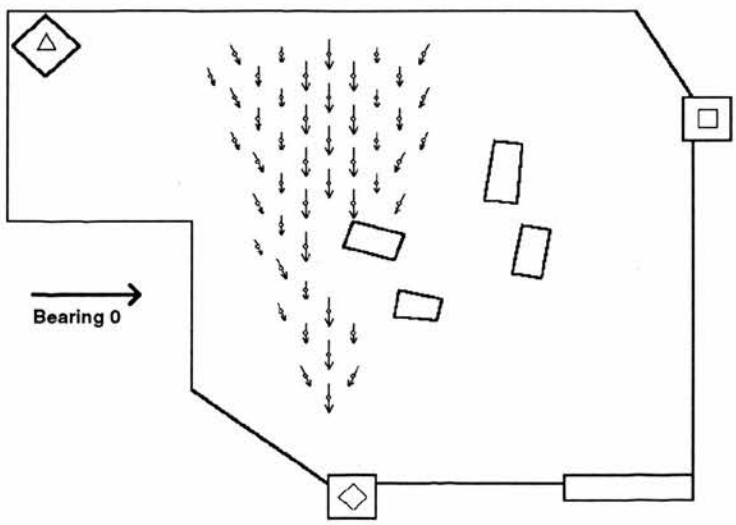

Figure 6.21. Typical visibility map of infrared beacon 1

Several observations can be made from Figure 6.21. First, it is clear that there are only four broad regions in the map: a large area where the beacon is seen to be at

bearing 4; two small areas where it appears at bearing 3 or 5; and a very large region within which it is not seen at all. Therefore, we know that if the certainty analysis is to be believed, then under no circumstances must it ever posit that the robot has more than 2 bits of information available to it about place. Second, while there is some partitioning by intensity within the larger regions, the macroscopic segmentation is a function of beacon bearings which are not affected by the looseness of the match parameter. Third, with so little information about place, it must be the case that the system will usually estimate its position to be at the centre of the pen, so that the positional errors should generally be large.

Figure 6.22 shows the results of the certainty analysis for localisation by infrared beacon 1 alone. It is consistent with the subjective observations in the previous paragraph: the system never has more than 2 bits of information available to it, and therefore can never distinguish between more than 4 places in its environment; on the other hand, the system never has much *less* than 2 bits of information on average, because of the qualitative distinguishability of beacon *directions* independently of the *intensity* of their signals.

Likewise, the results of the accuracy analysis given in Figure 6.23 are unsurprising. Independently of match strictness, the system is able to localise itself to a reasonably high degree of precision when it is within view of the infrared light source. When the match is tight, this precision increases. However, at all other places in the environment, it is not able to localise itself at all, so that its best estimate of position is the centre of the pen; under these circumstances, average positioning error, as was seen with the sonic system, is about 100 cm. Therefore, the localisation error plotted in Figure 6.23 averages this "worst case" behaviour with the precision possible in the beacon-visible zone; since the beacon-invisible area is much larger than the region in which it can be detected, this latter precision is severely diluted.

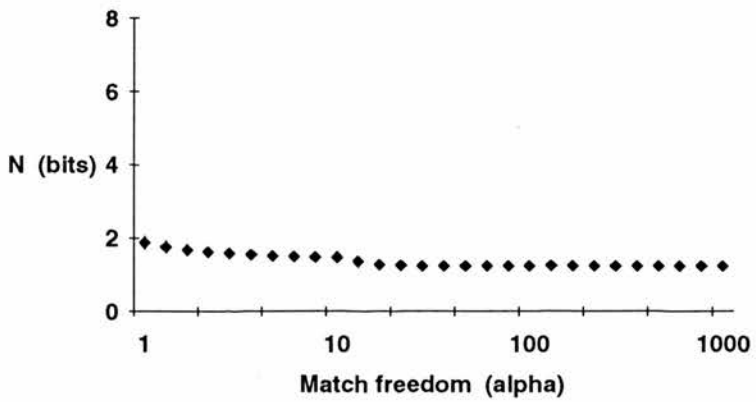

Figure 6.22. Information in the field of infrared beacon 1

Neither the conditional expectation nor the MAP distinguishes itself as a substantially better estimator than the other. For α less than 8, the MAP is on average up to 5 cm more accurate than the conditional expectation, but the overlap in their standard error bars shows this difference not to be significant.

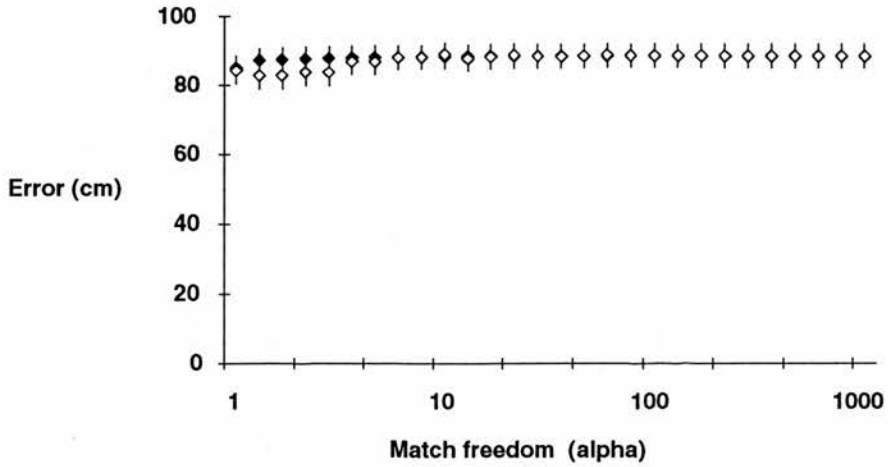

Figure 6.23. Accuracy of localisation by infrared beacon 1; ♦ marks conditional expectation as an estimator, and ◊ denotes the MAP estimator

The variance of position estimate distributions is shown in Figure 6.24. It is seen to be highly uniform, across the entire range of α. This is not entirely surprising: for those positions where the beacon is not visible, the system estimates that it could be at any of many possible places, and this large amount of variance is not a function of match strictness, as nothing is matched. However, if the robot is at any location within the beacon-visible region and if the matching is strict, we would expect that the position estimate distribution be peaked, i.e., that it have little variability.

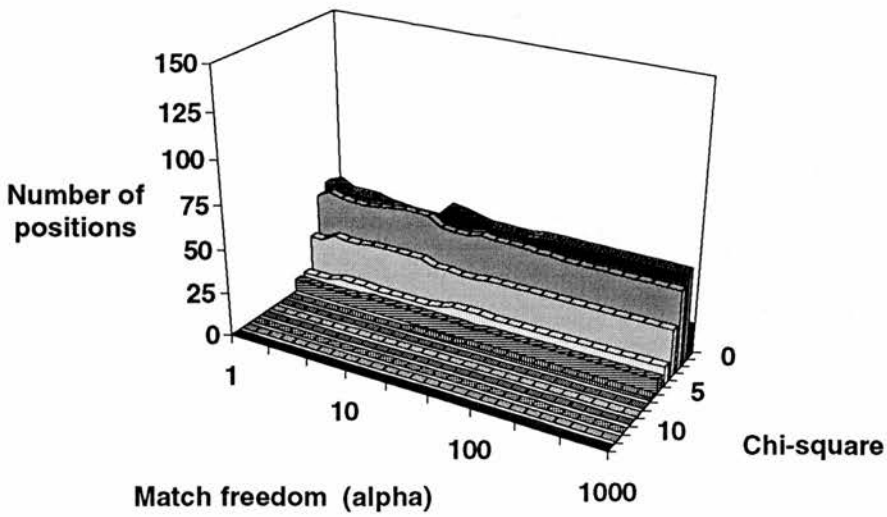

Figure 6.24. Histogram of conditional variances per position for IR beacon 1

Even if this situation occurs, as it did in the case of the sonic localisation system, the histogram for $\alpha = 1$ in Figure 6.25 shows that there is no similar trend towards a large number of positions at which the variance of estimates was smaller than the system's actual precision could support.

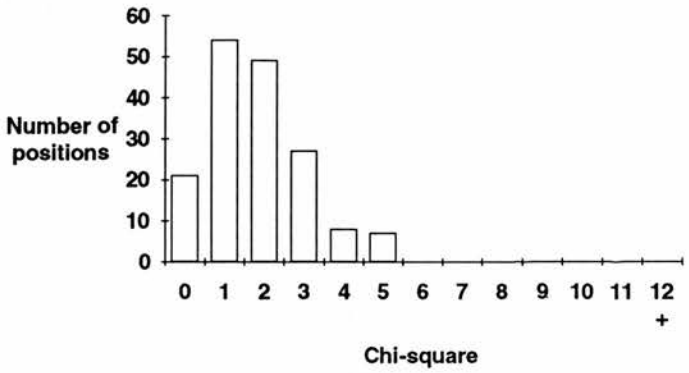

Figure 6.25. Histogram of conditional variances per position for $\alpha = 1$

In summary, then, it seems that the analytical tools applied to localisation by means of acoustic beacons also generate results for infrared beacon localisation that are consistent with subjective impressions of how well a robot using such a system might be expected to perform. Unlike the sonic system, it is found for the infrared system that the segmentation of space by beacon direction confers on it the distinct advantage that even when matching between mapped and current signal intensities is absolutely free, some information about location remains, and with it a small degree of positioning accuracy.

6.2.2 Localisation by all three infrared beacons

For the sake of brevity, the two-beacon case is bypassed in favour of progressing directly to the analysis of infrared beacon localisation when all three beacons are active. As with the sonic system, and from the discussion of composition and combination in Section 5.2.5, the analysis proceeds by computing position estimate distributions for each of the three infrared beacons independently, and combining them multiplicatively. The algorithm is omitted here as it is effectively identical to the one given in Section 6.1.4 for the sonic system.

A typical qualitative map for the three-beacon case is given in Figure 6.26. In this diagram, one set of arrows shows the bearings and intensities of each of the three IR beacon signals; no distinction has been made between the set corresponding to a

particular beacon, as this distinction is obvious from the geometry of the situation.

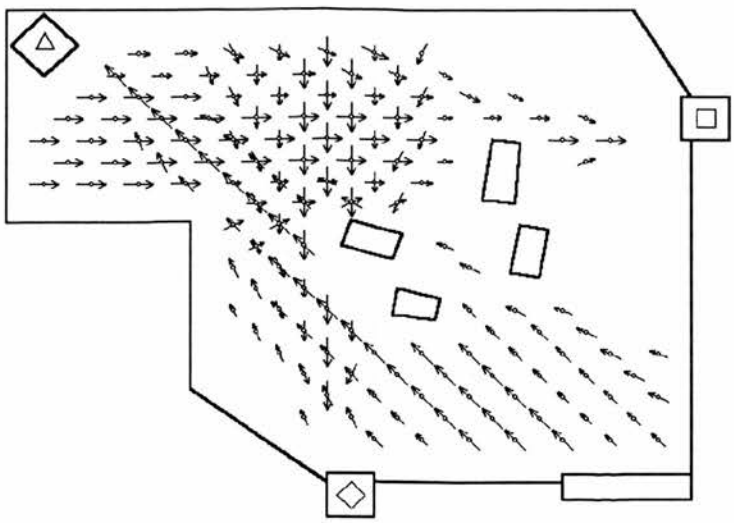

Figure 6.26. Typical visibility map of infrared beacons 1 to 3

A decomposition of this pattern by eye suggests that if only the number and orientation of arrows per position are considered, then there are about eight to ten separate regions in the room. If the length of the arrows is considered (signal intensities), then the number of regions is perhaps doubled or tripled.

Figure 6.27 shows that the certainty measure results for three-beacon localisation are entirely consistent with intuition. When the match strictness is tight (i.e., the length of the arrows in Figure 6.26 plays a part), the room is segmented into about 2^5 or 32 areas. For α greater than about 10, only the number and bearings of IR beacons are usable for localisation; but this reduced data is still able to provide the robot with at least 3 bits of information, so that it continues to be able to distinguish eight or nine regions, again consistent with the subjective segmentation of the pattern in Figure 6.26.

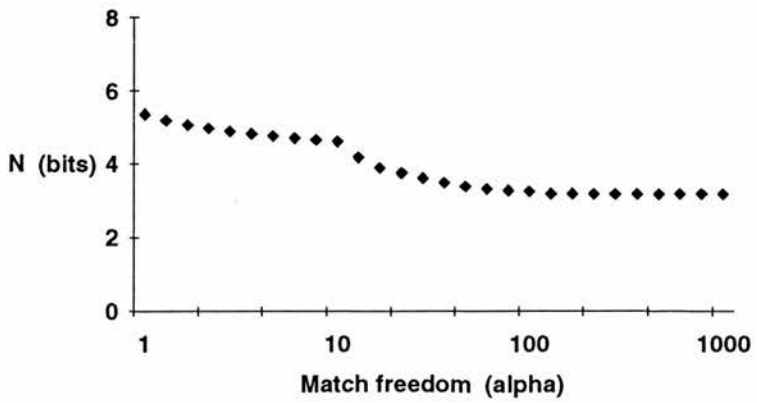

Figure 6.27. Information in the fields of infrared beacons 1 to 3

The results of computing the accuracy of the system's position estimates are given in Figure 6.28. Here it can be seen that the both the conditional expectation and MAP estimator perform about equally well for $\alpha < 50$, with both starting at about 30 cm average error, and becoming monotonically worse as the match freedom is increased.

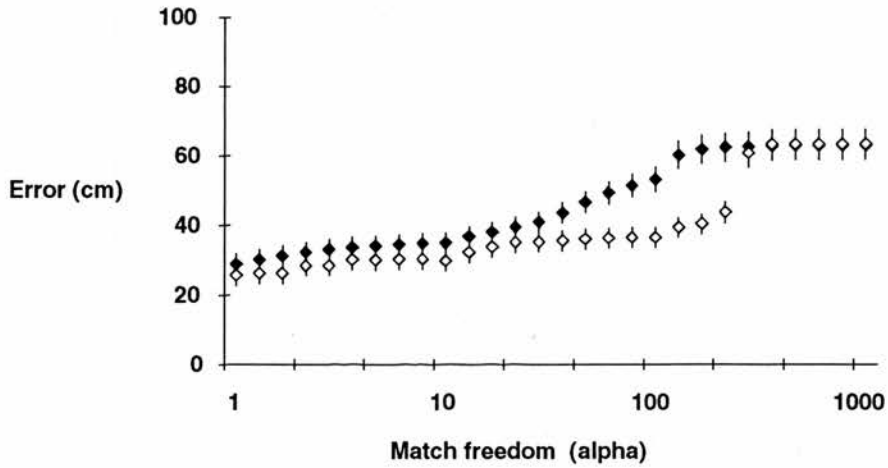

Figure 6.28. Accuracy of localisation by infrared beacons 1 to 3; ♦ marks conditional expectation as an estimator, and ◊ denotes the MAP estimator

As with the sonic system, the MAP position is a better estimator than the conditional expectation on average with α in its second decade; again, the cause of this is presumably that as new positions become likely, the conditional expectation worsens as these are included, whereas these new positions will not be more likely than the current MAP, so it remains stable. Similarly, when the match freedom is maximum, both conditional expectation and MAP can do no better than to estimate the robot's location as being at the centre of its set of likely positions. However, even in this worst case situation, the inherent segmentation afforded by seeing versus not seeing a beacon means that the magnitude of the localisation error is on average just over 60 cm, as compared with the sonic system's worst case of 100 cm.

The conditional variances, as is apparent in Figure 6.29, are not as consistent as they were for a single beacon. For example, the histogram for $\alpha = 1$ (in Figure 6.30) shows that there are now five or six positions for which the variance is unjustifiably small; as before, these are locations where the set of plausible matching positions is very small, so that the system judges its estimate variance to be smaller than is warranted by the actual reliability of these estimates. There is also the effect, in the second decade of α, that the small-variance positions suddenly become larger-variance positions (as the signal intensity differences become insignificant) even though the conditional expectation remains a reasonable position estimator. Just as occurred for

the sonic system, this phenomenon of larger-than-expected variances seems to be maximum at α about 30 or 40. On the whole, however, the variances are much more uniform than they were for localisation by three acoustic beacons.

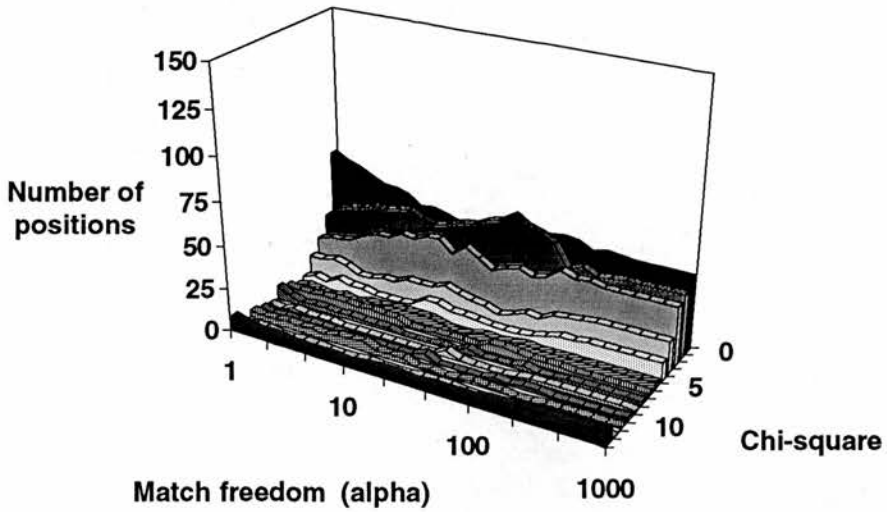

Figure 6.29. Histogram of conditional variances per position for IR beacons 1 to 3

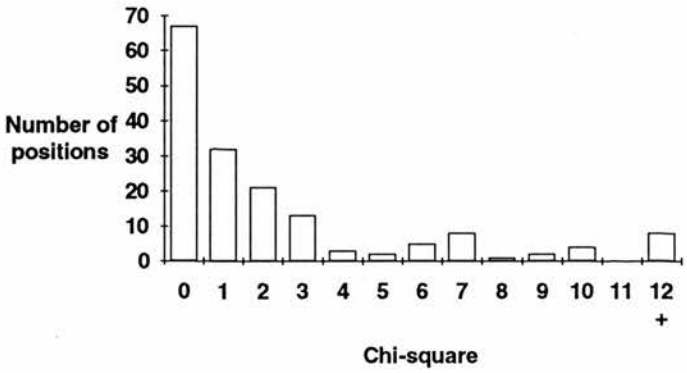

Figure 6.30. Histogram of conditional variances per position for $\alpha = 1$

Figure 6.31 plots the overlap between the the position estimate distributions for all three infrared beacons. It can be seen that the average- and best-case overlaps between estimates are fairly consistent for the three beacons. This consistent overlap is a result of the fact that each beacon has regions of invisibility, within which its position estimates will inevitably differ from estimates based on a beacon that is currently visible. Since the size of these regions is independent of match strictness, the overlap is similar for any α, although the average overlap doubles in size between strict and loose matching as signal intensity differences become insignificant.

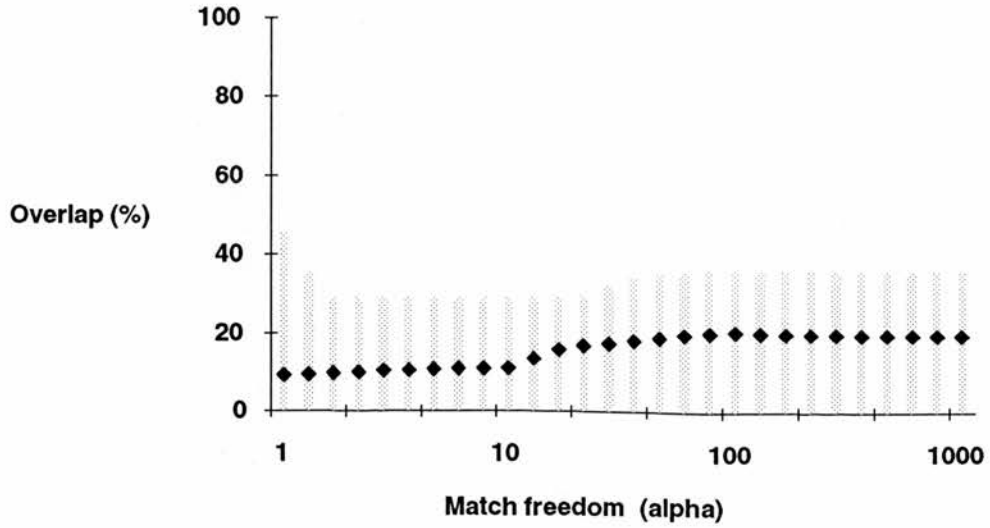

Figure 6.31. Overlap between estimates for all three IR beacons

6.2.3 Comparison of IR beacons and IR beacon combinations

As this section proceeded from the analysis for a single, particular infrared beacon directly to the combination of all three beacons, it seems appropriate to include a brief comparison of the performance of the system with each of the other two IR beacons alone, and with the various pairs beacons. In Figure 6.32 these comparative results are given for the certainty measure. It is obvious that the three individual beacons are clustered and segment space less well than the three paired arrangements, which in turn perform less well than is possible with all three beacons active.

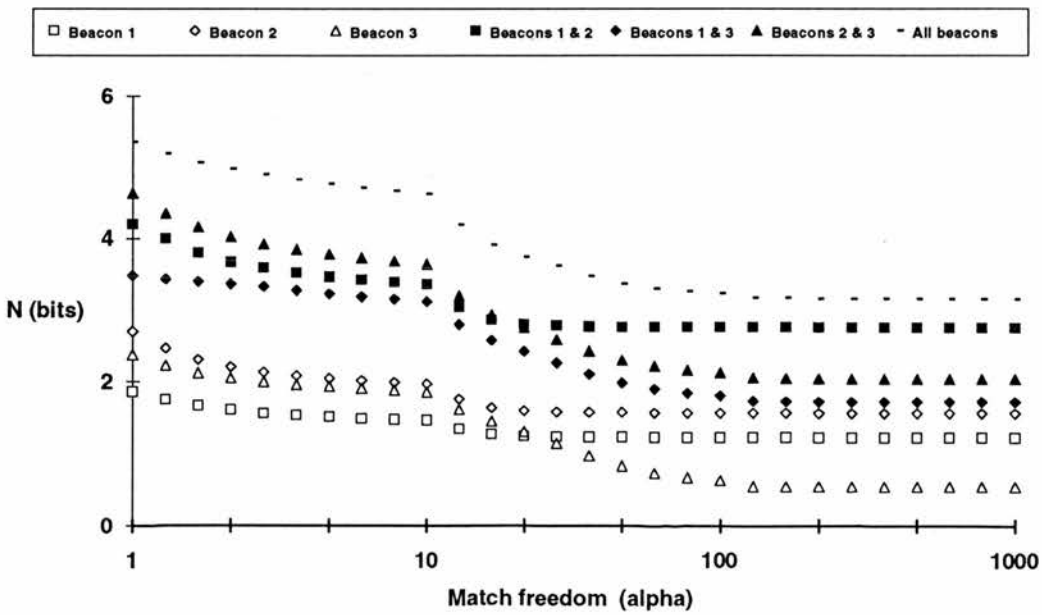

Figure 6.32. Information in the fields of all seven infrared beacon combinations

One obvious feature of Figure 6.32 is that in the worst case situation ($\alpha > 100$), beacon 3 tells the robot less about place than the other two beacons. This beacon segments space less effectively than the other two as it is visible from fewer positions in the pen; consequently, neither of the two pairs that include it is significantly better in the worst case than, say, beacon 2 alone. Likewise, the beacon 1 and 2 pair is almost as effective as all three beacons together when matching is loose.

It is not surprising that the comparative accuracy results in Figure 6.33 exhibit similar trends. That is, for loosest matching we find that the localisation error for beacon 3 approaches 100 cm, suggesting that the system using it alone can at best estimate the robot's position to be somewhere close to the centre of the pen, regardless of where it actually happens to be. Similarly, beacons 1 and 2 together are nearly as accurate for loose matches as all three beacons together, with an average error in this case of about 65 cm. For strict matching, beacon 3 is clearly much more accurate than either of the other two beacons alone. Again, this is a consequence of its small field of view: when it can be seen at all, the robot can be reasonably certain of where it is, and its estimate is also fairly precise.

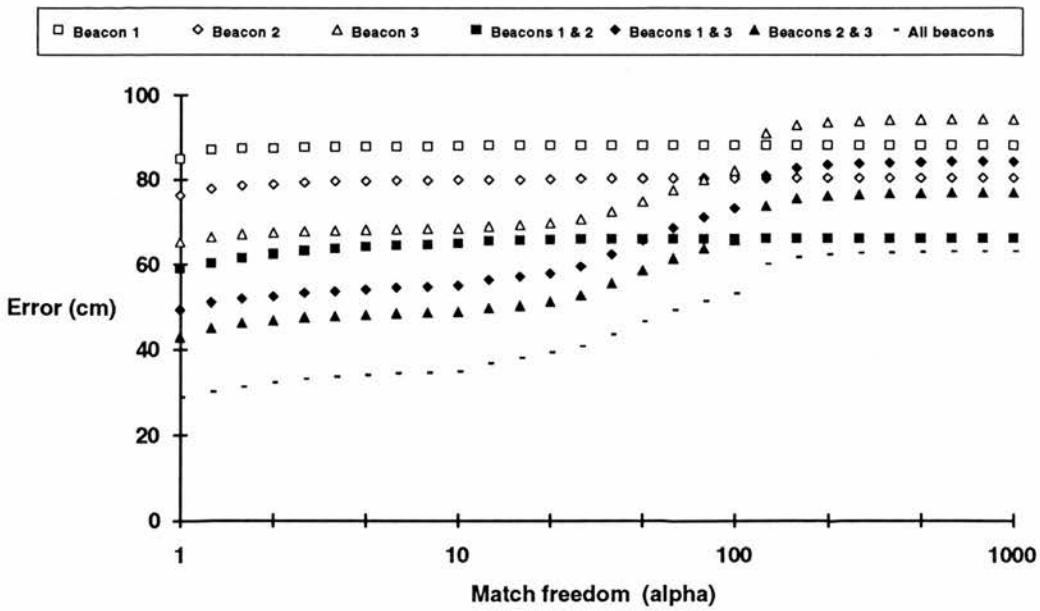

Figure 6.33. Accuracy of localisation for all seven infrared beacons combinations; ♦ marks conditional expectation as an estimator, and ◊ denotes the MAP estimator

6.2.4 Summary of infrared beacon localisation

For each of three infrared light sources, a qualitative IR map of the experimental pen

consists of pairs of numbers for each of the 166 sampling positions, with one number being the bearing of the beacon, and the second the amplitude of its signal in that direction. These maps were evaluated in this section to determine how certain a robot using them could be about where it is, and how accurate they allow its position estimates to be.

Certainty

When the match strictness parameter α is less than 10, the robot can distinguish about 8 regions in the pen with one beacon, 16 regions with two beacons, and up to 32 regions with all three beacons, which is in agreement with subjective decompositions of the bearing and intensity patterns for the various IR beacon combinations. As α is increased, the system can no longer distinguish intensities, but is still able to recognise the bearing and number of beacons it can see. Even with maximum match freedom, it is thus able to distinguish about 4 regions with one beacon, and up to 8 or 9 with two or three beacons.

Accuracy

When α is less than 10, the conditional expectation and MAP of the position likelihood distributions are comparable estimators of position, on average. For the worst single beacon, estimates are in error by 87 cm on average, for the worst pair of beacons by about 65 cm, and for all three beacons by 30 cm. As α is increased through an order of magnitude to 100, the conditional expectation worsens somewhat as an estimator, while the MAP estimator remains generally good. With α greater than 100, the conditional expectation and the MAP estimators are comparable again, and by virtue of the bearing-segmentation of space, the worst single beacon still has an average positioning error of about 90 cm, while for all three beacons the positioning error is steady at just over 60 cm.

Variance

The conditional variance of the system's position estimates is quite uniform, although it exhibits a slight tendency towards overconfidence with strict matching, and underconfidence with loose matching, just as did the sonic system (see Section 6.1.6 for further explanation).

Synopsis

The infrared beacon detector localisation system works best when the matching

between what it has seen in the past and what it is seeing presently is defined to be quite strict. This of course requires that the signal strengths of the infrared beacons remain reasonably constant over time. Under these circumstances, the system benefits in terms of certainty and accuracy from increasing the number of IR light sources, at least as they have been arranged in the experimental pen considered here. However, the system's performance with three beacons remains quite good even when signal strengths are not differentiable (maximum match freedom), because the segmentation of space afforded by seeing versus not seeing a beacon is a persistent source of localisation information. Naturally this requires that the positions of the beacons relative to the robot's operating space remain reasonably constant over time.

6.3 Ultrasonic module

Whereas with the sonic amplitude and infrared beacon systems it was relatively obvious what constituted a reasonable currency for the construction of a qualitative map, the same cannot be said of the sonar system. As described in Section 3.2.5, this module works by generating a brief 40 kHz tone, and then measuring the amplitude of this tone's reflections over time. Normally, this tone is produced for 1 ms, and is repeated every 35 ms. If the speed of sound is around 332 ms^{-1}, the transmitted pulse must be about 33.2 cm long, and in 35 ms travels 9.5 m. The system is therefore capable in principle of detecting reflections of the pulse from objects up to 4.7 m away. A typical amplitude-time plot for the module is depicted in Figure 6.34 (repeated from Figure 3.11).

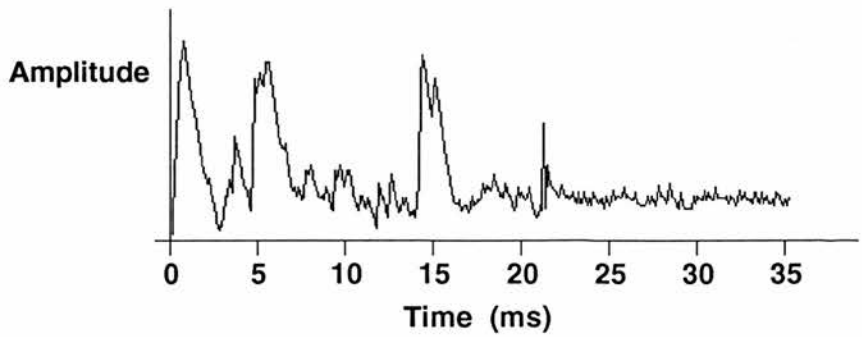

Figure 6.34. Typical amplitude-time plot for sonar module

Clearly the temporal variation in the signal in Figure 6.34 is complex. By analysing signals such as this quantitatively, it is possible to construct rich characterisations of

the robot's operating environment. However, the goal of this work has been to treat sensing systems as qualitative space-segmentation devices. Therefore, it is necessary to collapse the complexity of an amplitude-time plot such as the one above into some qualitative description of it. Various options are conceivable: for example, the number of signification peaks could be tallied, so that space could be segmented into areas of many or few ultrasonic specular reflectors.

As proposed in Section 5.2.2, the approach adopted here is to consider only the first prominent (i.e., above some chosen amplitude threshold) reflection with the sonar module facing in each of 16 directions. A qualitative ultrasonic map of the experimental pen then consists of a list of 16 numbers for each of the 166 sampling positions. The first number in the 16 is the distance to first sharp reflection of the sonar signal after its emission, with the robot facing "direction 0". The next number is the distance to the first reflection with the robot facing "direction 1" (22.5° clockwise of direction 0), and so on.

Unlike the infrared beacon detector, which can be rotated through more than 180° while the robot's orientation remains fixed, the ultrasonic module is attached to the vehicle, and so maintains a constant angle relative to it. Therefore, it is necessary for the entire robot to rotate in order to sample the ultrasonic field at different orientations. At some positions in the pen, such as near one of the walls, it is not possible for the robot to rotate through a full 360°, so that data is not necessarily available for all 16 directions at all mapping positions. This limitation, suprisingly, will be shown to confer on the system a significant locative advantage.

6.3.1 Localisation by sonar first-reflections

Suppose, as before, that data set 1 (collected on the robot's first journey around the pen) specifies the "current" data, and set 2 specifies the "map" data, where "Dist*n*(*pos*,*dir*)" denotes the distances to the first prominent reflection in direction *dir* that was perceived at position *pos* on mapping visit *n*, and "Data*n*(*pos*,*dir*)" is defined to be 1 if it was possible to collect data for direction *dir* at position *pos* and is 0 otherwise. The localisation analysis then proceeds as follows:

for each of the 166 positions A_j:
  1. let the distances to the first reflections in the 16 directions be B_i, $i = 0,\ldots,15$, such that $B_i = \text{Dist1}(A_j,i)$, and let the validity of the distance values be specified by

C_i, $i = 0,\ldots,15$, such that $C_i = \text{Data1}(A_j, i)$; and

2. as discussed in Section 5.2.3, compute

$$P(A_j | B_k, C_k) = \frac{P(B_k, C_k | A_j)}{\sum_{i=1}^{166} P(B_k, C_k | A_i)}, \qquad j = 1,\ldots,166, \; k = 0,\ldots,15,$$

where

$$P(B_k, C_k | A_i) = \begin{cases} 0 & \text{if } C_j \neq C_j(A_i), \; 0 \leq j \leq 15, \\ \prod_{j=0}^{15} e^{-\{[B_j - B_j(A_i)]/\alpha\}^2} & \text{otherwise,} \end{cases} \qquad k = 0,\ldots,15,$$

with $B_j(A_i) = \text{Dist2}(A_i, j)$ and $C_j(A_i) = \text{Data2}(A_i, j)$, $j = 0,\ldots,15$, and α a parameter that controls match strictness.

In other words, assume that the robot visits each marked position in the pen, and perceives the distances to the first prominent sonar reflections in each of the 16 directions to be the same as they were when data set 1 was collected (step 1 above), but that it does not itself know where it is; this "current" data is therefore matched against the "stored" values for each of the 166 positions collected as data set 2 (step 2); given that it is perceiving distances B_i, $i = 0,\ldots,15$, at the moment, its likelihood of being at each of the 166 positions is computed. Just as with the sonic and IR systems, we thus have 166 distributions of match likelihoods for each of the 166 positions (taken to represent the probabilities that the robot is at each possible position), which may be expressed as

$$P\{A_j | \text{Dist1}(A_i, k), \text{Data1}(A_i, k)\}, \; (k = 0,\ldots,15), \; j, i = 1,\ldots,166.$$

Since the experimental pen is about 2.5 m wide and 3 m long, the distances computed have been specified as ranging between 0 and 255 cm, with any distance actually greater than 255 cm simply truncated to this amount. The certainty and accuracy measures can thus be computed with respect to the same range of the match strictness parameter α, namely from 1 to 1000 (actually from 2.55 to 2550, as discussed in Section 6.1.1).

Figure 6.35 attempts to represent the pattern of ultrasonic first-reflection distances in the experimental pen. At each of the recording positions, a line has been plotted in each of the 16 distinguished directions. The length of the line is proportional to the distance to the first reflection. By way of further clarification, suppose that for a given position, the first reflections in all directions were large; then, the plot for that position would like example A in Figure 6.35 (notice that only 12 rather than 16 lines have been drawn, for clarity). In reality, however, the situation is usually like example B; here,

there are fairly distant reflections in the directions represented as "upper left" and "lower right" in the diagram, relatively close reflections in "upper right" directions, and either very close or no reflections at all in "lower left" directions.

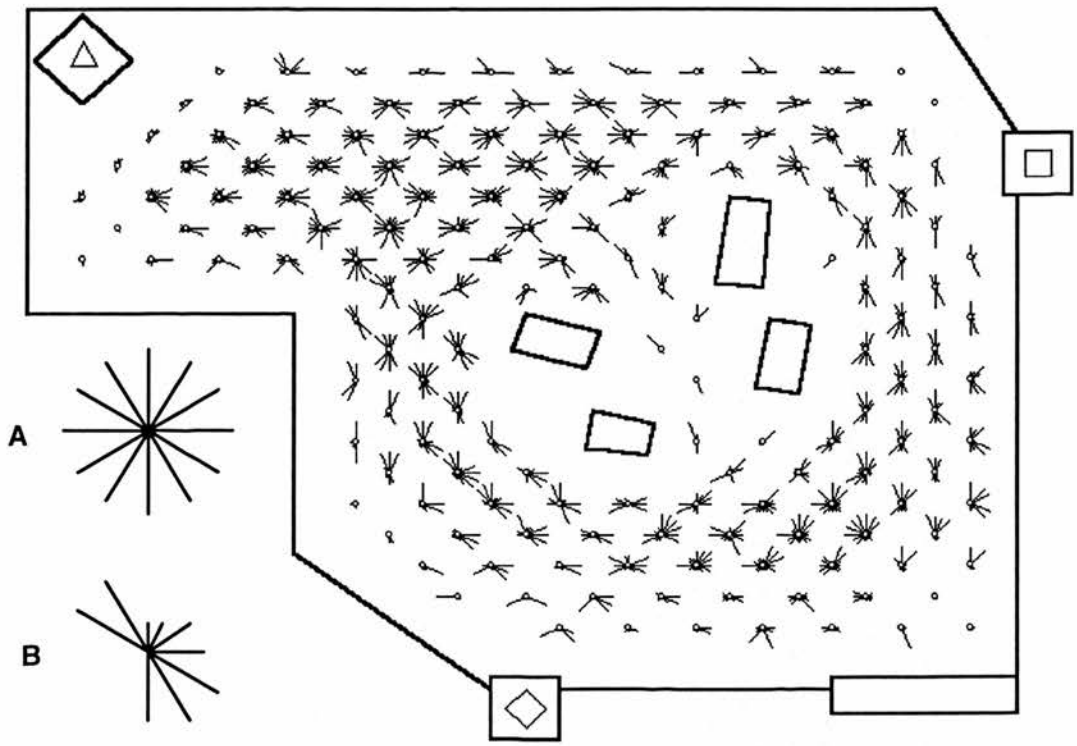

Figure 6.35. Typical sonar first-reflection pattern; a position at which there were no reflections within 255 cm would appear like the example at A (only 12 directions are shown for clarity), although most positions have patterns like B, where there are some near reflections, some far ones, and possibly no data at all in the left lower quadrant

The ultrasonic system shares a feature with the infrared beacon detector that makes them both different from the sonic system. In the case of IR system, there was an additional dimension of knowledge afforded by the simple matter of being able to see a beacon at a certain bearing, versus seeing it a different bearing or even not being able to see it all, regardless of how *well* it was seen. The amount of information in the IR beacon thus never dropped below some lower bound, whereas the sonic system lost all certainty about place when matching between current and mapped data was allowed to be loose.

In the case of the sonar system, similar conditions prevail: for some positions, it is impossible for the robot to scan in certain directions, because of the geometric contraints of the pen. Therefore, even if α is very large, so that any reflection distance matches any other reflection distance, if the robot is currently someplace where it is able to scan to its left, then it cannot be at a position in its map where it was not

possible to scan to the left, perhaps because of the presence of a wall. This is a segmentation of space that allows the system to maintain some degree of certainty about where it is, independently of the actual distances it is sensing.

Figure 6.36 is generated with the same sonar reflection data as Figure 6.35. This time, however, all distances are considered to be equal (which is certainly the case when $\alpha = 1000$). Therefore, any direction in which scanning was possible has a line of fixed length, and any direction in which scanning was not possible has no associated line.

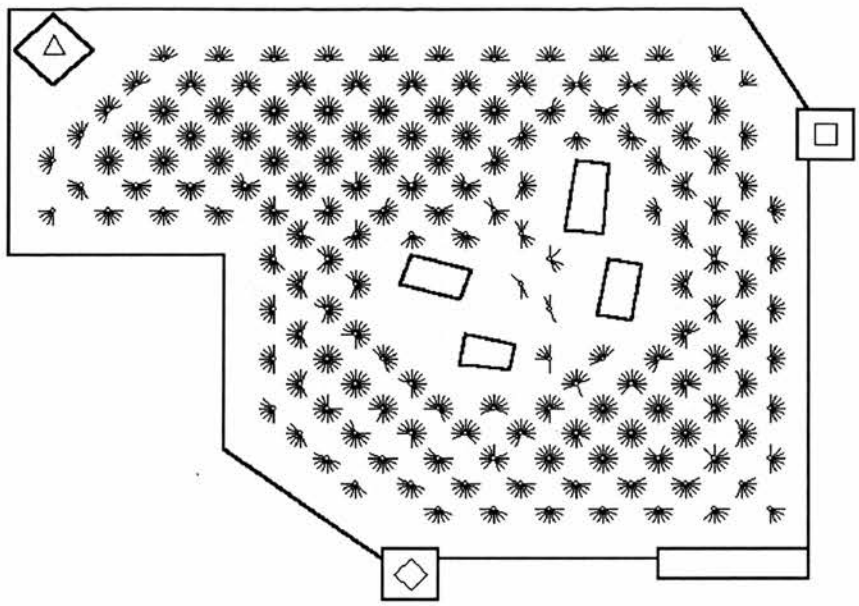

Figure 6.36. Typical sonar first-reflection pattern when all distances are set to be equal

It is clear that the number of subjectively distinguishable regions on the basis of the data versus no data condition is quite large. There are now, for example, distinctly recognisable wall zones for walls of different orientations, and also distinguishable areas that are "just a bit in" from walls. In fact, even the most uniform areas, which are those in open space where the robot can scan in all directions, are not large, so that not being able to localise any more precisely than to near the centre of one of these areas is still likely to be quite accurate localisation.

Of course, with usable distance information, even the small regions ought to be segmented more finely, so that we would expect the system on the whole to be able to tell the robot much more about place than either of the sonic or infrared systems. In Figure 6.37, the information measures for the ultrasonic system are plotted. As was to be expected, the sonar first-reflection field has 5 or 6 bits of information in it even

when the actual reflection distances are indistinguishable. A robot using this system could therefore recognise 30 to 50 different places in the pen. In fact, for a certain value of the match freedom parameter (around $\alpha = 30$), the field approaches its maximum information content of 7.38 bits, meaning that all 166 marked positions could be identified uniquely.

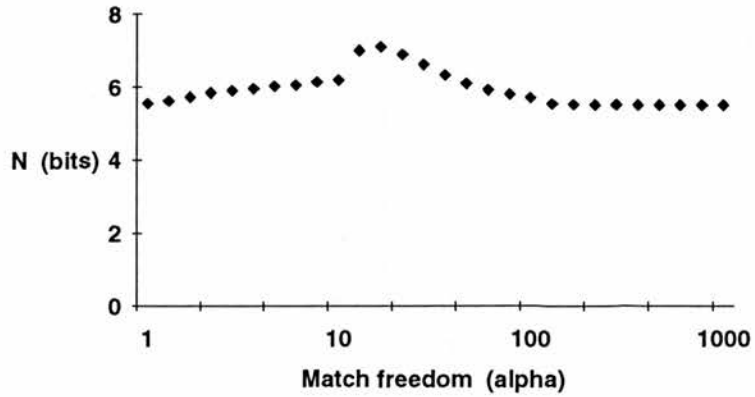

Figure 6.37. Information in the ultrasonic first-reflection field

Note that the shape of the plot is different from the shape of the plots for the other systems. This is because the other systems became decreasingly certain for increasing α, so that the curves had a descending trend throughout. Due to the greater variability from run to run in the scanned sonar data, its best performance obtains for a less-strict match; thus, the curve increases towards this value of α, and then drops off again.

The system's estimates are also quite precise, as is apparent in Figure 6.38. The error in either conditional expectation or MAP as estimators is seldom more than 40 cm. In fact, positioning error by the MAP estimator drops to about 15 cm on average with α between 10 and 100. Thus, with α around 30, the robot can be quite certain of its location (the information content of the first-reflection field is almost maximal at this match freedom), and its estimate appears also to be very accurate at this value of α.

The variability of position estimates (plotted for the full range of α in Figure 6.39) is consistent with general trends seen in the sonic and infrared beacon systems. The principal difference is the shape of the histogram at $\alpha = 1000$. Since positional accuracy remains good even when this maximum match freedom condition prevails, in most cases the actual position is quite close to the mean. However, as the variance is found to be such that the actual position lies within the $0 \leq \chi^2 < 1$ region (bin 0), this suggests that variances for the sonar system tend to be larger in general than they were

for sonic and infrared localisation. This is explained as follows: it is clear from Figure 6.36 that distinguishable "regions" tend to be small in terms of surface area, but large in extent; for instance, the many wall zones have few positions in them but are long and thin. Thus, the distance between the conditional expectation and the actual position is on average half the distance between the expectation and the furthest position from it, but if all of the positions in the zone are equally likely to be where the robot is (which is the case when $\alpha = 1000$), then the variance of a direction along the wall will be quite large.

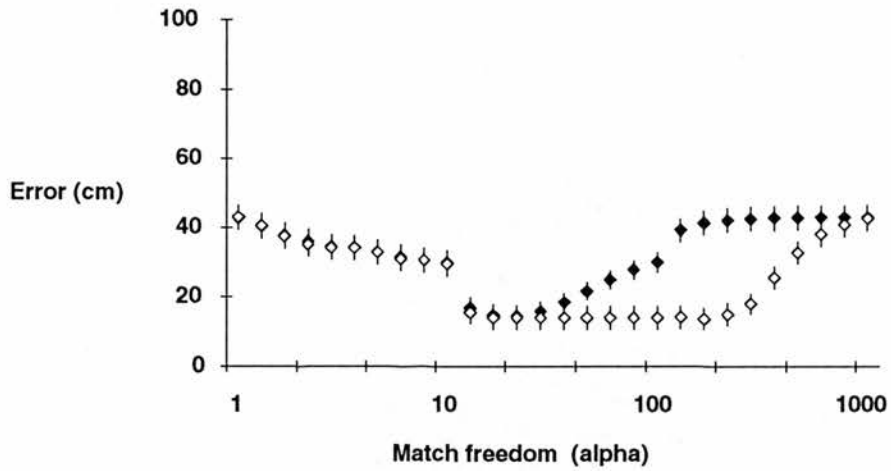

Figure 6.38. Accuracy of localisation by ultrasonic first-reflection patterns; ♦ marks conditional expectation as an estimator, and ◊ denotes the MAP estimator

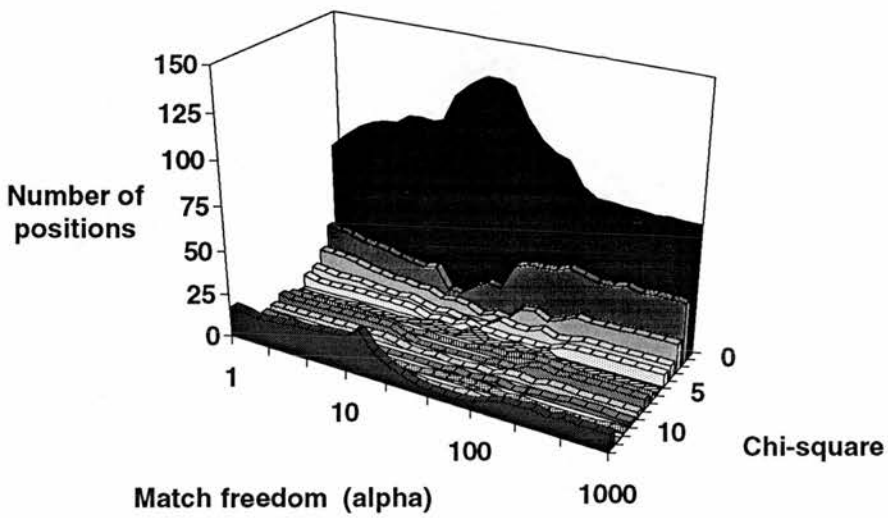

Figure 6.39. Histogram of conditional variances for ultrasonic first-reflections

For strict matching the pattern is as with the previous two systems studied, where

at some positions the estimate distribution variances are unjustifiably small (i.e., there is a significant number of positions in the 12+ bin in Figure 6.40, the $\alpha = 1$ histogram).

It is also the case that with α near 30, more positions (over two thirds of all of them) count in chi-square bin 0 (see Figure 6.41) than do for $\alpha = 1000$. In the sonic and infrared analyses, it was suggested that this indicated underconfidence (variances larger than expected, given the actual magnitude of positional errors). This explanation is consistent for the sonar module as well: it is known that the system is both maximally certain and very accurate for this match strictness, so that we would expect estimate variances to be quite small; this would engender a situation in which for some locations the actual position of the robot falls within few chi-square intervals of the expectation, and for some it falls within many. In Figure 6.41 we see that this is partly the case, since for about 20 positions (counting in the 12+ bin) the variance is obviously very small. However, most of the remaining positions fall into bin 0, suggesting for these that the variance is larger than expected.

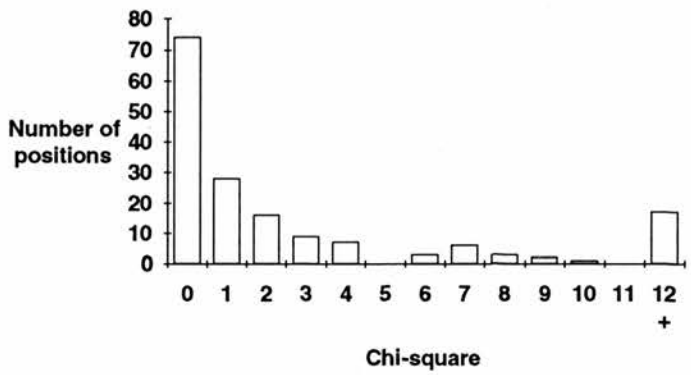

Figure 6.40. Histogram of conditional variances per position for $\alpha = 1$

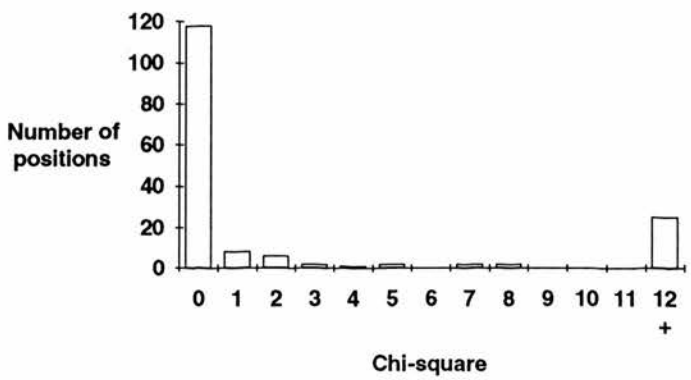

Figure 6.41. Histogram of conditional variances per position for $\alpha = 30$

6.3.2 Summary of ultrasonic localisation

A qualitative sonar map of the experimental pen consists of a list of 16 numbers for each of the 166 sampling positions, where the i-th number (i between 0 and 15) in the list is the distance to the first prominent (i.e., above some arbitrary amplitude threshold) reflection with the robot facing $22.5i°$ clockwise from some 0° reference direction. This map was evaluated here to determine how certain a robot using it could be about where it is, and how accurate such a map allows its position estimates to be.

Certainty

Regardless of match strictness, the robot can distinguish about 30 to 50 regions in the pen. For α near 30, it can distinguish nearly all 166 mapping positions uniquely. This is in agreement with subjective decompositions of typical first-reflection distance patterns. For the exponential matching function employed, an α of 30 means that the robot has about a 50% likelihood of being at any position where the sum of squared distance differences between the mapped and current data is about 4000 (so, the map could differ from the current data by about 15 cm in all 16 directions, or be identical in 15 directions but differ by 63 cm in a single direction, etc.); the robot has a 90% likelihood of being at any position where the sum of the squared differences is about 576 (so, the map data for this position might differ by 6 cm in all directions, or by 24 cm in just a single direction, etc.).

Accuracy

Both conditional expectation and MAP of the position likelihood distributions become better estimators of position as α is increased from 1 to 30. At this point, the system is most accurate, producing an average positioning error of about 15 cm. As the match freedom is increased still further, MAP remains generally as good, while the conditional expectation steadily worsens. Both estimators are minimally accurate, in error by just over 40 cm on average, by the time α reaches 1000 (any reflection distance matches any other).

Variance

The conditional variance shows a tendency towards overconfidence with strict matching, and underconfidence with loose matching, as was found for both the sonic and infrared systems (see Section 6.1.6). The shape of qualitative regions found in this system's map (long and thin) tends to lead to large estimate distribution variances.

Synopsis

Sonar first-reflection localisation works best when the matching criterion is fairly loose. This suggests that there is considerable variation in the sonar signals it is likely to receive on successive trips around the pen. As the configuration of directions in which it is even possible *a priori* to collect an ultrasonic scan is an orthogonal but persistent device for segmenting space, the system stays well informed about place even when the actual distances to prominent reflections are not differentiable (maximum match freedom). Naturally this only works when the configuration is entrained by an environmental layout that remains reasonably constant over time.

6.4 Summary of individual system results

In this chapter, three locative sensing systems were subjected to the analytic methods proposed in Chapter 5. These were the sonic signal amplitude subsystem, the infrared beacon detector, and the ultrasonic module, measuring distance-to-nearest-specularity in several directions. A data-field map of an experimental pen was constructed for each system by programming a mobile robot to travel around the pen, collecting sensor data at a large number of closely-spaced positions. The systems were then evaluated by computing the various statistical quantities of their respective data fields.

6.4.1 Comparative performance

It was found in all cases that the in-principle performance results derived were intuitively plausible, at least insofar as intuitions based on subjective examination of the data field maps were concerned. The results for the three systems are compared in Figure 6.42 (positioning certainty), and Figure 6.43 (positioning accuracy). For the sonic and infrared systems, only results for all three beacons active are given, as these indicate the best-case performance for both systems.

From Figure 6.42 it is clear that with matching fairly strict, the data fields for the three systems have about the same amount of information in them; the 5 or so bits of information available mean that a robot using one of these systems could distinguish about 30 places in the experimental pen. As the strictness between matching mapped and current data becomes loose, the certainty diverges for the three systems: the ultrasonic system is better for a short time, and then settles at 5.5 bits of information; the infrared system grows gradually less certain, stabilising at about three bits; the

sonic system's information drops ultimately to zero bits, so that no two places are distinguishable.

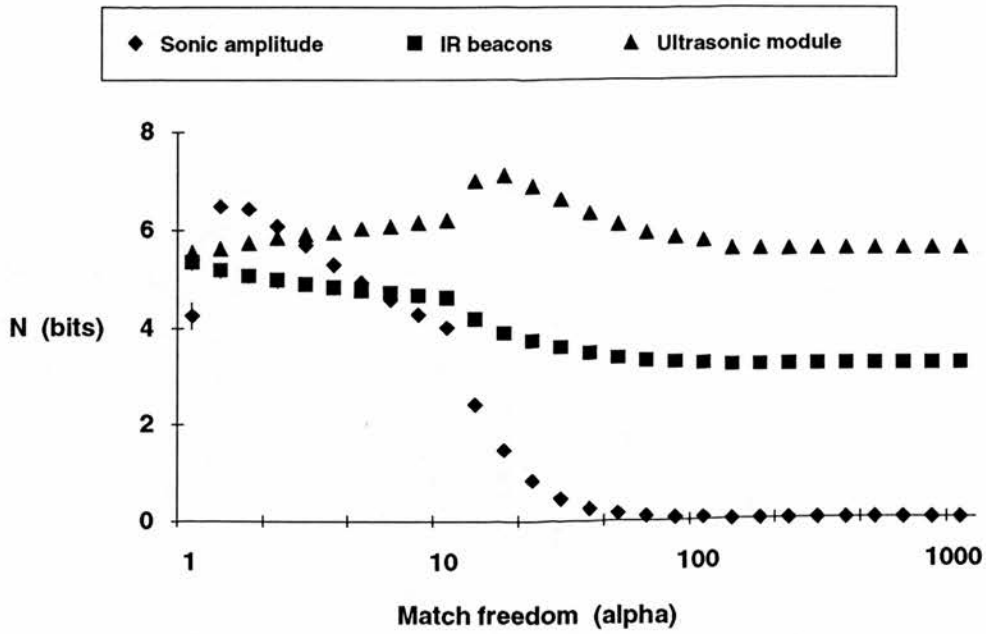

Figure 6.42. Comparative positioning certainties of all three systems studied

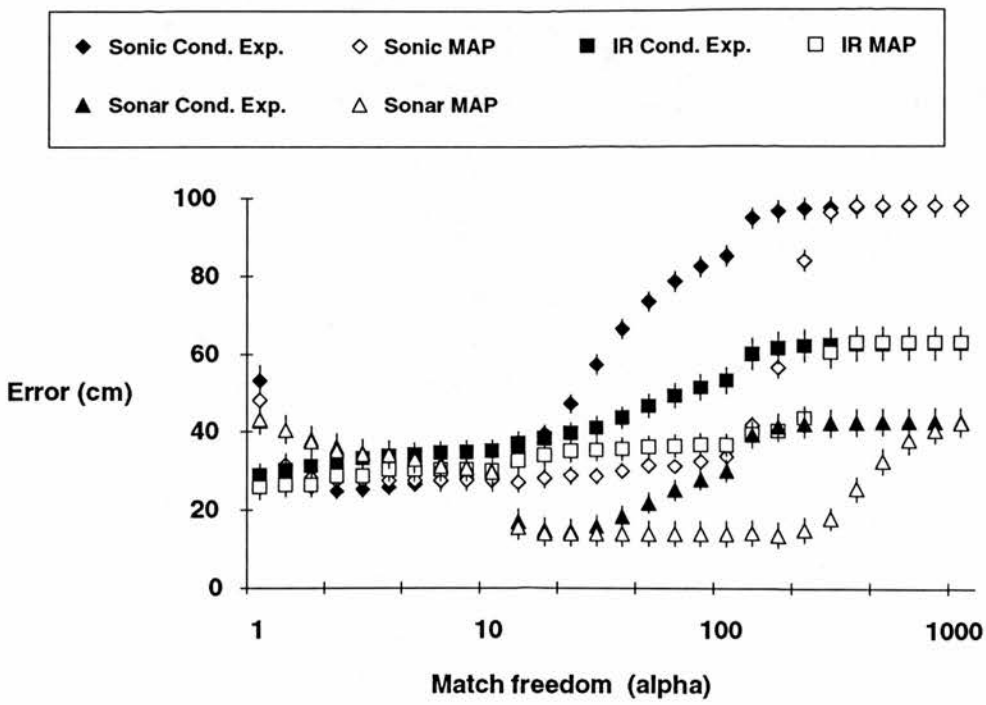

Figure 6.43. Comparative positioning accuracies of conditional expectation and MAP estimators for all three individual systems studied

The accuracies of the systems' estimates behave similarly. All three systems afford localisation to within about 30 cm of a sampling position with strict matching. With

loosened matching, the sonar system becomes briefly better, achieving the best performance of all three systems of 10 cm positioning error (at α around 30); the MAP position estimators of the infrared and sonic systems remain as good as they were for strict matching, while both conditional expectations grow steadily worse. Just as for the certainty measures, the accuracies of the three systems reach a steady state for maximum match freedom, with the ultrasonic system in error by 40 cm on average, the infrared system by about 60 cm, and the sonic system by almost 100 cm. This latter 100 cm represents a worst-case value that arises because the best the sonic system can do at $\alpha = 1000$ is to guess that the robot is somewhere in the middle of the pen, regardless of its actual location.

When the sonic and infrared systems are used with fewer than three beacons, their respective certainties and accuracies are diminished.

6.4.2 Best- and worst-case performance

As an illustration of the various results, we consider a set of 28 sampling positions that form a square within the main area of the experimental pen. These 28 positions are marked by the dashed line in Figure 6.44. The ability of each of the three systems to distinguish between these 28 positions will be assessed.

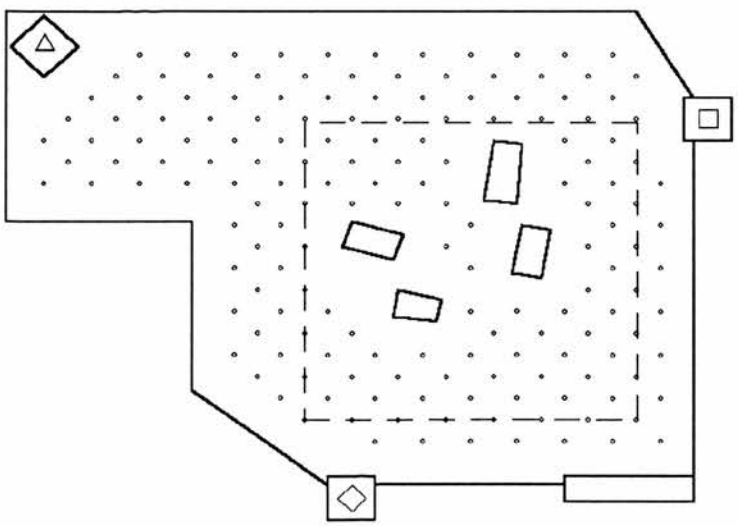

Figure 6.44. The 28 positions chosen to illustrate performance are marked by the dashed line

The "best" case results for a particular position are obtained by setting α to the value at which average positioning error for the system was minimal. The "worst" case results are obtained with $\alpha = 1000$, as this match freedom entrains the largest average position estimate error for all three systems.

Sonic System Best-Case Performance

If only beacon 1 is active, the performance for the sonic system is as in Figure 6.45. The figure shows the system's estimates of where it might be among the 28 positions, given that it is at each one of them. Perfect localisation would present as a single "wall" of likelihoods of 1, as in Figure 6.46; that is, if the robot is at (reference) position x, then it determines that it has a likelihood of 1 of being at match (map) position x, and a likelihood of 0 of being at any other map position.

It is obvious from Figure 6.45 that the sonic system does not perform particularly well for a single beacon, considering that these are best-case data. Localisation is correct at only a handful of positions. The performance of the system when both beacons 1 and 2 are active is given in Figure 6.47, and it is clear that the system's ability to localise has improved substantially. It can now localise correctly at perhaps twice as many positions as it could with just one beacon.

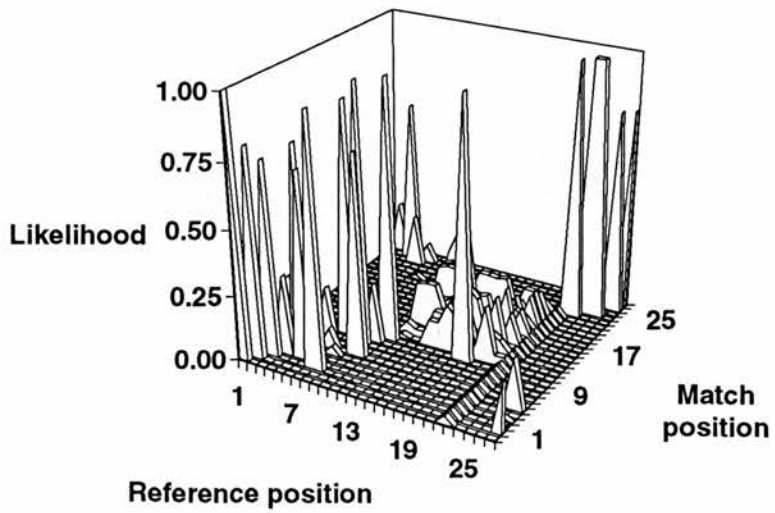

Figure 6.45. Best-case performance of sonic system with beacon 1 only ($\alpha = 1$)

Finally, in Figure 6.48 the results are given for sonic localisation with all three beacons. Again, performance has improved with the additional beacon. In fact, the system is able to localise correctly at almost all of the 28 positions considered. For several of these, the peaks are rather wide, which shows that there remains ambiguity between adjacent positions (note that positions 1 and 28 are adjacent, i.e., the set of positions "wraps around", so the fact that position 1 is confused with position 28 is not entirely unreasonable).

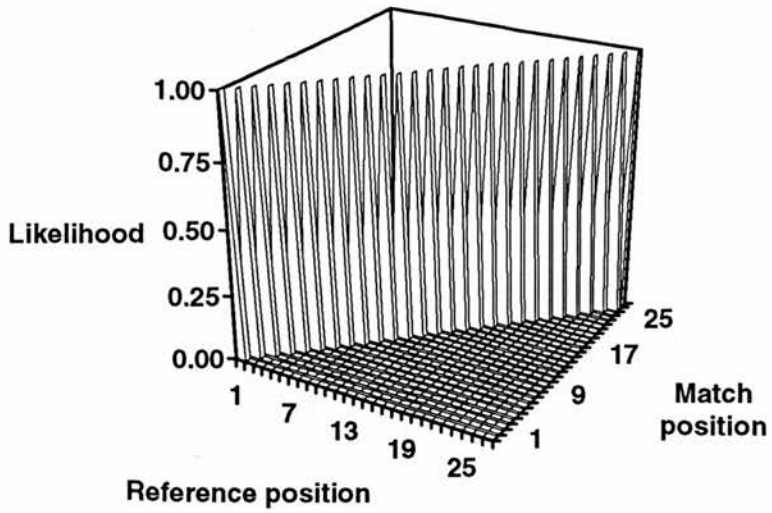

Figure 6.46. Optimal performance; each position is matched correctly

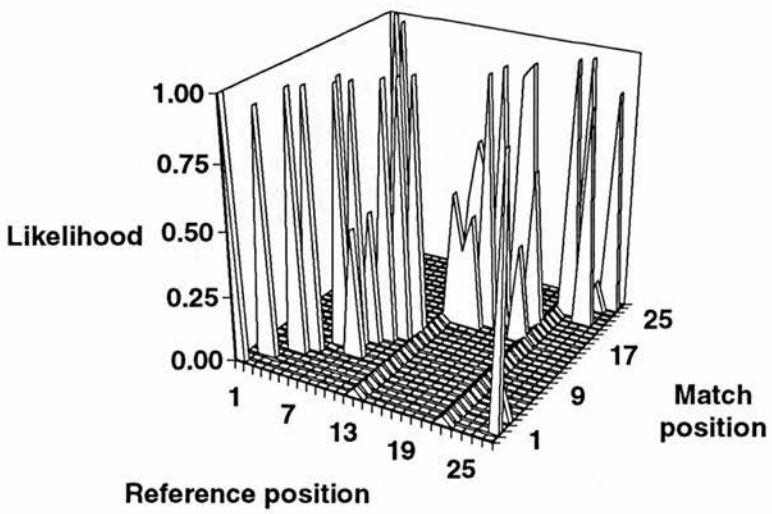

Figure 6.47. Best-case performance of sonic system with beacons 1 and 2 ($\alpha = 2$)

Infrared System Best-Case Performance

If only infrared beacon1 is active, the performance of the system is as in Figure 6.49. The various spatial domains associated with the single IR beacon are quite visible. As beacon 1 is detectable at a small subset of the 28 positions, the system is able to localise correctly only in these locations. If the beacon is not visible, then the system has an equal likelihood of being at any of the positions where this is the case. The large flat regions in Figure 6.49 correspond to this situation.

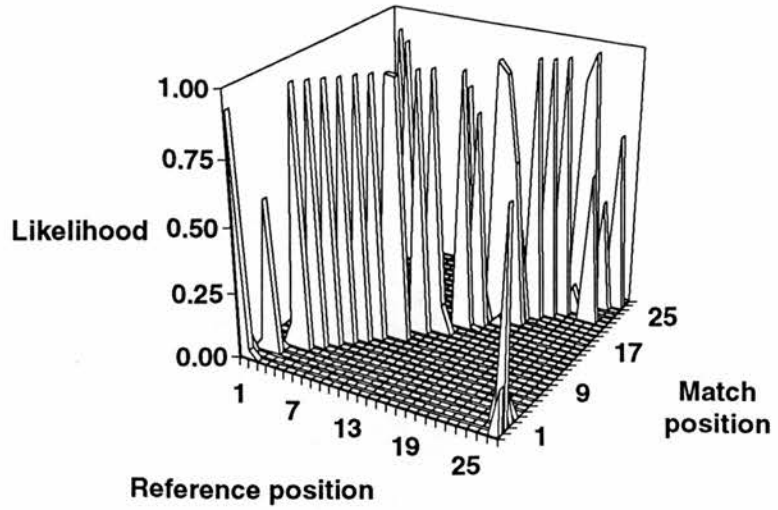

Figure 6.48. Best-case performance of sonic system with all three beacons ($\alpha = 4$)

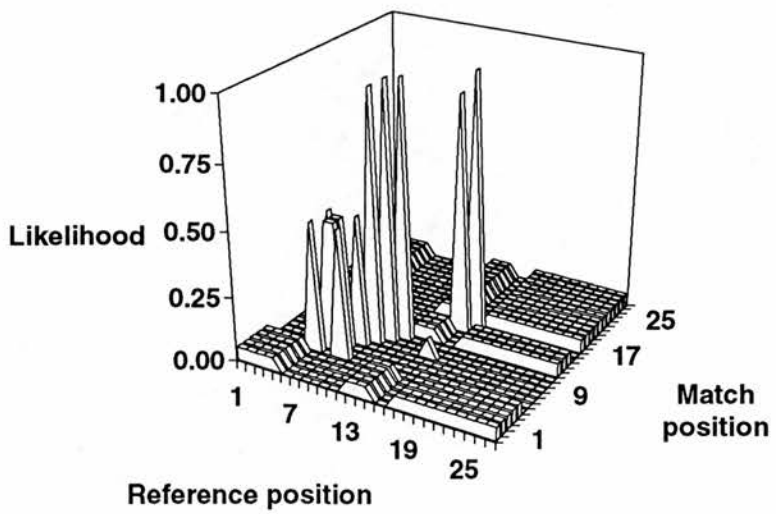

Figure 6.49. Best-case performance of infrared system with beacon 1 only ($\alpha = 1$)

On the other hand, when all three beacons are available, performance is as in Figure 6.50. As the combinations of visibility-invisibility of the three IR sources at the 28 sample positions are reasonably varied, the system is able to localise correctly at nearly all of them. In situations where it is uncertain (such as at positions 18 to 20 or in the group near position 7), its confusion is usually local (i.e., it only confuses adjacent positions); the exception to this is the group of positions 24 to 28, 1, and 13, which are apparently indistinguishable, possibly because none of the IR beacons are visible from these positions.

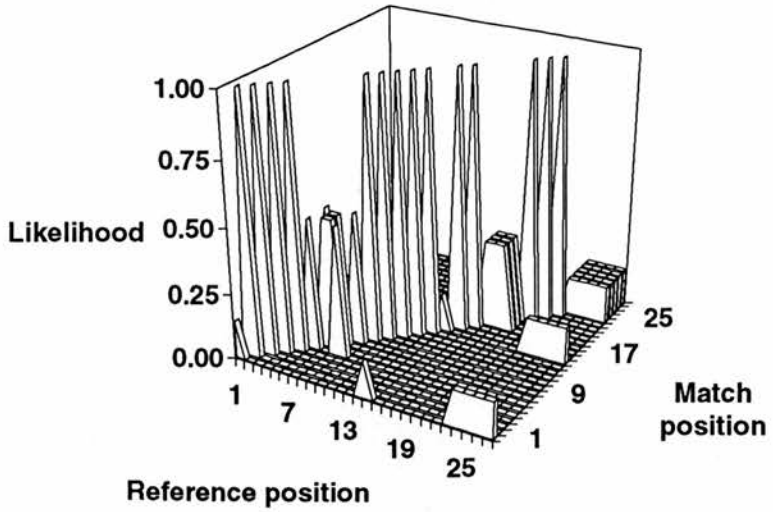

Figure 6.50. Best-case performance of infrared system with all three beacons ($\alpha = 1$)

Ultrasonic System Best-Case Performance

The best case for the ultrasonic system is given in Figure 6.51. Given that this system has nearly maximum information (maximum distinguishability of positions), it is not surprising that its best-case performance is almost perfect. There are some slight ambiguities between adjacent positions (such as positions 13 and 22), but localisation is correct everywhere else.

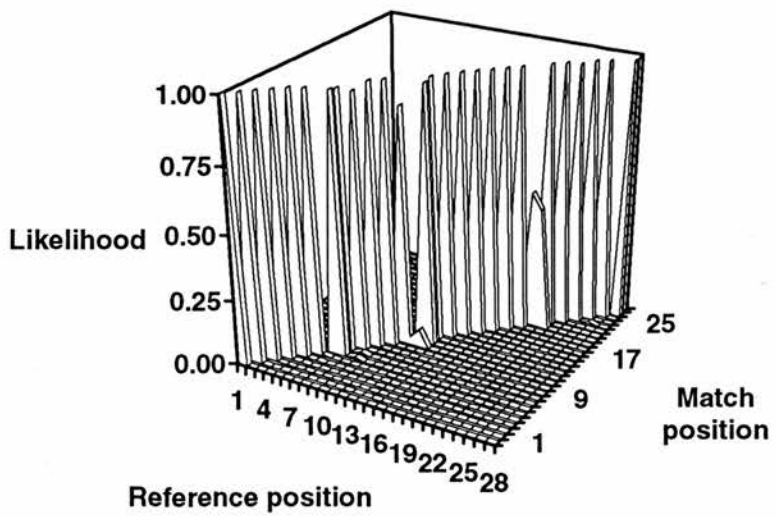

Figure 6.51. Best-case performance for sonar system ($\alpha = 30$)

When the match control parameter α is set to 1000, the sonic system is unable to distinguish between intensities, and therefore has no basis for distinguishing between positions. The result is that at worst, it performs very poorly. In Figure 6.52, it can be seen that the position likelihood distribution is essentially flat as a consequence.

Despite loose matching between map and current data, the infrared system retains a degree of locative capability just by virtue of the inherent segmentation of space that comes from being able to see a beacon in some positions but not in others.

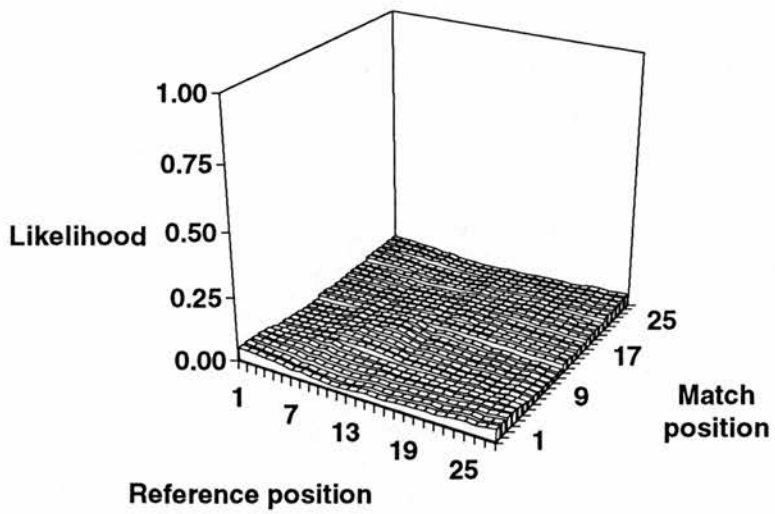

Figure 6.52. Worst-case performance for sonic system with all three beacons

Therefore, the infrared system continues to be able to achieve a measure of spatial localisation, as can be seen in Figure 6.53. There are several positions at which localisation is perfect, presumably where particular unique combinations of beacon bearings obtain. There are also areas in which the localisation ambiguity occurs only between adjacent positions, suggesting that one or more beacons are visible in these positions, but that the beacons look the same from all of them. As before, the system confuses positions 24 to 28, 1, and 13, because none of the beacons are visible at the locations in this group.

As expected, the even greater segmentation of space inherent in the sonar system, a consequence of the unavailability of ultrasonic data from some directions at some vantage points, means that the system continues to offer quite good localisation even in the worst case. In Figure 6.54 it is clear that there remain several positions at which

localistion is still perfect, and for virtually all remaining positions, the only ambiguity in location estimation is between adjacent positions. The only exceptions to this are position 2, which is apparently slightly confused with 9 and 10, and positions 6 to 10, which obviously share indistinguishable features with positions 21 to 23.

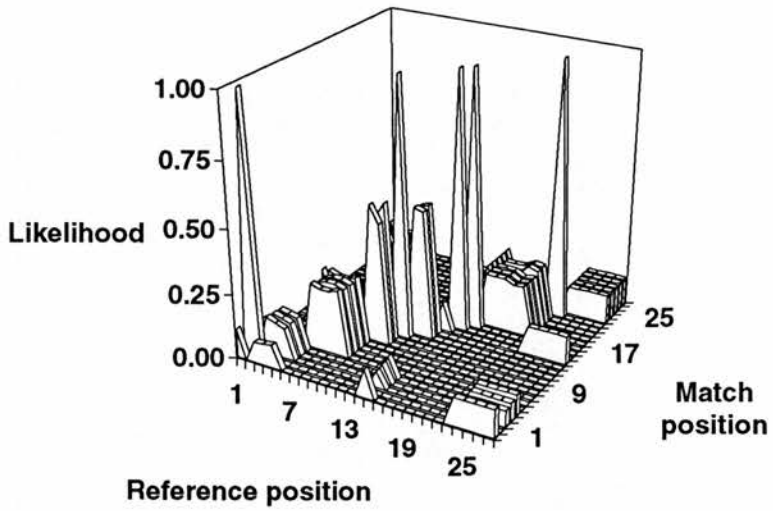

Figure 6.53. Worst-case performance for infrared system with all three beacons

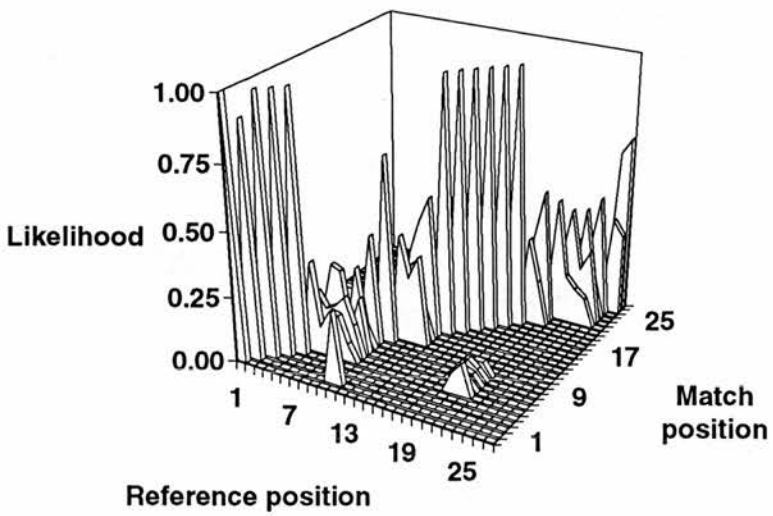

Figure 6.54. Worst-case performance for ultrasonic system

6.4.3 Conclusions

Each of the three systems considered works best under particular conditions.

Localisation by means of the amplitudes of tones from loudspeaker "beacons" is most accurate, and most certain, when the matching between stored data from past visits and the data currently being received is quite strict. This is also true of localisation by the bearing and intensity of infrared light sources. It follows that the data fields for these two systems were stable, at least during the two journeys the robot made around the environment which formed the basis for the "map" and "current" signal comparisons. For both systems, it was found that the greatest accuracy obtains when all three beacons (loudspeakers or IR light sources) were active. On the other hand, localisation by means of the pattern of distances to the closest ultrasonic specular reflector in a number of directions works best with somewhat less-strict matching, suggesting that the ultrasonic data was more variable between visits.

The certainty and accuracy of each system's position estimate was computed with the robot placed at each of 166 positions in an experimental pen. These 166 certainties and accuracies were averaged as a means of summarising a system's performance. The fact of small standard errors in these averages show them to be usefully representative of a system's typical performance for a given match freedom.

Certainty

All three systems are able to segment the environment into a similar number of domains for strict matching. However, as the match freedom increases, the sonic system loses its only means of segmentation (namely by intensity differences, which are diminished with decreased match strictness), and therefore eventually cannot distinguish any two places in the pen. The infrared and ultrasonic systems also segment space less well as the match freedom increases, but as there are other dimensions to their abilities to segment space, both reach a steady state certainty that is greater than zero (the infrared beacons have zones of visibility and invisibility, which persist regardless of intermediate degrees of visibility; the sonar system also has data/no-data conditions dictated by the geometry of the environment, as it is impossible for the robot to obtain sonar scans in some directions at some positions). The inherent segmentation in the ultrasonic system is the greater of the two, so that this system is always able to distinguish at least 30 to 50 places in the pen.

Accuracy

In general, it was found that the three systems were about equally accurate for strict matching, in error by 30 cm on average. As the match freedom is increased, the ultrasonic system enters its region of optimal performance, achieving the best average

error of all three systems, about 10 cm. With maximum match freedom, the sonic system loses all localisation ability, and constantly estimates the robot's position to be in the middle of its operating space; its positioning error is therefore the average of the 166 distances between the sampling points and their centroid, which is close to 100 cm. The inherent segmentation of space afforded to the infrared and ultrasonic systems means that their estimate errors in these extreme conditions are smaller, namely 60 cm and 40 cm on average, respectively.

Variance

For all three systems, two trends emerge in the behaviour of the systems' estimate variabilities across the range of the match strictness parameter. First, when matching is strict, there tend to be several positions at which estimate variances are smaller than is warranted by the magnitude of the actual positioning errors. This is an artefact of the instability of strict matching where, for example, a system might believe that there are a few positions where the robot could be, so that the estimate distribution is highly peaked (has little conditional variance). Second, the overconfidence with strict matching contrasts with a trend towards underconfidence for less-strict matching. This effect occurs because, on the one hand, the increased match freedom means that new positions become possible locations for the robot, so that the spread of the estimate distributions (variance) increases as well, and on the other hand, the conditional expectation remains a reasonably good estimator of position; the disproportionate increase of conditional variance versus conditional expectation shows the systems' to be underconfident in these circumstances.

Summary

To summarise, then, it is clear that the ultrasonic localisation method is the most certain and accurate of the three systems considered in this chapter, and that this is primarily the result of the inherent segmentation of space afforded to it by the way that the constituents of the environment in which it is being used create persistently recognisable data/no-data zones. However, localising by specularity patterns relies upon a relative fixity of the geometry of the space (i.e., assumes that layout does not change much between visits). The sonic and infrared systems do not perform as well, but assume only that their reference beacons remain relatively fixed, which is almost certainly a less demanding condition of use; in fact, the infrared system also enjoys some partitioning of space that comes from the visibility-invisibility pattern of the infrared beacons, although to a lesser degree than the sonar system.

Therefore, it is necessary to consider the apparent trade-off between accuracy and assumptions about environmental stability, and this matter will be addressed in Chapter 8. First, though, we discuss localisation by combinations of the systems treated separately here, as the results of this analysis will be found to bear on the environmental-constancy issue.

Chapter 7

Localisation Analysis for Combined Systems

In Chapter 6, sensory fields of the sonic signal amplitude subsystem, the infrared beacon detector, and the ultrasonic module were analysed in order to determine how well a robot using any one of these devices might be able to localise itself. It was found that each of the systems works best under particular conditions. The pattern of distances to the first ultrasonic signal specular reflection in several directions affords the most certain and accurate localisation of the three systems. The sonar system is able to estimate the position to within 10 cm of one of 166 "marked" positions in an experimental pen of 10 m^2 surface area, approximately three times better than the best performance of the sonic and infrared systems.

However, the pattern of ultrasonic reflections is a function of the geometry of surfaces in the environment in which the robot is operating. While surfaces such as walls can usually be relied upon not to change as the vehicle moves within its domain, spaces such as offices and laboratories are likely to gain new objects, or at least are susceptible to having existing objects moved around. Any such change in the number or orientation of surfaces in the environment is bound to affect the performance of a system that depends on consistency of spatial layout through time.

The sonic and infrared systems rely upon "beacon" reference signals (a tone emanating from a loudspeaker, or the emissions of an IR light source, respectively). Devices such as these can reasonably be expected to remained fixed in the environment. A fuller discussion of environmental stability and its implications for localisation by the various systems follows in Chapter 8; for the moment, it does not seem unreasonable at least to suggest that combining sensory devices might allow the accuracy of a layout-sensitive one to benefit from the resistance to change of another (even if the latter is substantially less accurate than the former). Furthermore, combining systems is likely to increase the robustness of localisation in the face of perturbations like equipment failure.

This chapter has as its theme the assessment of the certainty and accuracy of localisation by means of combinations of the three systems treated individually in

Chapter 6. Each of the four possible combinations (three pairs and one threesome) will be evaluated by calculating the positioning certainty and accuracy measures for combined distributions, as proposed in Chapter 5 and already performed for the multi-beacon sonic and infrared systems in Chapter 6. The chapter concludes with comparisons of the results for the various combinations.

7.1 Sonic and infrared systems

As discussed in detail in Section 6.1, the qualitative map of the sonic system consists of the pattern of a beacon's intensities measured at each of the 166 sample positions in the experimental pen. The corresponding infrared map (see Section 6.2) comprises pairs of numbers for each position, where the first number in the pair is the bearing of an IR light source relative to the position, and the second its intensity. The pen is equipped with three each sonic and IR beacons, and in the relevant individual system analyses, it was shown that the both localisation systems worked best with all three of their respective beacons active. Therefore, for the sake of brevity, it is fair to compare the sonic and infrared systems assuming that each has all three beacons available.

In fact, it was in combining the sonic maps for individual beacons that the procedures for combinational analysis were first introduced. With the robot at some arbitrary location, the intensity of each beacon's signal was the basis for constructing a separate distribution of likelihoods of being at each of the marked positions in the pen. For the exponential matching operation proposed in Section 5.2.3, it was shown to be sufficient to multiply and renormalise the individual probability distributions in order to construct position estimates combining the influences of the individual beacons.

The same approach is adopted here for analysing the combined performance of the sonic and infrared systems. In brief, the procedure runs as follows: the robot is assumed to be placed at some position i; from the intensities of the three acoustic beacons at this position (specified by the intensities that were collected as data set 1, on the robot's first sampling journey around the pen) and the bearings and intensities of the three infrared light emitters, and given that the systems are aware of the intensities and bearings that prevailed everywhere in the pen on previous visits (specified by data set 2), six individual distributions obtain, each describing the likelihoods of the robot being at the 166 sampling locations in the pen; these six distributions are multiplied together and renormalised to produce a single combined

distribution (in practice, we need only take the three-beacon sonic and IR distributions computed in Sections 6.1.4 and 6.2.2 respectively, and multiply these).

The performance of the sonic and infrared systems is then computed from this combined distribution, as before. For example, the conditional expectation and maximum *a posteriori* probability (MAP) position can be determined, and the distance between these two and the robot's actual position show how accurate localisation is with the two systems at once. As in Chapter 6, a combined position likelihood distribution can be computed wherever the robot is placed; by stepping through the 166 sampling positions, assuming the robot to be at each one in turn, and computing the various statistical measures at each step, it is possible to establish how the combined systems perform throughout the operating space.

Certainty

Figure 7.1 summarises the amount of information available to the robot given both sonic and infrared systems, with three active beacons apiece. The results in the figure were computed as follows: for a given value of the match strictness parameter α, the robot was assumed to be placed at each of the 166 marked positions, a combined position likelihood distribution was constructed, and the entropies in these 166 distributions were averaged and plotted. One-standard-error bars are shown, although these are small.

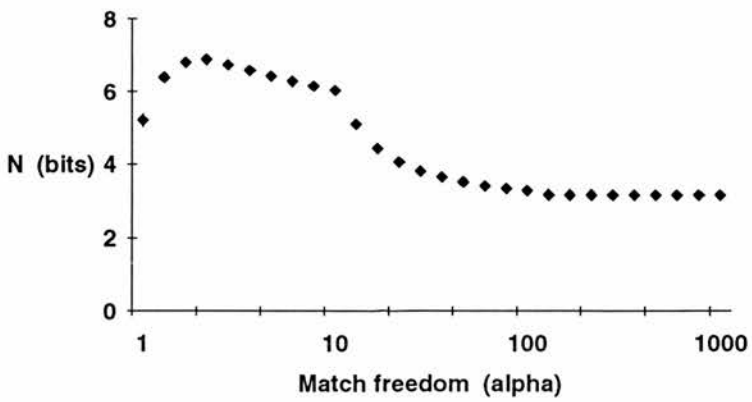

Figure 7.1. Amount of information in the combined sonic and IR fields with three beacons each

As expected from the definition of the entropy measure, if the curve in Figure 7.1 is compared with the information plot for the sonic system with three beacons (Figure 6.14) and the infrared system with three beacons (Figure 6.27), it is found that the amount of information in the combined distribution is never less than the information in

any of the individual distributions. It is clear that under optimal matching conditions given the "current" and "map" data sets, namely for $\alpha = 4$, the systems in combination have maximal information about "place" (about 7.38 bits). The vehicle is thus able under these circumstances to distinguish all 166 sampling positions uniquely.

Furthermore, the inherent segmentation of space afforded to the infrared beacon detector (by virtue of the fact that being able to see or not to see one of the reference light sources is independent of exactly how *well* it can be seen), means that even with maximum match freedom the combined systems have about 3 bits of information about place. A robot using them can effectively always distinguish at least eight regions in its operating space.

Accuracy

Just as the amount of information in the combined system is always at least as large as the information in the individual distributions, it can be seen in Figure 7.2 that the accuracy of localisation is always at least as good as the most accurate of the individual systems. The combined systems are most accurate for α near 5, when the error in both the conditional expectation and MAP position as estimators is less than 10 cm on average.

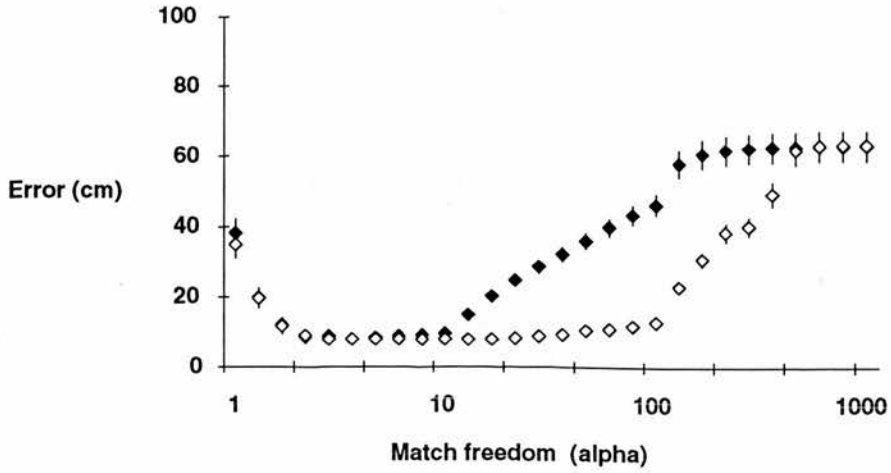

Figure 7.2. Accuracy of combined sonic and infrared systems

As for the individual sonic and infrared systems, it is clear from Figure 7.2 that the conditional expectation and MAP are both about equally good for tight matching, that the MAP estimator is better than conditional expectation for α in its second decade, and that as matching becomes maximally free both estimators are ultimately equally poor. It is not surprising that the positioning error for the pair of systems is not greater than 60 cm on average, even with $\alpha = 1000$. This is the error associated with the

infrared system on its own: for maximum match freedom, the sonic system estimates the position of the robot to be anywhere with equal likelihood; thus, the combined estimates degenerate to the estimates for the IR system alone, and its maximum match freedom performance prevails.

Variance

The variances of the combined distributions, for three decades of the match freedom parameter α, are summarised in Figure 7.3. Given the variances of the position estimates for the individual systems, it is not surprising to find similar trends in the combined data. Figure 7.4 shows the contributing separate histograms for $\alpha = 1000$ and $\alpha = 1$, on the left and right respectively. As before, the $\alpha = 1000$ histogram is a comparison baseline. Since the sonic system contributes nothing under the conditions of maximum match freedom, the histogram here is identical to that of the infrared system alone (just as was the case for the entropy and accuracy measures).

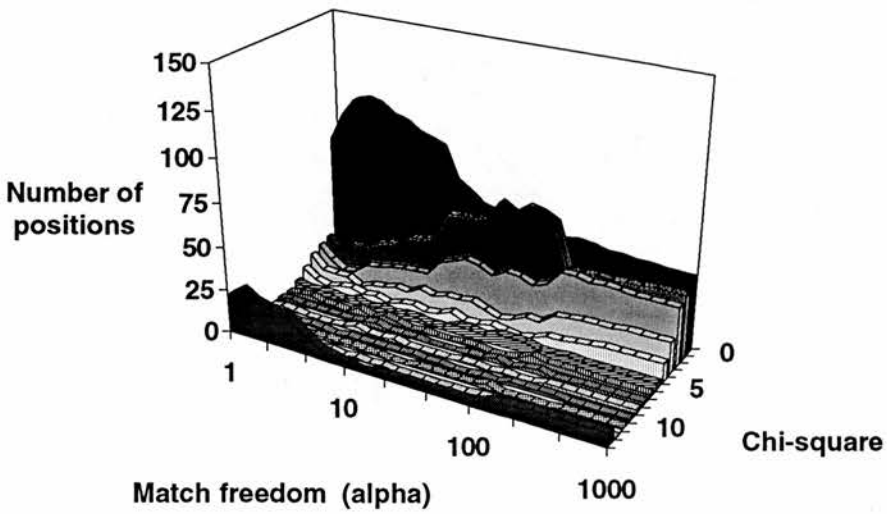

Figure 7.3. Conditional variances of the combined sonic and infrared distributions

Compared with the $\alpha = 1000$ baseline (Figure 7.4, left), we see similar trends to those witnessed earlier for the three individual systems. For very strict matching, the variances incline towards being smaller than expected, given the actual magnitude of the combined systems' estimate errors; as for the individual systems, the set of plausible matching positions is very small for strict matching, so that the combined estimate variances are judged to be smaller than is warranted by the actual reliability of these estimates. Also, there is the tendency in the second decade of α for these small-variance positions suddenly to become larger-variance positions (as the signal intensity

differences become insignificant) even though the conditional expectation remains a good position estimator; in other words, the variances are larger than expected for these positions.

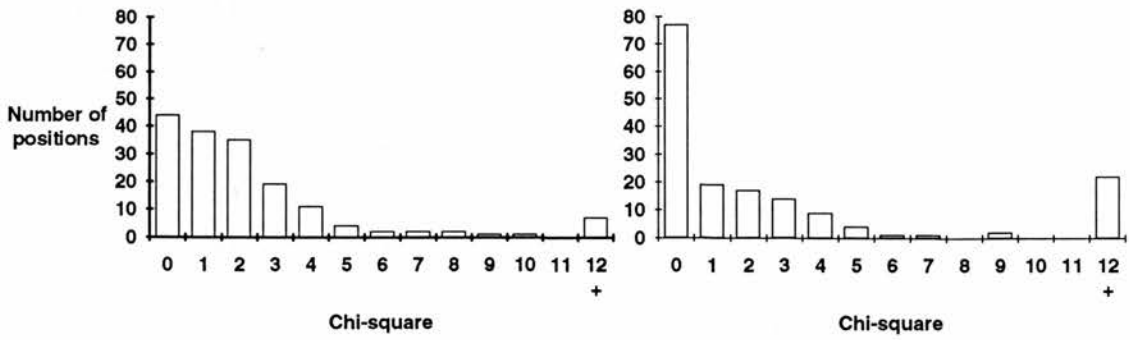

Figure 7.4. Variance histogram for $\alpha = 1000$ (left), and $\alpha = 1$ (right)

Conflict measures

The overlap between the three-beacon sonic and infrared position estimate distributions is given in Figure 7.5. The series of black diamonds marks the average overlap across the 166 sampling positions in the experimental pen. The thick grey lines mark the range between maximum and minimum overlap for each value of α. It can be seen that for strict matching, there are positions where the two systems agree completely, and positions where they disagree completely, although on average, their estimates overlap by about 20%.

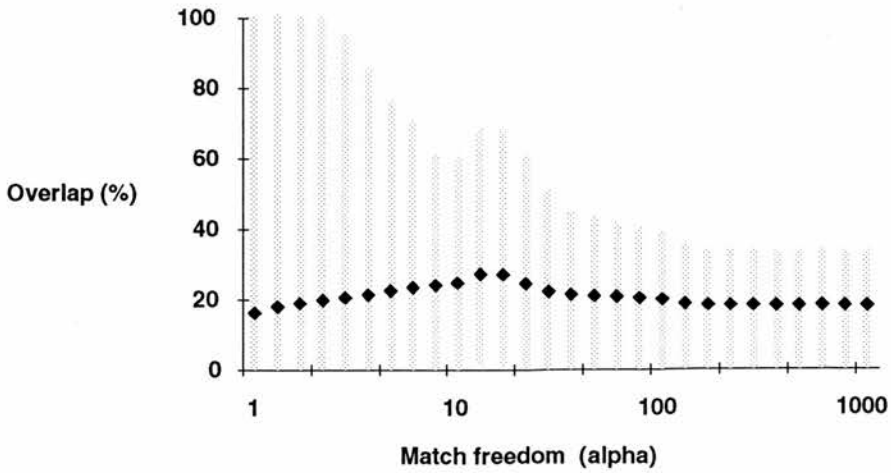

Figure 7.5. Overlap between sonic and infrared estimate distributions

In fact, the average overlap is consistently about 20%, but as the matching becomes less strict, it is found that the maximum overlap drops steadily. At maximum

match freedom, the estimate distributions for the sonic system are known to be flat (all positions in the operating space are equally likely candidates for the robot's location); since the infrared system continues to be able to segment space even under these conditions, the overlap reaches some steady-state amount.

Summary

As the sonic and infrared systems both perform best for fairly strict matching, the combination of the two systems benefits from the union under these conditions, both in terms of positioning certainty and accuracy. However, in the combined case the best performance is obtained for α's around 4 or 5, rather than the values of 1 to 3 which were best for the individual systems. Even though the sonic system is information-free for loose matching, and estimates the robot's position always to be somewhere near the centre of the pen, regardless of where it actually happens to be, the combined distributions preserve the infrared system's persistent small degree of certainty (about 3 bits) and accuracy (average estimate errors of 60 cm, substantially less than the sonic system's 100 cm average error under these conditions).

7.2 Sonic and ultrasonic systems

As discussed in detail in Section 6.3, the qualitative ultrasonic map format chosen consists of a list of 16 numbers for each of the 166 sampling positions; the first number in the 16 is the distance to first sharp reflection of the sonar signal after its emission, with the robot facing "direction 0"; the next number is the distance to the first reflection with the robot facing "direction 1" (22.5° clockwise of direction 0), and so on. The sonic map is as above, and again the assumption of three active beacons is made.

Certainty

Figure 7.6 summarises the amount of information available to the robot given both sonic and ultrasonic systems. The results in the figure were computed in the same manner as for the combined sonic and infrared systems. Again, we see that the combined information is never less than the information available to the best-informed of the two individual devices. Since it was clear from the results in Section 6.3 that the ultrasonic first-reflection field is very well segmented regardless of α, by virtue of the constraints on sonar data collection imposed by the geometry of the room, it is not surprising that the combined sonic and ultrasonic field is similarly information-rich. Just

as did the sonar system alone, the combined systems can always distinguish 30 to 50 places in the pen, and under optimal match conditions (α near 30) all 166 sampling positions are uniquely recognisable.

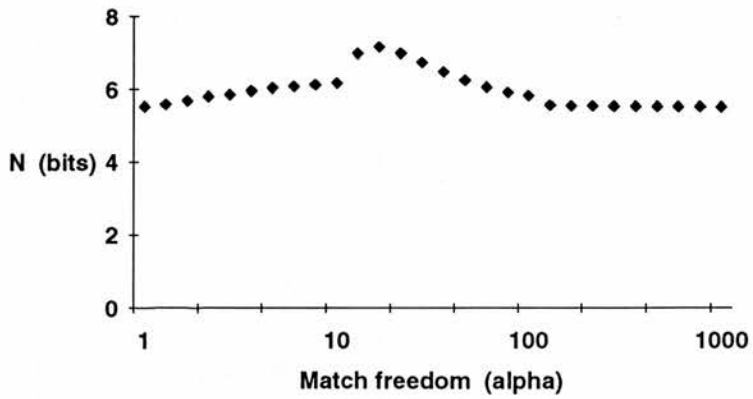

Figure 7.6. Amount of information in the combined sonic and ultrasonic fields

Accuracy

The accuracy of the conditional expectation and MAP estimates of the combined distributions (Figure 7.7) is influenced by the characteristics of the individual systems. The optimal matching conditions for the ultrasonic system occur with α over 30, and here the combined systems estimate the robot's position on average to within less than 10 cm of where it actually is. For strict matching, however, the ultrasonic system is the limiting factor, so that average errors in this case stay between 30 cm and 40 cm. At maximum match freedom, the combination of systems is as accurate as the sonar system alone (positioning errors on average 40 cm), even though the sonic system's estimates are off by an average of 100 cm under these conditions.

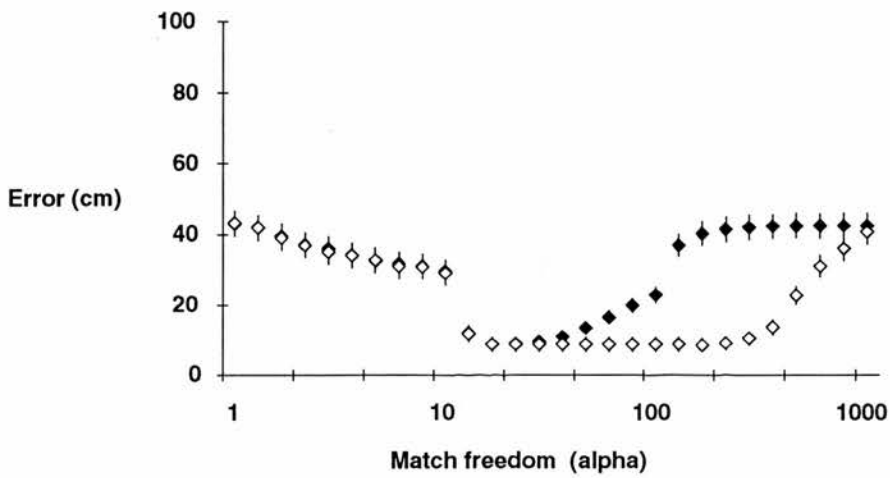

Figure 7.7. Accuracy of combined sonic and ultrasonic systems

Variance

As the certainty and accuracy results for combining the sonic and ultrasonic sensory systems are dominated by the results for the ultrasonic system alone, it is not surprising that the same is the case for the pattern of estimate distribution variances, given in Figure 7.8. Briefly, it seems again that the combined system tends to be under-confident with α less than 10, and overconfident with α between 10 and 100. A fuller discussion of this particular variance pattern can be found in Section 6.3.1.

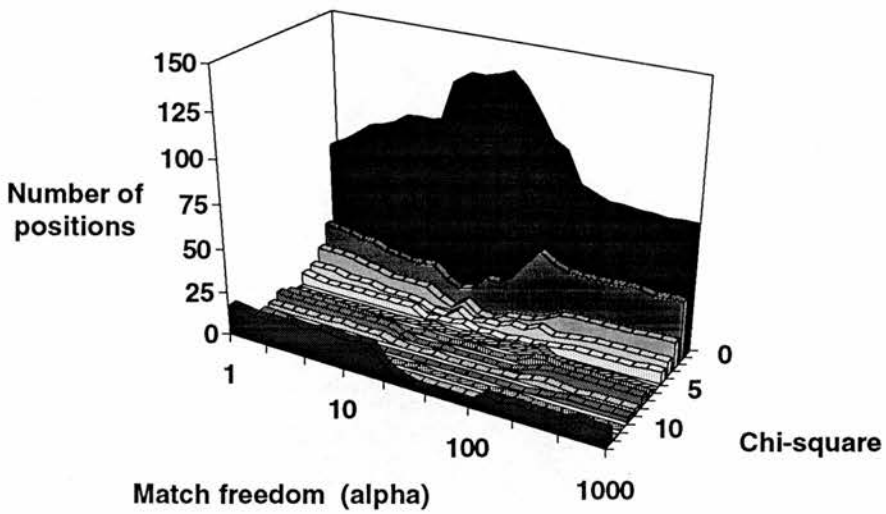

Figure 7.8. Conditional variances of the combined sonic and ultrasonic distributions

Conflict measures

The pattern of overlaps between the sonic and ultrasonic estimate distributions is given in Figure 7.9. In the figure, it is clear that the overlap is greatest for strict matching, with complete agreement at some positions, no agreement at others, but about 20% overlap on average. Both the sonic system and the ultrasonic system are comparably accurate in the first decade of α. However, for larger α, the sonic system's estimate distributions rapidly broaden, while the ultrasonic system remains quite accurate; thus, the average overlap between the two shrinks to an average of only 5%, with no positions at which the overlap is greater than 20%.

Summary

The sonic and ultrasonic systems perform best for strict and moderate matching, respectively. Therefore, the combination of the two systems is not much better in terms

of certainty or accuracy than the best of the individual devices, which happens to be the sonar system. However, as the sonic system is still reasonably accurate where the ultrasonic system performs optimally (at α near 30), position estimates for the combined systems are in error by a few cm less on average than they were for sonar alone. Also, although the sonic system is uncertain and inaccurate for loose matching, the combined distributions preserve the ultrasonic system's persistent large degree of certainty (nearly 6 bits) and accuracy (average estimate errors of 40 cm, compared with the sonic system's 100 cm).

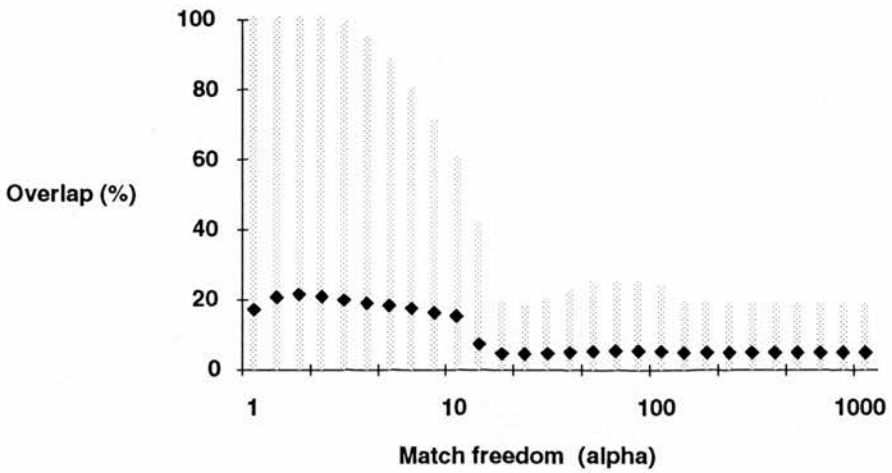

Figure 7.9. Overlap between sonic and ultrasonic estimate distributions

7.3 Infrared and ultrasonic systems

There remains a third pairing of qualitative maps, namely those of the infrared and ultrasonic systems. As in the previous sections, position estimate distributions for these two systems can be multiplied and renormalised to produce combined distributions.

Certainty

If the information measure is computed for the ultrasonic and infrared distributions in combination, the results are as in Figure 7.10. The combined systems are, as expected, at least as well informed as either of the individual systems. The best of the two systems (ultrasonic) is already very certain about position, and the infrared system segments space quite well also, even under conditions of maximum match freedom. Consequently, when these two systems are combined, we find that their hybrid estimate distributions have on average over 6 bits of information, regardless of α, so that the combined systems can always distinguish more than 60 places in the pen.

Where the sonar system performs optimally (α near 30), the information peaks at its maximum of about 7.38 bits (all sampling positions uniquely identifiable).

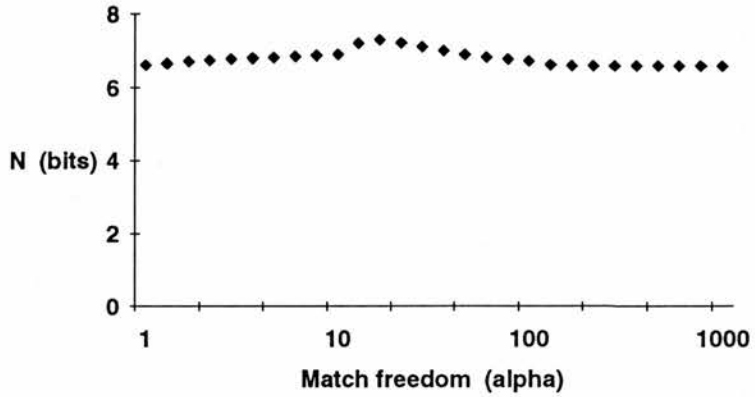

Figure 7.10. Amount of information in the combined ultrasonic and infrared fields

Accuracy

Since the combined infrared and ultrasonic fields have nearly the maximum possible amount of information in them, it follows that the position estimate distributions must be quite sharply peaked. The robot using this combination of systems is therefore certain of its estimates. It is apparent from Figure 7.11 that the accuracy of the conditional expectation and MAP estimates of the combined distributions is also quite high. The two estimators do not differ significantly for any match freedom; at best (α near 30) both estimate the position of the robot to within an average error of about 5 cm, and at worst to within less than 20 cm.

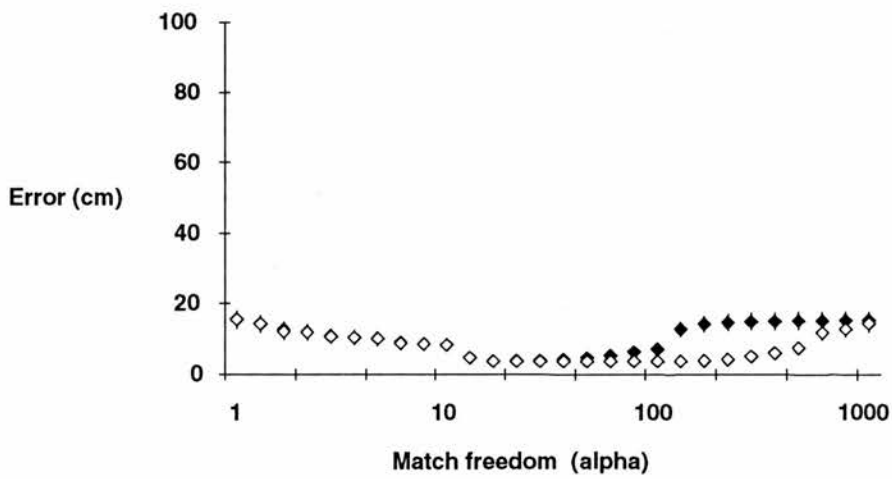

Figure 7.11. Accuracy of combined ultrasonic and infrared systems

Variance

The pattern of the conditional variances for the range of match freedoms (Figure 7.12) is particularly interesting. The comparison baseline is the $\alpha = 1000$ slice, shown also in the left half of Figure 7.13. Here, it can be seen that almost everywhere in the pen, the actual position lies within a small number of "standard deviations" of the conditional expectation. This is a direct consequence of the accuracy of the combined system. Since the position estimates are known to be very accurate, the actual position is bound to lie close to the spatial mean in deviation terms, unless the conditional variance is very small indeed.

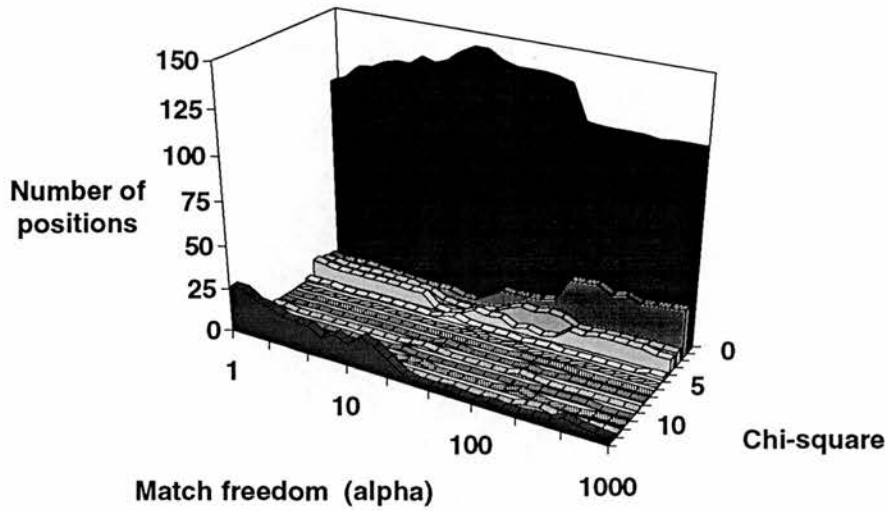

Figure 7.12. Conditional variances of the combined ultrasonic and infrared distributions

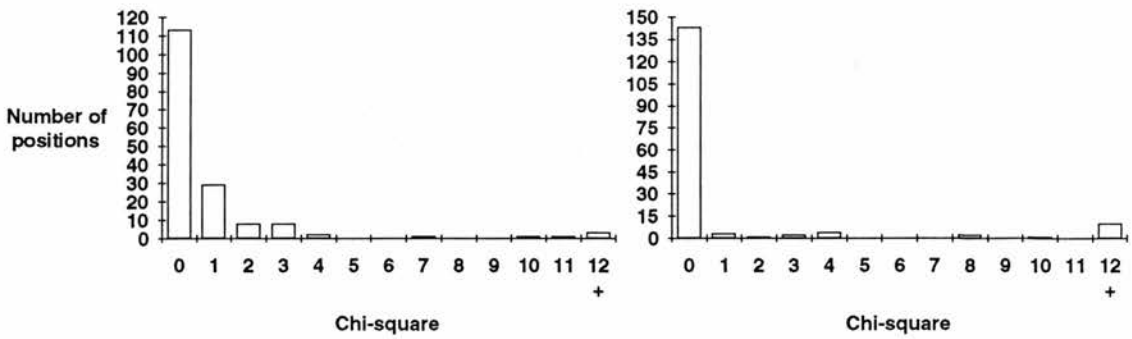

Figure 7.13. Variance histogram for $\alpha = 1000$ (left), and $\alpha = 40$ (right)

Still, the combined systems' variances do follow the two general trends seen repeatedly in the analyses in Chapter 6 and thus far in this chapter as well: for tight

matching, there is a slight decrease in the accuracy of the conditional expectation as a position estimator, but the variances of the distributions presumably remain quite small, so that the system is overconfident of its estimates at some positions (apparently about 25 of them for $\alpha = 1$); similarly, where the accuracy of the combined systems is maximum (near $\alpha = 40$), the variances of the distributions again seem to remain unchanged, so that given the increased accuracy, the systems are a little underconfident of their estimates (this can be seen in the right half of Figure 7.13, which shows that almost all positions fall into the chi-square 0 bin for $\alpha = 40$).

Conflict measures

The pattern of overlaps between the infrared system's estimates and those of the ultrasonic system is given in Figure 7.14. It is clear from this figure that regardless of match freedom, there are positions where the agreement is perfect, and positions where there is no agreement at all. On average, the pairs of distributions tend to overlap by around 20%. As the average overlap does not appear to peak near $\alpha = 30$, it seems reasonable to conclude that the increased accuracy of the combined system at this match freedom derives from the increased precision of the sonar system's estimates under these conditions, but that this does not engender an increase in the agreement of the two contributing systems.

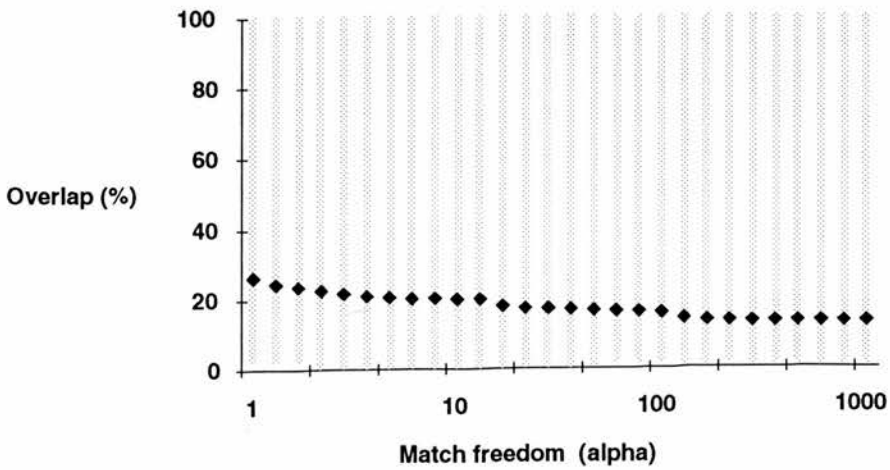

Figure 7.14. Overlap between ultrasonic and infrared estimate distributions

Summary

The infrared and ultrasonic systems combined perform very well for any match freedom from strict to loose. At best, position estimates from the combined systems are in error by 5 cm on average, and at worst by less than 20 cm. This high certainty

and accuracy derive largely from the inherent segmentation of space afforded to these systems by simple data/no-data considerations. Although the infrared system's persistent degree of certainty is only around 3 bits, and that of the ultrasonic system less than 6 bits, the particular geometry of this inherent segmentation is such that when combined, these certainties merge to reach a steady state minimum of nearly 7 bits.

7.4 Sonic, infrared, and ultrasonic (i.e., all) systems

It remains only to consider the case when all three individual systems are taken in unison. By multiplying and renormalising the position estimate distributions for the sonic system (with three active beacons again), the infrared system (also with three beacons on), and the ultrasonic system, combined distributions are generated whose certainty and accuracy measures can be computed in order to characterise the expected performance of localisation by all three systems at once.

Certainty

The entropy/information analysis results for the three systems are shown in Figure 7.15. It can be seen that the addition of the sonic system does not lead to a three-way combined system with significantly more certainty than was available in the distributions of the infrared-ultrasonic pair (see Figure 7.10). This is not surprising, as the sonic data field has generally less information than either of the other two data fields. Thus, as for the IR-sonar pair, by combination the estimate distributions of all three individual systems, the robot always has nearly 7 bits of information about place.

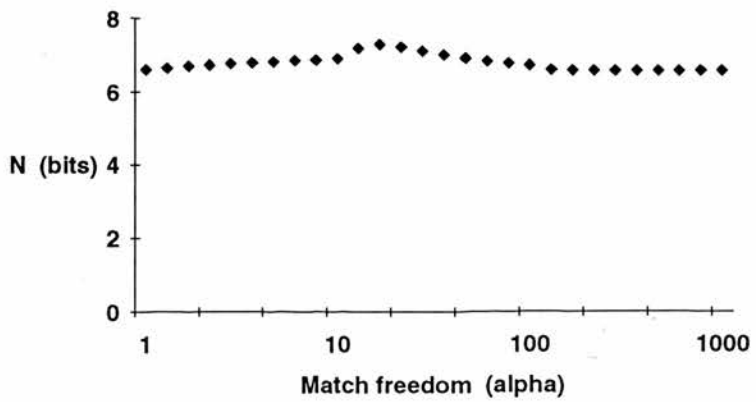

Figure 7.15. Amount of information in the combined sonic, infrared, and ultrasonic fields

Accuracy

Similarly, the accuracies of the conditional expectation and MAP of the combined distributions as estimators of position are not significantly different to what they were for the IR-sonar pair. This can be seen by comparing Figure 7.16 below with Figure 7.11. Again, position estimates are at their best with α near 30, for which they are in error by around 5 cm on average.

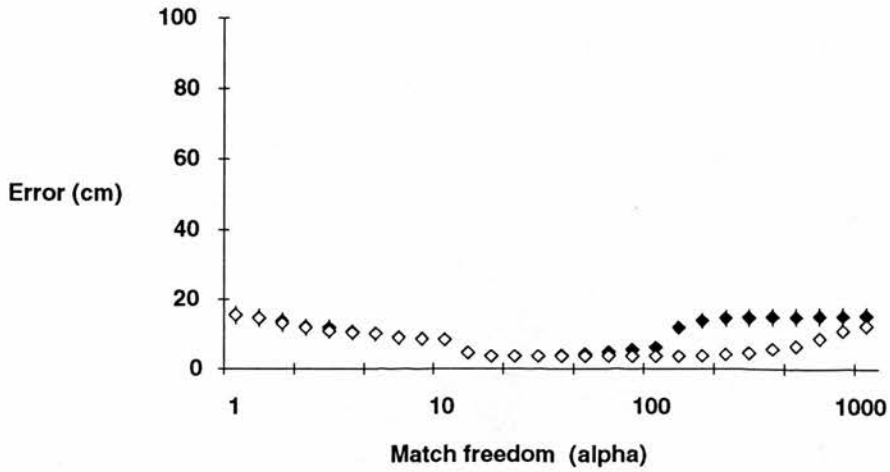

Figure 7.16. Accuracy of combined sonic, infrared, and infrared systems

Variance

Finally, we have the conditional variances for the three combined systems portrayed in Figure 7.17. Again, it can be seen that the pattern of variances is quantitatively almost identical to the results presented for the infrared and sonar system pair, with the usual over- and under-confidence trends.

Conflict measures

While the certainty and accuracy of localisation by all three systems are virtually identical to the results of the same analysis for only infrared and ultrasonic systems combined, the nature of overlaps between the three distributions is decidedly different, as can be seen at once by comparing Figure 7.18 and Figure 7.14. For the three systems in unison, the maximum overlap continues to occur where α is less than 10. Here, there are some positions where all of the systems agree completely, and some where they disagree completely, although the overlap is on average only 10%. However, for larger α, the sonic system's estimate distributions rapidly broaden, while both the infrared and ultrasonic systems remain reasonably accurate, by virtue of the inherent segmentation of space that is afforded to them by the data/no-data condition.

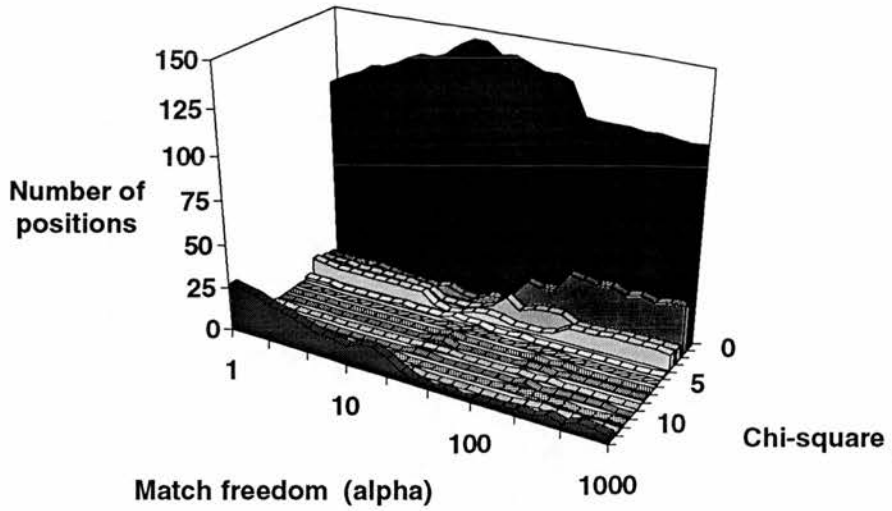

Figure 7.17. Conditional variances of the combined sonic, infrared, and ultrasonic distributions

Thus, where for the infrared-ultrasonic pair there was perhaps a 20% overlap even for loose matching, this already fairly small agreement between two peaked distributions sits atop broad sonic distributions; and the overlap between a peak and a fairly flat plane is just the small disk-shaped segment of the plane at the base of the peak. The consequence is seen in Figure 7.18 as the decrease of the average overlap between the three systems to almost zero, with even the maximum overlap being something less than 10%.

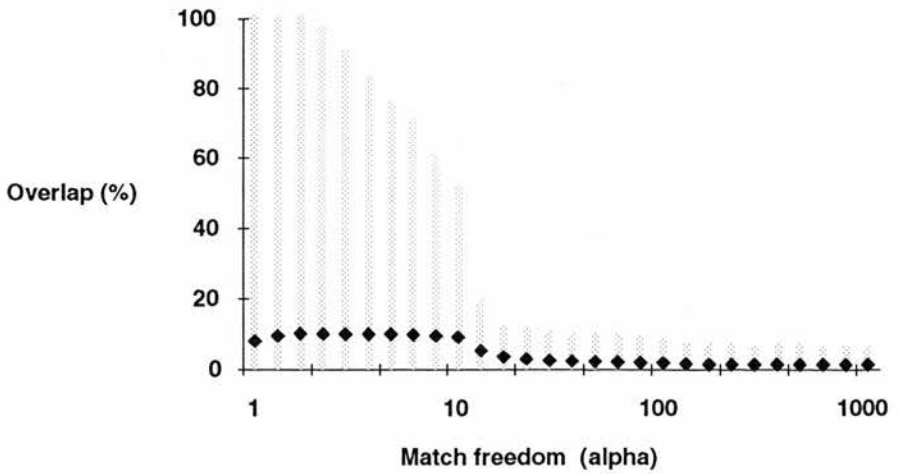

Figure 7.18. Overlap between sonic, infrared, and ultrasonic estimate distributions

Summary

The sonic, infrared, and ultrasonic systems combined perform no better than only the infrared and ultrasonic pair of systems. At best, position estimates from the combined systems are therefore again in error by 5 cm on average, and at worst by less than 20 cm. The high certainty and accuracy derive largely from the segmentation of space inherent in the IR and sonar data fields, which are apparently neither helped nor hindered by the presence of the sonic system.

7.5 Summary of combined systems

In this chapter, the three locative sensing systems analysed in detail individually in Chapter 6 were merged so that the performance of combinations of systems could be assessed. The three possible pairings were studied (sonic with IR, sonic with sonar, and IR with sonar), as well as the union of all three systems. The analysis proceeded by multiplying and renormalising the position estimate distributions for the individual systems in order to generate combined distributions, as proposed in Section 5.2.5. The various system combinations were then evaluated by computing the statistical certainty and accuracy quantities in the combined distributions.

7.5.1 Comparative performance

The positioning certainty results for the three pairs of systems are compared in Figure 7.19. Note that since the results for the three-way combination of sonic, infrared and ultrasonic were nearly quantitatively identical to the results for just IR and sonar together, only the latter have been included, for clarity.

From Figure 7.19 it is clear that with matching fairly strict ($\alpha < 10$), the data fields for all combinations of systems have about the same amount of information in them; the 6 or so bits of information available mean that a robot using one of these combinations of systems could distinguish over 60 places in the experimental pen. As the match strictness is decreased, the certainty diverges somewhat for the various combinations: the sonic and infrared systems have the least information in them individually, but the inherent segmentation in the infrared data field means that their pair settles at just about 3 bits of information for maximum match freedom; the ultrasonic system has more inherent certainty than its IR counterpart, so that the sonic-ultrasonic combination reaches a higher steady-state certainty of about 5.5 bits; finally,

the persistent information in the individual IR and sonar fields means that when paired, their combined data fields contain a constant 6 or more bits of information, as does the combination of all three systems, whose performance is obviously most heavily influenced by these two.

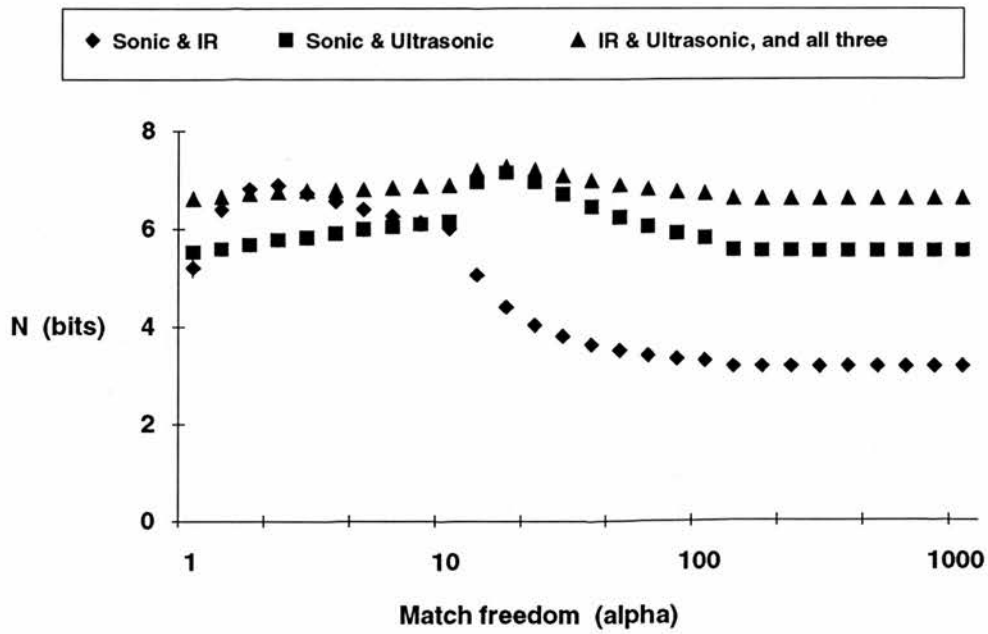

Figure 7.19. Comparative positioning certainties of three pairs of systems studied; the results for the three-way combination are almost identical to those for the IR-US pair, so are omitted

The accuracies of the various combinations of systems do not behave in quite the same way, as shown in Figure 7.20. For strict matching, the sonic-ultrasonic pair is significantly less accurate than either of the other two pairings; in fact, this combination appears only to be as accurate as was the sonar system on its own, with both conditional expectation and MAP estimators in error on average by about 30 cm. In contrast, both other pairs and the three-way combination are considerably more accurate for strict matching ($\alpha > 3$), in error by perhaps 10 cm on average. Note that the IR-ultrasonic pair does not suffer from the ultrasonic system's inaccuracy for strict matching as much as the sonic-ultrasonic pair does, as the IR system on its own is considerably more accurate than the sonic system under these circumstances (see Figure 6.43).

However, as the match strictness conditions become optimal for the sonar system, the pairs in which it participates become very accurate localisation systems. The MAP position estimates the robot's location with less than 10 cm error on average, and for the IR-ultrasonic combination, this average is improved to 5 cm.

When match freedom is increased towards its maximum, the conditional expectations become steadily worse as position estimators, and even the MAP estimators eventually reach some steady-state average positioning error. For the IR-ultrasonic pair, this average error is only about 15 cm. The sonic-US combination fares considerably less well at nearly 40 cm average error. Finally, the sonic-IR combination is least accurate for maximum match freedom, with an average error of just over 60 cm.

The positioning errors for the combination of all three systems are effectively identical to those for the IR-ultrasonic pair, for any value of the match freedom parameter α, so that these results are omitted in Figure 7.20. Since the same effect was seen for the certainties with all three systems in combination, it is clear that the sonic system does not contribute in a significant way (either positively or negatively) to the three-way combination.

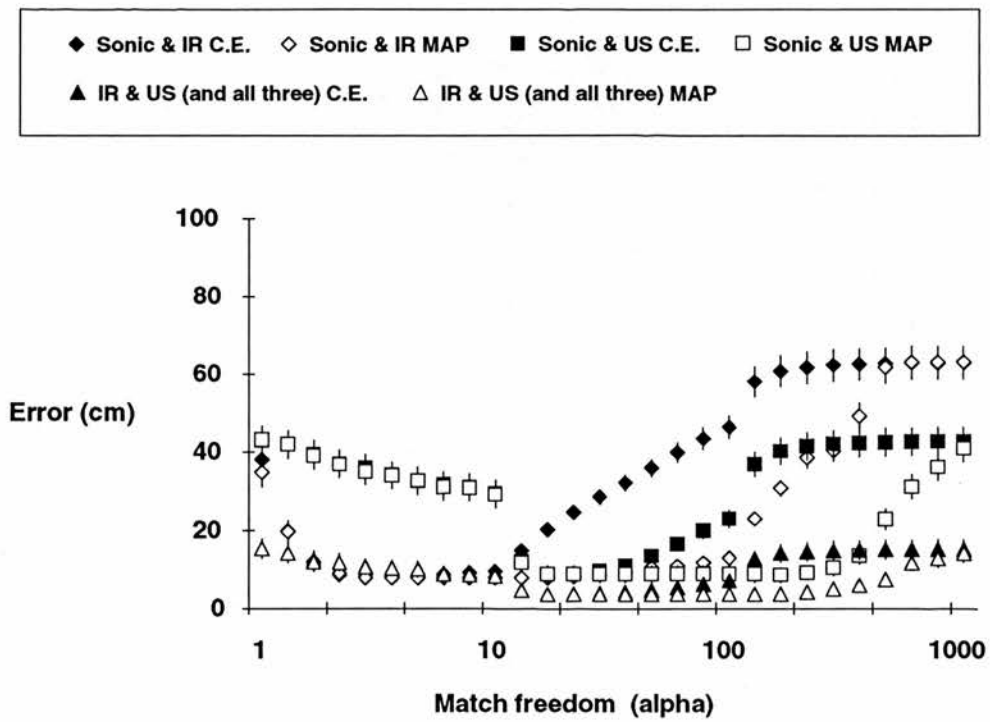

Figure 7.20. Comparative positioning accuracies of conditional expectation and MAP estimators for three pairs of systems studied; again, the results for the three-way combination are almost identical to those for the IR-US pair and are therefore left out

Although the relevant results have not been included here, it has been found that when the sonic and infrared systems are used with less than their maximum of three beacons, the certainties and accuracies of system combinations in which they participate are diminished.

7.5.2 Best- and worst-case performance

As in Section 6.4.2, we illustrate the performance of the various combinations of systems by comparing the abilities of the combinations of systems to distinguish between the same square-shaped set of 28 sampling positions (see Figure 6.44). Again, the "best" case results for a particular position are obtained by setting α to the value at which average positioning error for the system was minimal, and the "worst" case results are obtained with $\alpha = 1000$, which is the match freedom at which all four combinations of systems suffer from the largest average position estimate error.

Sonic and infrared combination

The best- and worst-case performance for the sonic and infrared combination is as in Figures 7.21 and 7.22, respectively. It is obvious that the best-case performance is very good, in that almost all positions are correctly identified. In a few cases, there are minor confusions between adjacent positions, but there are no instances of ambiguities between non-adjacent positions.

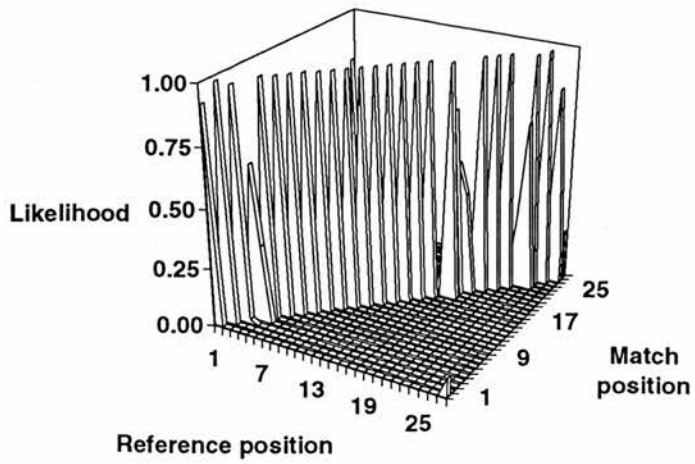

Figure 7.21. Best-case performance of sonic and infrared systems combined ($\alpha = 3$)

The worst-case performance (Figure 7.22), shows as we would expect that the combination of systems benefits from the infrared system's persistent small degree of certainty about place even with maximum match freedom between mapped and current signal intensities. The results here are effectively identical to the worst-case for the infrared system alone (see Figure 6.53), as the sonic system's ability to localise under worst-case conditions is practically non-existent (see Figure 6.52).

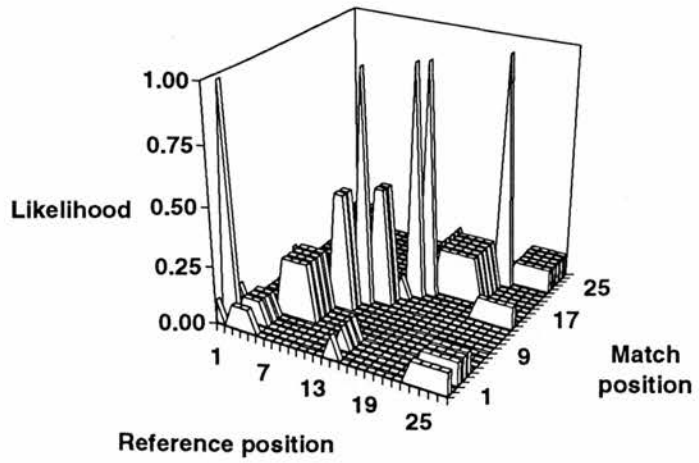

Figure 7.22. Worst-case performance for sonic and infrared systems combined

Sonic and ultrasonic combination

Again, the best-case performance (Figure 7.23) for this combination of systems is almost perfect localisation. There appear to be some small confusions between adjacent locations near positions 7, 13, 22, and 28; localisation is correct everywhere else.

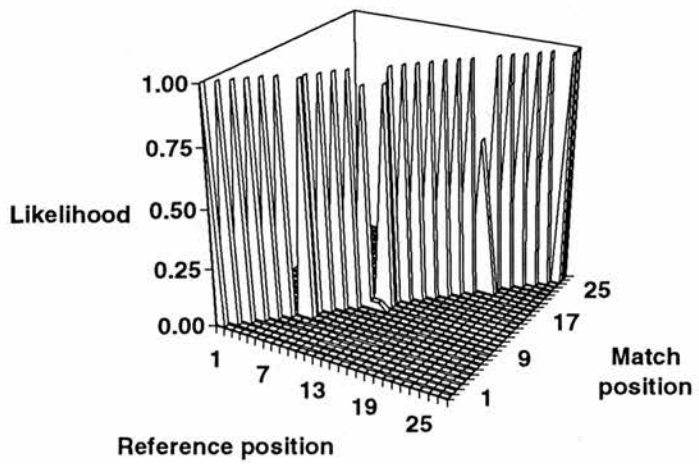

Figure 7.23. Best-case performance of sonic and ultrasonic systems combined ($\alpha = 30$)

As expected, the worst case results in Figure 7.24 are obviously influenced primarily by the ultrasonic system. Comparing this plot with that of the sonar system alone (Figure 6.54), we see that there are several positions in both at which localisation is correct even for maximum match freedom; it is also clear from both that ambiguity in

location estimation tends to be between adjacent positions, except at position 2, which is apparently slightly confused with 9 and 10, and positions 6 to 10, which seem to share indistinguishable features with positions 21 to 23.

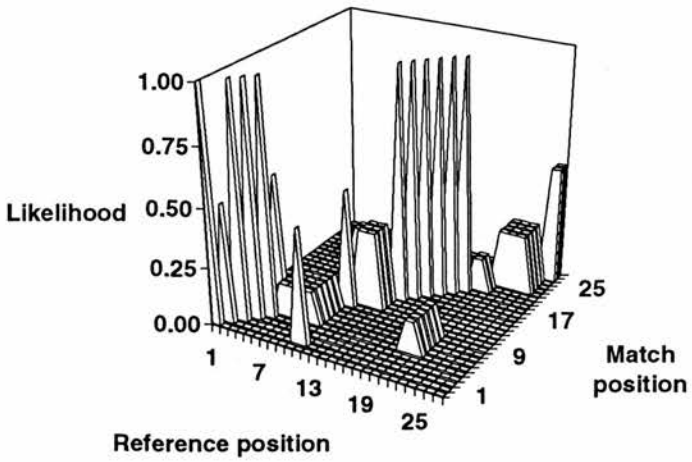

Figure 7.24. Worst-case performance for sonic and ultrasonic systems combined

Infrared and ultrasonic combination

The combination of infrared and ultrasonic also achieves near-perfect localisation with the right choice of match strictness (Figure 7.25). Adjacent-position ambiguities are again seen near positions 7, 13, 22, and 28, suggesting that these confusions come from the ultrasonic system, as they did not occur in the sonic-IR pairing. Localisation is correct everywhere else.

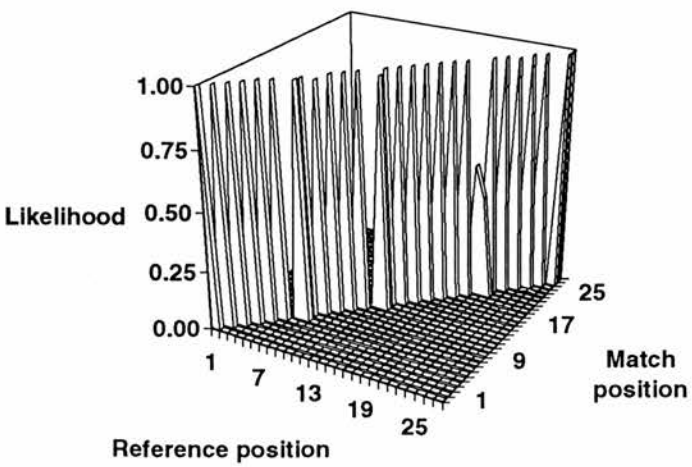

Figure 7.25. Best-case performance of infrared and ultrasonic systems combined ($\alpha = 30$)

From Figure 7.26 it is clear that the worst-case performance of the infrared and ultrasonic combination benefits from the differences in the segmentation of space inherent in both systems individually. That is, with IR alone it is still possible to recognise about 7 of the 28 places reasonably well even for maximum match freedom (see Figure 6.53). Several more remain distinguishable by the ultrasonic system (Figure 6.54). In combination, localisation is still perfect at 15 positions under maximally free matching conditions, showing the segmentation of space that comes from the bearings of IR beacons to be different to the sonar system's segmentation by room geometry, so that the partitioning of space with these systems in combination is less "coarse" (contains a greater number of regions) than for either alone.

At positions where localisation is not perfect, ambiguities invariably occur only between adjacent positions: in particular, the pairs of positions 13 and 14, 20 and 21, and 27 and 28 are not distinguishable, nor is the threesome 24, 25, and 26, and the foursome 6, 7, 8, and 9. These ambiguities were visible to a lesser degree in the best-case plot (as mentioned previously), and coincide with places where the IR beacons are not visible, or appear to have the same bearing as at neighbouring positions, and the geometry of the room (and therefore pattern of ultrasonic specularities) is not unique either.

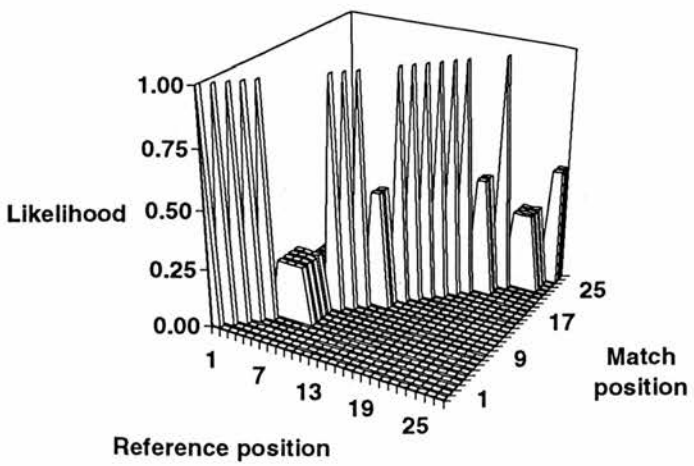

Figure 7.26. Worst-case performance for infrared and ultrasonic systems combined

Combination of all three systems

As discussed several times already in this chapter, most recently in the performance comparison section (7.5.1), the combination of all three systems tends to perform both

at best and at worst almost identically to the infrared-ultrasonic pair. The 28-position plots for the three-way combination are therefore omitted.

7.5.3 Conclusions

We saw in Chapter 6 that localisation by means of the amplitudes of tones from loudspeaker "beacons" is most accurate, and most certain, when the matching between stored data from past visits and the data currently being received is quite strict; the same was found to be true of localisation by the bearing and intensity of infrared light sources. However, localisation by means of the pattern of distances to the closest ultrasonic specular reflector in a number of directions works best with somewhat less-strict matching. Consequently, combinations of these three systems work best under particular conditions as well.

Thus, since the ultrasonic system is much more effective than the other two individually, any combination in which the ultrasonic system participates is best used with less-strict matching. Similarly, for the sonic-infrared pair, the best performance obtains with strict matching.

In order to compare the various combinations, the certainty and accuracy of the combined position estimates were computed with the robot placed at each of 166 positions in an experimental pen. Just as for the individual systems, these 166 certainties and accuracies were averaged in order to summarise the performance of a particular combination of systems. The relatively small standard errors in both the certainty and accuracy averages show them to be usefully representative of a combination of systems' typical performance for a given match freedom.

Certainty

All combinations of systems segment the environment into about 60 domains for strict matching. However, as the match freedom increases, the sonic system becomes ineffective, and eventually (at maximum match freedom) cannot distinguish any two places in the pen. Therefore, for the sonic-infrared and sonic-ultrasonic combinations, the performance of the pair eventually degrades to the performance of just the infrared or ultrasonic systems alone, respectively.

The infrared and ultrasonic systems continue to be able to segment space even at maximum match freedom, so that both reach a non-zero steady state certainty (as discussed in Chapter 6, the infrared beacons have zones of visibility and invisibility,

which persist regardless of intermediate degrees of visibility; similarly, the sonar system has data/no-data conditions dictated by the geometry of the environment, as it is impossible for the robot to obtain sonar scans in some directions at some positions). In fact, this "inherent" segmentation in the sensory fields differs for the two systems, so that in combination, it is more finely resolved. Therefore, the infrared-ultrasonic pair maintains a positioning certainty of about 6.5 bits (i.e., around 90 places are recognisable in the pen) under any circumstances.

Since the infrared and ultrasonic systems are generally more effective than the sonic system, these former two dominate the results for the three-way combination of all three systems, to the point where the results for the threesome are virtually indistinguishable from those of the IR-US pair.

Accuracy

It was found that the sonic-infrared pair performed best for strict matching, in error by less than 10 cm on average. As the match freedom is increased, the ultrasonic system enters its region of optimal performance, achieving the best average error of all three individual systems. The sonic system's accuracy quickly becomes poor in this range, although the infrared system remains reasonably accurate; thus, the infrared-ultrasonic pair attains the best average error of the three pairs of systems, about 5 cm.

With maximum match freedom, the sonic system loses all localisation ability, and constantly estimates the robot's position to be in the middle of its operating space; the infrared system still performs reasonably, and the ultrasonic module even better, because of the inherent partitioning of space afforded to them. It is not surprising, therefore, that the infrared-ultrasonic pair fares best in these extreme conditions (average positioning error around 10 cm), that the sonic-ultrasonic pair is somewhat worse at 40 cm (effectively the performance of the ultrasonic system on its own), and that the sonic-infrared pair is least accurate on average at 60 cm error (the performance of the infrared system on its own).

Just as for the certainty results, the three-way combination of sonic, infrared, and ultrasonic has about the same average accuracy across the entire match freedom range as infrared and ultrasonic combined without the sonic system.

Variance

For all combinations of systems, we see the same two trends in the behaviour of the combined estimate variabilities across the range of the match strictness parameter as

those that emerged in the analysis of individual systems. The first of these is that when matching is strict, there tend to be several positions at which estimate variances are smaller than is warranted by the magnitude of the actual positioning errors, resulting from the instability of the process of strict matching; the combined estimate distributions are likely to be highly peaked (they are products of distributions that are themselves highly peaked). The combined systems might even believe that there is only one position where the robot could be, so that the estimate distribution has very little conditional variance.

The second trend is the tendency towards underconfidence that arises for less-strict matching (α between 10 and 100). This effect occurs because the increased match freedom means that new positions (possibly spatially distant from the set of likely candidates prevailing for strict matching) become possible locations for the robot, which spreads the estimate distribution variances. However, it is clear from the results in Figure 7.20 that the conditional expectation is still a good estimator of position (extremely good for the IR-US pair); where there is a disproportionate increase of conditional variance versus conditional expectation, the combined systems' are underconfident.

Conflict

The overlaps between the estimate distributions of the individual systems as they are combined into a single distribution are generally less than 20%. This is an artefact of the peaked shape of distributions likely to result where an exponential matching function is used. Where the sonic system participates in a combined system, the overlap drops almost to nothing for loose matching, as the sonic system's distributions are flat under these circumstances; the overlap of a peaked distribution with a flat one is just the disk in the flat distribution that lies below the peak, which represents only a small percentage of the total possible overlap between the two distributions.

For the infrared-ultrasonic system pair, it is found that at some locations in the environment, the two systems agree perfectly about where they think they are (100% overlap), although they disagree completely at others, across the full range of match freedom. On average, however, the overlap hovers around 20%.

Summary

To summarise, then, it is clear that localising with the infrared and ultrasonic systems is as good as localising with all three systems, in terms of positioning certainty and

accuracy, and is better than localising with either of the other two pairs of systems. This is presumably because the ultrasonic system was the best individual performer, and the infrared system the second best, so that it is not unexpected that they form the most effective pair. The results suggest that the sonic system neither adds to nor detracts from the performance of infrared and ultrasonic, as it apparently contributes nothing to the certainty or accuracy of the three-way combination.

At the end of Chapter 6, it was pointed out that localising by specularity patterns relies upon a relative stability of the geometry of the environment (i.e., assumes that layout does not change much between visits); this was less of a constraint for the sonic and infrared systems, as they assume only that their reference beacons remain relatively fixed, which is almost certainly a less demanding condition of use. While the question of environmental stability will be discussed in the next chapter, it seems fair to say for the moment that since combining systems apparently results in improved performance, a robust approach to localisation might be to combine a system that is less accurate but also less susceptible to changing circumstances (like the infrared beacon detection system), with one that is accurate but intolerant of change (like the ultrasonic system).

Chapter 8

Extensions and Future Work

There were a number of suppositions made in the course of the localisation analysis that will now be explored in somewhat more detail. These are primarily concerned with assumptions made to simplify the analysis, and as such warrant further discussion and possibly justification. In addition, we will indicate ways in which both localisation and locomotion could benefit from the techniques of machine learning. Finally, we will discuss other extensions of the spatial competences, including procedures for uniting localisation and navigation.

8.1 Analysis assumption-based matters

The localisation analysis proceeded from a few presumptions that will be considered more thoroughly in this section. Briefly, these were:

1. that an exponential function should be used in comparing mapped and current data;

2. that although the sensory fields of the various devices are analogue and continuous, it was reasonable to decompose space into a mesh of discrete sampling positions for the analysis;

3. that the environment was static between initial mapping and subsequent localisation runs;

4. that the robot had a direction reference that was fixed to the world; and

5. that the robot might be anywhere each time it tried to localise itself.

Each of these assumptions will be taken up in turn.

8.1.1 Match function effects

In Chapter 5, it was argued that the requirements of the task constrain the type of data matching function that is appropriate. Essentially, the match function must map the

difference between the mapped and current sensory states onto a number that quantifies the goodness of match. To recapitulate, this function must return a large value for close matches, and a monotonically decreasing value for increasingly bad matches. Its strictness must be adjustable to allow it to compensate for any natural variability in the sensor data that might arise even under comparable physical circumstances. Finally, where more than one pair of mapped/current data are to be compared, it would be convenient if the value of the match function as applied to the sum of the differences in each pair were equal to the product of the values of the match function applied to each pair individually.

These constraints effectively define an exponential function, and in fact the localisation analysis was based on the function

$$\text{match} = e^{-[(\text{mapdata} - \text{currentdata})/\alpha]^2},$$

where α controls the strictness of the match. While this exponential function is the most appropriate match function for the analysis, the question arises as to whether or not variations on the function itself would have a significant effect on the overall results of the analyses.

Therefore, all of the analyses documented in Chapters 6 and 7 were repeated with two variants of the basic function. The aim was to test the effect of a stricter function (sharper peak at maximum likelihood), and a less strict function. To construct a stricter function, we simply compute the Manhattan distance between data values, rather than the more conventional Euclidean distance, so

$$\text{match} = e^{-|\text{mapdata} - \text{currentdata}|/\alpha^2}.$$

This function is just a pair of exponential functions back to back, so that if the mapped and current data are identical, the match likelihood is 1, and otherwise the likelihood drops off very quickly.

A reasonable less-strict function is the hyperbolic secant function, with Euclidean distances, which is given by

$$\text{match} = \frac{2}{e^{-[(\text{mapdata} - \text{currentdata})/\alpha]^2} + e^{[(\text{mapdata} - \text{currentdata})/\alpha]^2}}.$$

This function looks like a gaussian curve with a flat top. This flat top means that there is a fairly large range of differences between map and current data that will return a high match likelihood.

For all three sensor systems, the data were expressed as 8-bit integers, so that the maximum difference between mapped and current data is 255. In Figures 8.1 to 8.3,

the three match functions are compared across the entire range of data differences, from -255 to 255. The three figures show the effect of the match control parameter. In each case, all three curves reach a maximum value of 1 (maximum match likelihood) when map and current data are identical.

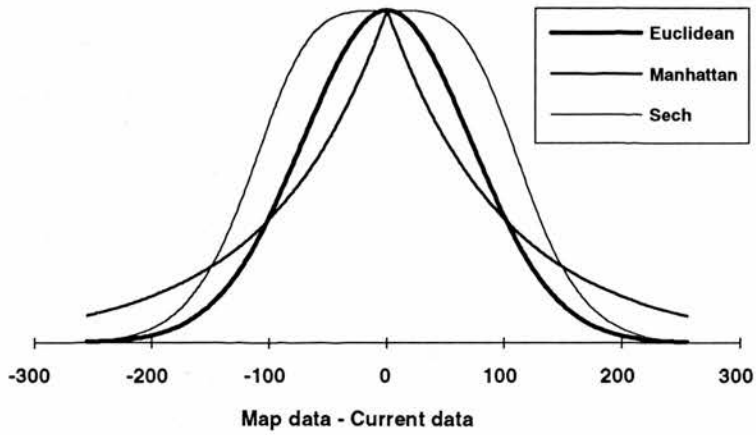

Figure 8.1. Comparison of different match functions for $\alpha = 40$

Figure 8.1 shows the different nature of the three match functions most clearly. The Euclidean curve, showing the match function from which the results in Chapters 6 and 7 were generated, is the usual bell curve. With the Manhattan distance metric, the curve is very sharp, so that only perfect matches are reasonably likely. Finally, the hyperbolic secant is shaped like a bell curve with a flat top (large range of data differences that will still be considered likely matches).

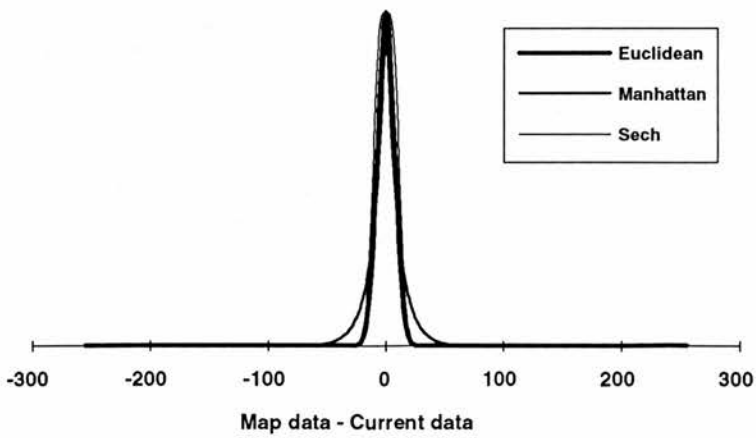

Figure 8.2. Comparison of different match functions for $\alpha = 4$

For comparison, Figure 8.2 shows the three curves with the match control parameter quite small. They are all three in effect equal for matching this strict. Finally, Figure 8.3 shows that for a fairly large α, the sech curve is nearly flat, so that any mapped data will match with any current data; the other two functions are almost as

loose. Even when the data are as different as possible, the Manhattan distance metric function returns a match likelihood of greater than 50%.

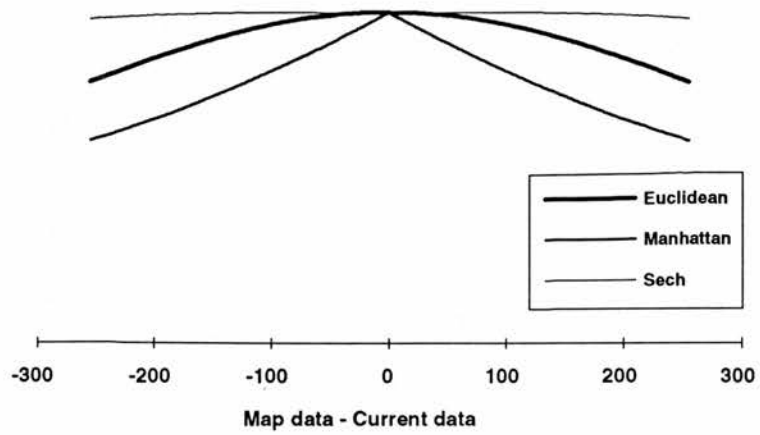

Figure 8.3. Comparison of different match functions for $\alpha = 200$

In repeating the entire spectrum of analyses with the two additional functions, we find no significant differences in the results. Since all three functions are comparably strict with $\alpha < 10$, and comparably loose with $\alpha > 100$, it is not surprising that the conclusions of the analyses hold in these ranges. However, even with α in its second decade, where the differences between the three functions are most noticeable (see Figure 8.1), there are only minor quantitative differences between the three sets of entropy and accuracy results, although the broadness of the top of the secant function generally results in slightly larger standard errors.

The only area of considerable variability is in the conflict measures. With the stricter Manhattan distance function, the overlaps between position estimate distributions are smaller, as the distributions themselves are more peaked. Similarly, the overlaps with the secant function are greater, as this function is broader than the basic Euclidean distance function. In both cases, the conditional expectations and MAP positions remain good estimators of the robot's actual position. This variability is consistent with the nature of the functions.

8.1.2 Continuity versus discreteness of qualitative maps

For the localisation experiments outlined in Chapter 5, the robot followed a path that exhaustively covered the free space accessible to it in an experimental pen. At regular intervals it sampled its sensors and recorded the data. Sensor data for the 166 positions at which this sampling took place are composed to form discrete qualitative data field maps for each sensory modality. This decomposition of space simplifies the localisation

analysis, and since the mesh is reasonably fine (sampling positions are only 23 cm apart), we have argued that inferences drawn from it would hold if the sensory data fields were treated as a continuum.

In order to test this claim, we draw upon an additional set of sensor data gathered for work not reported in this document. The objective of the work was to study camera-based localisation and navigation, and entailed the robot collecting and analysing images at several positions in the experimental pen. In addition to the video data, sonic, infrared, and ultrasonic recordings were made at these positions. Nine of the locations did not coincide with any of the 166 sampling positions considered in the localisation analysis. These are marked by **X**'s in Figure 8.4

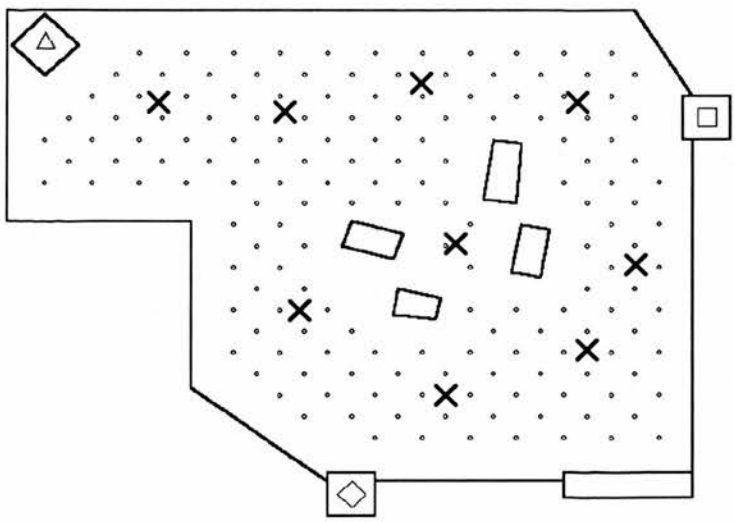

Figure 8.4. Off-grid positions used to test discreteness assumption

The applicability of the localisation analysis conclusions to continuous space was tested as follows: qualitative sensory maps were constructed for each of the three sensor systems from each of the three data sets, as in Chapters 6 and 7; these comprised the "map" data, and the sensor data gathered at the nine off-grid positions comprised the "current" data. Given the states of the sensory fields at each of the nine off-grid locations, the robot computed the likelihoods of being at each of the 166 grid positions. It was found that the conditional expectation of the combined estimates of the three sensor systems differs from the actual location of the robot by less than 30 cm on average, so that the performance for off-grid positions is about as good as for on-grid positions. The MAP estimator does not fare as well, as it attempts to locate the robot exactly on a grid position, while the conditional expectation (spatial mean) can easily lie in space between mesh points.

Figure 8.5 shows how the combined systems weight the robot's likelihood of being at the four grid positions that lie closest to each of the nine off-grid positions. It can be seen that the balance is correct, at least on subjective comparison with the positions of the **X**'s relative to their neighbouring mesh points in Figure 8.4. Therefore, we would argue that the claims made for localisation at the 166 discrete positions considered in Chapters 6 and 7 are also applicable to localisation in the actual continuous space of the experimental pen.

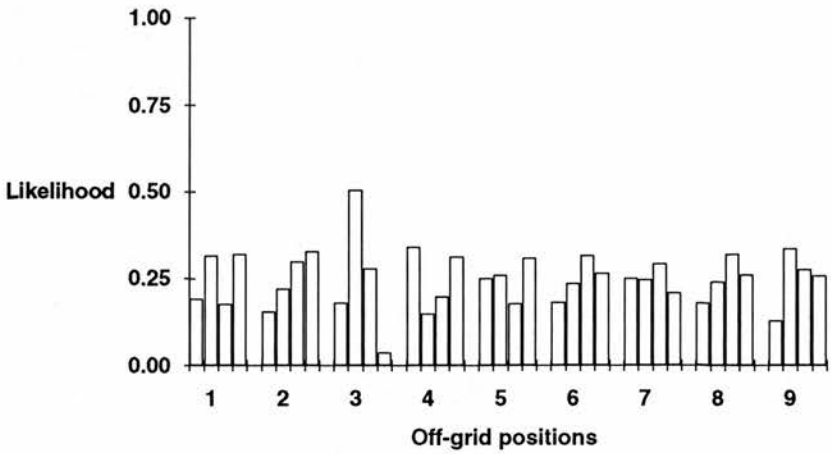

Figure 8.5. For each of the 9 off-grid positions, this plot shows the relative likelihoods of being at one of the four nearest on-grid positions

8.1.3 Beacon and environmental fixity

The sonic and infrared localisation systems depend on locative references that are fixed in the robot's working environment. The sonar system estimates where it is by the pattern of specular reflections it is receiving, thus depends on the geometry of its operating space (the arrangement of surfaces that make up the environment). Therefore, it is expected that the performance of the former two systems would be affected by changes in the arrangement or number of beacons, while the ultrasonic system is affected if the layout of the environment is perturbed.

For example, suppose that the three sonic beacons are moved so that beacon 1 is now placed where beacon 2 was, 2 is where 3 was, and 3 is where 1 was previously; also, assume that the robot's qualitative sonic map was constructed before the reconfiguration, and that it now attempts to localise in the new environment by the sonic system alone. Figure 8.6 shows the information content in the sonic field of the three beacons if they stay in place between robot visits (solid diamonds), and also if the beacons have been shifted between mapping and subsequent visits. It is not surprising

that the information quantity is essentially unchanged, as there are still three beacons with the same regions of audibility; the beacons have really only been "renamed" by the shift proposed above.

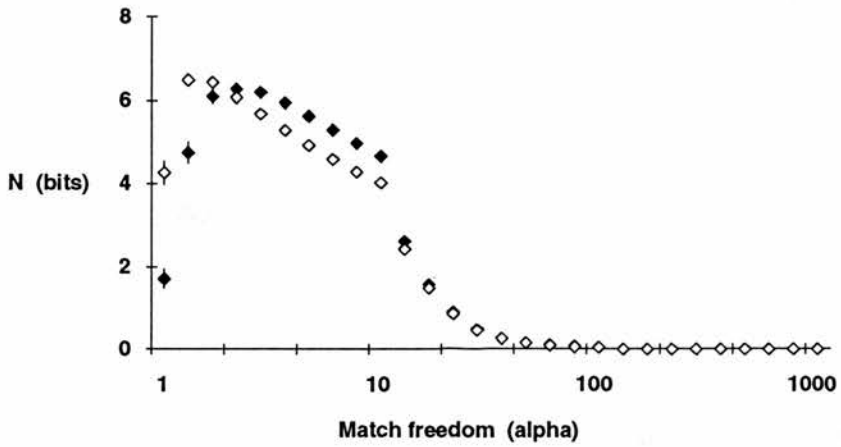

Figure 8.6. Comparison of information in the sonic field the three beacons; ♦ shows information when the arrangement of beacons is the same for mapping and subsequent localisation attempts, while ◊ shows the case when there has been an intervening change

However, Figure 8.7 shows that while the robot may still be as certain as ever about where it is, its estimates are now very much in error. The diabolical beacon shift means that for strict matching, where the actual intensities received matter, the conditional expectation and MAP position (not shown) are very poor estimators of position. This is not surprising, as the shift effectively forces the robot to be wrong, by dissociating the beacon with its previous position. As match freedom increases, the actual intensities become indistinguishable, so that in the limit the sonic system always estimates the robot's position to be somewhere near the middle of the pen, and it then fares no worse that it did when the beacon arrangement was fixed.

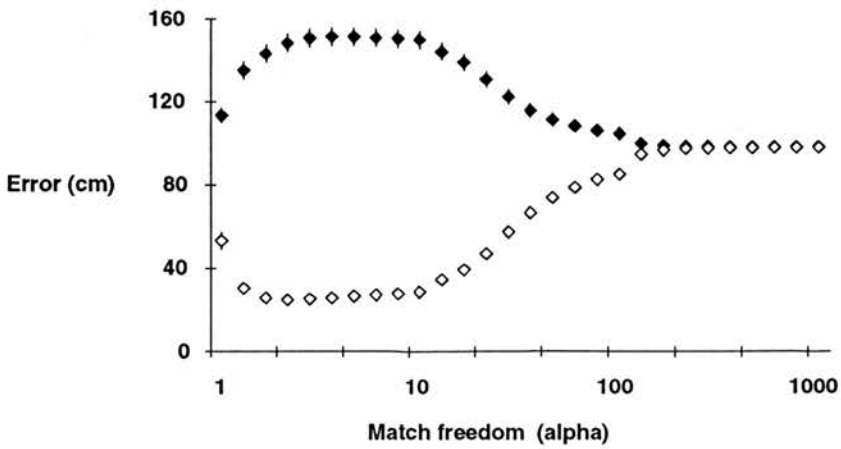

Figure 8.7. Comparison of localisation accuracy by three sonic beacons, ♦ with and ◊ without beacon shifts between mapping and subsequent visits

We would expect a similar effect if the infrared beacons where shifted around. For the sonar, we can test the stability of the system simply by removing the set of boxes usually present in the centre of the experimental pen between mapping and subsequent localisation attempts. This test is appropriate because in a typical operating environment, the localisation system should be able to rely on the walls staying in place, even though other objects may migrate freely. The results are as follows, for match strictness $\alpha = 1000$ (values in parentheses are standard errors):

	Boxes remaining between mapping and future visits	Boxes removed between mapping and future visits
Average information in ultrasonic data field (bits)	5.53 (0.14)	4.25 (0.15)
Average position estimate error (cm)	42.9 (3.5)	60.0 (3.6)

It can be seen that removing the obstacles changes the layout such that the robot is subsequently less than half as certain as it was before about "place"; also, it is now nearly 20 cm worse on average at estimating where it is.

In other words, the results of the analyses for the beacon-based systems are heavily dependent on the fixity of the beacons. Similarly, the ultrasonic system suffers if the layout of the environment is perturbed substantially. However, the latter does retain some accuracy and information by virtue of the temporal invariance of features like walls.

We do not consider it unreasonable to demand of a beacon-based positioning system that the beacons remain fixed through time, at least in the short term, which might seem to render the sonar system redundant wherever the robot's environment is subject to the effects of migrating objects. Nevertheless, beacon-based systems are not a panacea, for at least the reasons listed below.

1. For the experimental pen considered here, the sonic and infrared systems were less accurate than the ultrasonic system, even with three beacons each; we have shown that increasing the number of beacons increases the certainty and accuracy of the system, but it might not be feasible in some environments to provide the number of beacons that would be necessary a particular specified degree of accuracy.

2. The failure (or obscuring) of one or more beacons is not a condition from which the robot itself can necessarily recover (it has lost some or all of the currency by

which it operates); the ultrasonic system, on the other hand, can always remap the environment following changes, and is able in any case to retain a considerable level of performance thanks to fixed layout features like walls.

3. It may even be impossible to equip a particular environment with anything beacon-like; but most imaginable environments have some geometrical features (surfaces) that could be mapped by the ultrasonic system.

Therefore, as was suggested in the conclusion to Chapter 7, a parsimonious approach would be to combine a system like the ultrasonic module employed here with a beacon-based system like our infrared one. These devices are very simple, and yet between them afford accurate localisation even in environments that change. For example, suppose that we consider the effects on certainty and accuracy of localisation when, as above, the boxes in the middle of the pen are removed between mapping and subsequent localisation attempts, except that this time, we assume the infrared beacon system is also available. The results are as follows, for match strictness $\alpha = 1000$:

	Boxes remaining between visits	Boxes removed between visits	
		without IR	with IR
Average data field information (bits)	5.53 (0.14)	4.25 (0.15)	5.86 (0.14)
Average position estimate error (cm)	42.9 (3.5)	60.0 (3.6)	43.9 (4.7)

It can be seen that the provision of the stability of the infrared beacon detector has made the ultrasonic system effectively immune to the change in object layout.

8.1.4 Allocentric orientation reference

One of the fundamental assumptions of the localisation analysis was that the robot needed only to estimate where it was in the plane of its operating space, and had other means to determine its orientation relative to some fixed reference direction in that space. Such an orientation reference is provided in our robot reasonably accurately by a fluxgate compass, and over short distances by dead reckoning based on the wheel encoders (see Section 3.2.5).

This world-anchored reference is the sole means by which the robot's orientation can be deduced for the sonic signal amplitude system, as the amplitudes encode no directional information. The infrared system also depends on it: for example, at a given position it may sense a single beacon bearing straight ahead of the robot; this tells it the

relative direction of the beacon, but not the robot's orientation relative to any putative world-centred coordinate system.

The ultrasonic system depends on the orientation reference even more directly: it works by recording the distance to the nearest specular reflector in 16 directions during the mapping phase; subsequently, localisation proceeds by measuring the current pattern of specularities, and comparing these against the mapped patterns for different locations in the environment. Of course, this comparison relies on an orientation reference to keep the directions to specularities in the map in register with the directions of the specularities currently being compared with.

Without this direction reference, it would be necessary to compare the specularity distances in directions 0 to 15 in each mapped pattern in 16 ways against the current data, since it is not known which of the current directions corresponds to what was direction 0 in the map. That is, if the vector of mapped nearest-distances is **m** = (m0, ..., m15), and the current data is (c0, ..., c15), then there must be a component-wise comparison between **m** and (c0, ..., c15), between **m** and (c1, ..., c15, c0), between **m** and (c2, ..., c15, c0, c1), and so on up to the final comparison between **m** and (c15, c0, ..., c14).

As a test of the effect of disabling the directional reference, the ultrasonic system's performance was reanalysed as follows: the 16 "rotations" of the current pattern were compared against the mapped patterns for each position, and the best match of the 16 was defined to be the likelihood of the robot's being at that position. Thus, we have position likelihood distributions as before, and the usual analytical measures can be applied to them. Figure 8.8 shows the information in the ultrasonic data field for the customary range of match strictnesses (α) from 1 to 1000.

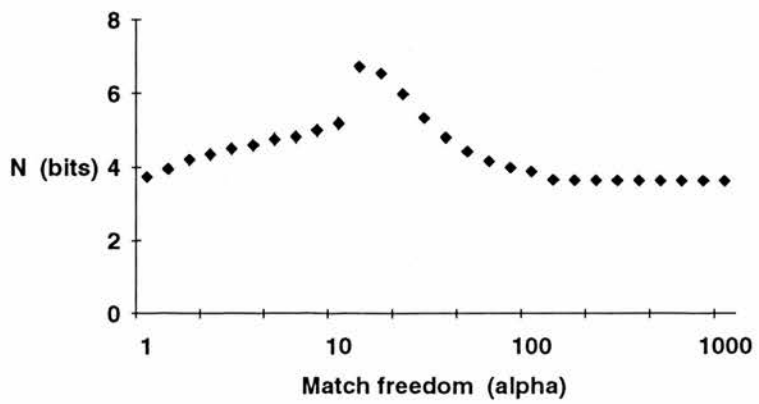

Figure 8.8. Information in the ultrasonic field without directional reference

Comparing these results with Figure 6.34 (information with directional reference), we see that the system is now on average 4 times (2 bits) less certain about its position estimates than it was with the orientation reference, except near α equal to 20 or 30. The high certainty at this match freedom does not actually mean that the system's estimates are accurate, however, as can be seen in Figure 8.9. The figure shows the average position estimate errors for the usual range of match freedom values.

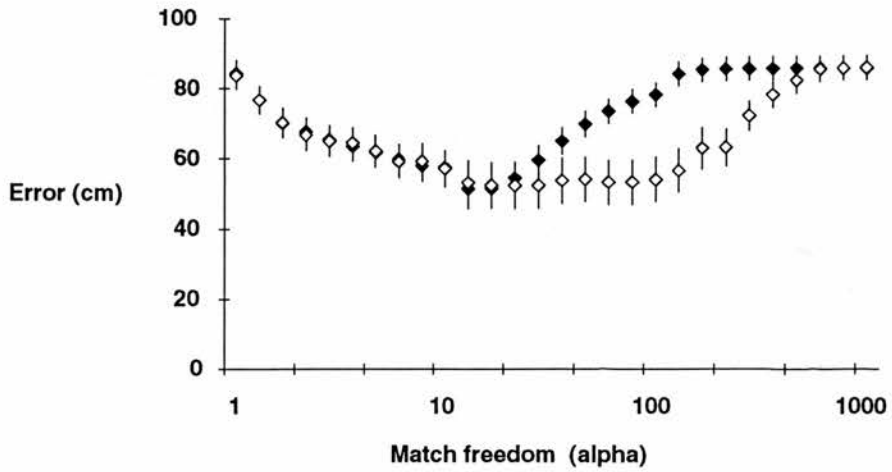

Figure 8.9. Accuracy of localisation by ultrasonic system without directional reference; as before, ♦ marks conditional expectation as an estimator, and ◊ denotes the MAP estimator

As before, the trends are that conditional expectation and MAP estimator are about equally good for $\alpha < 10$, that the conditional expectation worsens during α's second decade while the MAP stays quite good, and that both equalise again as α reaches 1000 (maximum match freedom). Notice that the standard errors in the MAP estimator are large, indicating that the accuracy of the MAP estimator is quite variable across the pen.

In any case, it can be seen that for the best choice of match freedom, the ultrasonic system is able on average to estimate the robot's position to within about 50 cm of where it actually is, and even for maximum match freedom fares better than the sonic system and not a lot worse than the infrared beacon detector under the same circumstances.

Most interestingly, however, the system is able to estimate the robot's orientation relative to the map data. For the simple test discussed here, the robot when collecting the "current" data was always aligned in the same direction as it was during the mapping process. Therefore, the best match between mapped and current data should always occur when **m** is component-wise compared with (c0, ..., c15). In other words, the robot should always estimate its orientation to be 0.

Figure 8.10 plots the ultrasonic system's estimates of the robot's orientation for three decades of match freedom, averaged across all 166 positions. As this average is close to 0 for all α, it is clear that the system estimates the robot's orientation correctly almost everywhere in the pen.

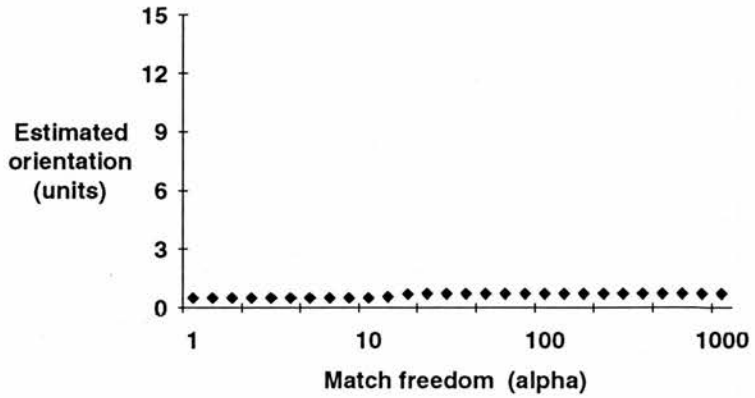

Figure 8.10. Average orientation estimates; the actual orientation was always 0

Therefore, the ultrasonic first-reflection pattern is a usable basis for localisation even without an absolute orientation reference. In fact, it is able to deliver both position estimates in the plane, and estimates of the robot's orientation relative to its orientation when it constructed its ultrasonic map.

8.1.5 Non-uniform prior distributions

A final fundamental assumption of the analysis was that each time the robot tries to establish where it is, it assumes that it might actually be anywhere. In terms of the probabilistic treatment of position, this assumption is implemented by using a prior distribution that is uniform in generating the position likelihood distributions. That is, we have used $P(A_i|B)$, $i = 1,\ldots,n$ to represent the probabilities that the robot is at each of the n positions A_i, given that sensed parameter $\tilde{B}$ currently has value B, where $P(A_i|B)$ is defined by Bayes' Rule

$$P(\tilde{A} = A_j|\tilde{B} = B) = \frac{P(\tilde{B} = B|\tilde{A} = A_j)P(\tilde{A} = A_j)}{\sum_{i=1}^{n} P(\tilde{B} = B|\tilde{A} = A_i)P(\tilde{A} = A_i)}, \qquad 1 \le j \le n.$$

The distribution $P(A_i)$, $i = 1,\ldots,n$ represents the *a priori* probabilities that the robot finds itself at each of the n positions, and we assumed the worst case, that the robot might be anywhere within its operating space, so $P(A_i) = 1/n$, $i = 1,\ldots,n$.

However, this is a pessimistic assumption. Once the robot has decided where it might be, it makes sense to use this distribution to shape the prior distribution for the next localisation attempt. For example, suppose the robot has determined that it is equally likely to be at positions A, B, or C in Figure 8.11. If it then moves by one grid-step, it is likely to be at one of the A', B', or C' positions. Therefore, when matching sensor data after this step, it need not assume that it could be anywhere, as it knows it has a higher a priori probability of being at one of the primed positions.

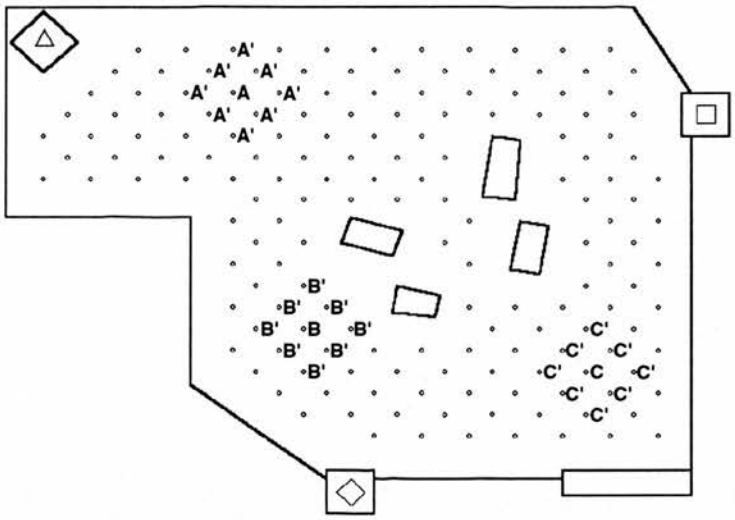

Figure 8.11. Example of non-uniform priors: if the robot might be at A, B, or C at one step, and then moves, it is probably at one of the A', B', or C' positions at the next step

As a simple test of this, we start the robot at the position marked A in Figure 8.11, and let it move towards the right along the straight line of sampling positions between A and the □-marked beacon. At the first step, it assumes it could be anywhere (uniform priors). From then on, the prior probability of being at a given position is proportional to the likelihood that it was at an adjacent position at the previous step. In particular, the robot finds for each position the set of probabilities that it was at an adjacent position at the last localisation attempt. Then, it initially sets the prior for this position to be the maximum of these. Finally, the priors for each position are normalised.

Along the stretch to be tested, the robot follows a low bookshelf. This feature was a source of confusion for the sonic system, so it seems appropriate to test whether non-uniform priors will benefit this system alone. Figure 8.12 plots the distance between the conditional expectation and the robot's actual position as it moves along, both for uniform priors, and for the basic tracking method described above. It is clear that for the uniform priors, the performance of the system is quite erratic, as expected.

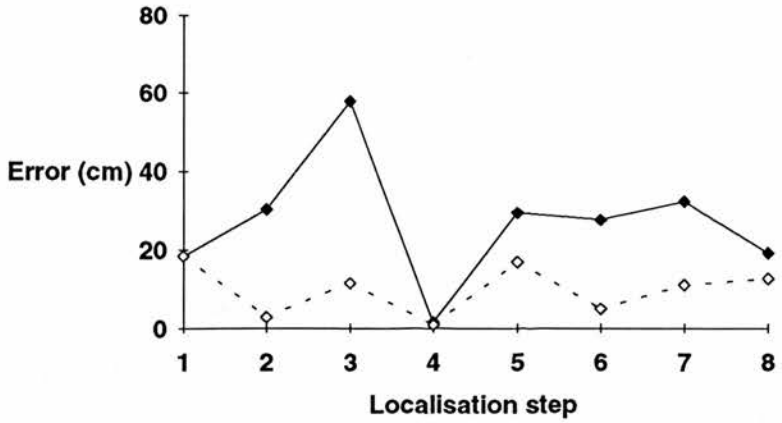

Figure 8.12. Difference between conditional expectation and actual position for a straight line motion in the pen, ♦ denoting uniform priors, and ◊ adaptive priors

However, by adjusting the prior distributions to account for the differential likelihood of being at each position after the first localisation step, the system is able to estimate its position quite accurately at the next step. Of course, the same confusing data that cause the system to estimate position incorrectly with uniform priors at the third step also affect the adaptive-prior version adversely; however, the effect is much less damaging. In fact, the system's position estimates are generally well within 20 cm of the robot's actual position

8.1.6 Summary

In summary, then, we have seen that:

1. while an exponential function is the best choice for matching mapped and current data, the precise form of the function is not critical, and stricter or looser variants on the function used to generate the results in Chapters 6 and 7 lead to the same conclusions;

2. decomposing the robot's operating space into a mesh of discrete sampling locations for the analysis is a reasonable step to take, and even if it constructs a qualitative map in this way, it is still able to localise at off-grid positions;

3. fixity of beacons and environment is not necessary if a beacon-based system like the infrared beacon detector is combined with something like the ultrasonic module which maps and localises by the geometry of the environment alone;

4. a direction reference fixed to the world is not necessary for the ultrasonic system, which can estimate current orientation relative to orientation during mapping quite well; and

5. forcing localisation systems to assume that the robot might be anywhere each time they try to establish where it is (uniform priors) is more likely to degrade their performance than to improve it.

Thus, none of the fundamental assumptions of the localisation analysis appears to have any major consequences, which, if it were necessary not to make these assumptions, would undermine the results compiled in Chapters 6 and 7.

8.2 Sensor system combination topics

The localisation procedures proposed and analysed in Chapters 5 to 7 were not meant to include an elaborate method for combining data from various sensory systems. Instead, the purpose of the exercise was to treat the various systems as simple qualitative devices, and to establish the limitations on localisation inherent in the various sensory data fields. Sensor data was combined in a manner shown to be equivalent to composing the various qualitative maps into a single, multidimensional whole.

However, the bare multiplicative combinational scheme adopted could potentially be improved in at least two different ways. The first is to provide a means by which more reliable systems contribute more to the ultimate combined position estimate than less reliable ones. The second is to introduce algorithmic or rule-based mechanisms for detecting situations in which, say, a particular system's estimates are judged to be unreliable *a priori*, and are therefore not included in a combined estimate. These two topics are addressed below.

8.2.1 Variance-weighted combination

In forming a combined position estimate, all three sensory systems were treated uniformly, so that each contributed an equal proportion to the final composed estimate of robot position. This approach presupposes that none of the three is more or less reliable than the other two.

Since this supposition is unlikely to be correct, it is customary in sensor fusion enterprises to weight the position estimates of a particular system by the inverse of the variance of its estimates. This is one of the reasons why we attempted to characterise the trustworthiness of the conditional variances in position estimate distributions in Chapters 6 and 7, based on our omniscient knowledge of how well localisation systems actually did perform. If a particular system considers itself to have a smaller conditional variance than its actual performance warrants (which often happened when α was small), the weighted-combination procedure suffers, because that system would be trusted more (weighted more heavily) than it actually deserved.

On the other hand, we know from the results in Chapters 6 and 7 that higher certainty generally coincides with smaller estimate errors. Therefore, an alternative approach to combination would be to weight a system's estimates by the certainty of those estimates. Comparing these and other options is left for future work.

8.2.2 Algorithmic methods

Somewhat orthogonal to the largely mathematical improvements to combination described above are algorithmic or rule-based mechanisms. For example, one of the principal advantages of a multisensory approach is that it is potentially more robust in the face of technical failure: should a particular system fail to obtain any data, then it must only return a flat position estimate distribution, thereby not contributing (positively or negatively) to any putative combined position estimate.

It is easy to imagine how more subtle "rules" could also be implemented, to the potential advantage of the robot. For instance, suppose that the position estimates of the ultrasonic system are found to be unreliable whenever the robot is in the field of view of a particular infrared beacon; then, the ultrasonic system could be accorded less credence under these circumstances. While relationships like this could be pre-programmed as part a localisation competence, it would be much more appealing to equip the system with mechanisms for detecting and exploiting such associations itself, i.e., to "learn" them somehow.

8.3 Mapping/learning questions

In the context of the work described here, designing a robot's competences is an exercise for the roboticist. Others have attempted instead to contrive conditions under

which the robot could acquire some of its basic competences on its own. For example, a robot can be programmed to execute random actions, and to observe the effects of particular actions on its sensory inputs; if it is designed so as to prefer not to have contact with other objects, it can discover that turning left avoids contact on the right, and vice versa, and thus "learns" how to avoid obstacles (see, e.g., Nehmzow *et al.*, 1989).

We believe that the general structure of such basic skills ought to be engineered-in to a robot's basic behavioural repertoire from its inception. Nevertheless, there is clearly a role for mechanisms that allow a robot's competences to be adapted automatically to its individual operational circumstances. Developing schemes for implementing behavioural plasticity is of widespread interest in robotics research; we consider this aspect of competence design to be the most important direction in which the work described here could be elaborated, for both locomotion and spatial capabilities.

8.3.1 Learning and locomotor competences

In particular, it was already discussed in Chapter 4 that the smoothness and efficiency of locomotion to some specified goal benefits from some means for anticipating the density of objects cluttering the path ahead. A simple way to implement this would be to associate some density measure with location, supplied by the robot's spatial sense.

For instance, if the vehicle had a qualitative map in which its operating space was segmented into n regions, it would need only to store an obstacle density (or corresponding maximum speed to travel) with each region: just a list of n numbers. If on subsequent visits to some region k the robot determined that the density of obstacles had changed (e.g., increased), it need only modify the entry in the list for region k to record the new density or corresponding optimal speed.

Furthermore, the pattern of object densities is itself a kind of qualitative spatial map. That is, suppose the robot's operating environment has a region of open space, and area with a few migrating objects, and a third area with many potential obstructions; the object density is now a simple device for discriminating between these three regions, and therefore is the basis for localisation. Even if this is very restricted spatial awareness, it allows the robot to go some way towards guessing where it is should all other positioning systems available to it fail.

8.3.2 Map acquisition and matching

The process of map acquisition as implemented here consisted in programming the robot to follow a fixed trajectory known to cover the floor of the lab, sampling the data fields of its various sensory systems at regular intervals. This procedure defines space according to sampled sensory conditions at a set of discrete locations. A compelling alternative is to sense continuously, and to define space in terms of transitions between sensory states. Both of these map construction techniques might be called "learning" in the sense that the robot ends up in either case with an association between sensed data and location. However, it could be argued that the former is basically a mechanical process of data collection, and it is difficult to describe acquiring a sensor data sample as "discovery": the pre-programming of trajectory and sampling positions makes map building more of an "assignment" operation (in the programming language sense) than an act of association.

The latter technique, on the other hand, has the robot travelling around its operating environment without preconceptions as to where significant sensory events will occur; therefore, map building includes an element of exploration, leading to a definition of space that is peculiar to the robot, the exact physical characteristics of its sensory systems, and the nature of its particular operational territory. Therefore, we would expect to implement map acquisition in a such a dynamic manner before being willing to describe it as discovering or learning anything about space.

Still, there is one obvious way in which the current approach to map building would benefit from adaptive mechanisms, namely in choosing an appropriate setting for the match freedom control parameter, α. We have seen in Chapters 6 and 7 that the most accurate localisation for the sonic and infrared systems occurred for α quite small, perhaps between 1 and 4; for the ultrasonic system, the best results were obtained for α about 30. The optimal choce for α is determined by the inherent variability in the sensor data.

For the ultrasonic system, this variability is much greater than for the sonic or infrared systems, so that localisation is better when matching between mapped and current data is less strict. Therefore, a simple but valuable feature would be for the robot, once it has collected sensor data from more than one visit to the same location, to track this inherent variability between past and present data samples for each sensor system, and to adjust the α it is uses for each system accordingly.

8.3.3 Learning mechanisms and qualitative maps

One of the principal advantages of a qualitative segmentation of space is that this kind of map seems intuitively to be well matched to distributed representations such as might be afforded by a neural net. For example, a network could be constructed of units receiving input from the various sensors, in such a way that particular units or groups of units could learn to fire in response to distinct combinations of sensory inputs. A characteristic pattern of network activity would then be associated with the robot's being at a certain place in its environment. Such "place" cells that are active only when an animal is at a particular location have been discovered in the rat hippocampus, and may be the basis for a distributed representation of space (O'Keefe and Dostrovsky, 1971; O'Keefe, 1976). In any case, once an appropriate distributed representation has been conceived, it is subsequently possible and convenient to draw upon the established spectrum of mechanisms that allow such representations to be learned.

8.4 Localisation and navigation

Ultimately, the purpose of maps and the ability to localise in space by means of them is to underpin navigation (the ability to get from one place to another). This is the final and most obvious enhancement to the work discussed here. In fact, the robot described in Chapter 3 is able to navigate in two ways. First of all, as discussed in Chapter 4, it can get from any place in the experimental pen to any of the three beacons by means of a simple tropism (if a particular beacon is not visible, it follows a series of random straight line trajectories until it enters the beacon's field of view, and then is free to approach it).

Also, given the fixed trajectory mapping scheme adopted for the localisation competence, it is an easy matter to store the spatial relationships between the 166 sampling positions in parallel with the sensor data maps. Once the robot has localised at one location, therefore, determining how to get to another is elementary geometry. This method improves upon the tropism in that the latter only allows the robot to approach positions at which there is a beacon.

However, if we are to liberate mapping from the limitations of pre-programmed trajectory sampling and to adopt a distributed representation such as described in the foregoing section, navigation is no longer basic geometry. For example, Nehmzow's

(1990) robot successfully builds maps of its operating space by recognising and remembering the arrangement of corners in walls, but this spatial representation does not permit it to resolve and trace a trajectory between two arbitrary locations.

Some neural-net-based mapping and navigational schemes have been proposed (e.g., Barto and Sutton, 1981), although most have only been demonstrated as part of computer simulations. It would be interesting to attempt to validate some of these approaches on a real robot; this would test whether the simulation assumptions generally made about the fidelity of sensor data, and the response of actual robotic platforms to motor commands, are sufficiently realistic, in which case adaptive spatial competences could be developed from them. There is considerable scope for future work in this direction.

Epilogue

This dissertation documents an investigation into the design of locomotor and spatial competences for an autonomous robotic vehicle. The fundamental behaviours such a device must be capable of displaying are primarily locomotor and spatial: in general, it must be able to move around without colliding with objects in its vicinity, and it must have some mechanism for determining where it is and how to get to other places. The academic robotics community has in recent years been divided into two distinct schools of thought with regard to questions of how to design such an agent.

The more traditional of the two groups founds robot control systems on explicit representations of the robot's relationship to its environment, and by *analysing* this relationship the robot plans a sequence of actions that achieve some desired end. These approaches were called *analytic*, for this and other reasons. On the other hand, the second school attempts to avoid explicit representations and their manipulation, believing instead that empirical insights derived from bottom-up *synthesis* of the system can point the way to an arrangement of control elements from which appropriate robot behaviour arises as a reaction principally to the present relationship between robot and world.

The guiding philosophies of these two groups have been compared and contrasted in terms of their respective approaches to locomotion and localisation. The starting premise of the work presented here was that locomotion was likely to fit well with a representation-free (or synthetic) style, but that accurate spatial skills might remain the preserve of the model and representation-based (or analytic) style. In fact, this intuition was not borne out. The following conclusions obtain from the work:

1. Smooth goal-directed locomotion works better predictively than reactively.

To begin with, it was seen that the smoothness of a mobile robot's locomotion benefits from adjusting its speed to the density of obstacles it encounters as it travels around. In regions of great clutter, it moves most efficiently if it slows down, whereas it is free to travel quickly in open spaces. In Chapter 4, a reactive, synthetic-type procedure for achieving this was presented. The scheme was based on the frequency at which

obstacle detection interrupted the robot's navigational system. High obstacle detection frequencies were suggested to be an analogue for high obstacle density, and therefore the robot's speed was controlled such that it varied inversely with this frequency.

However, it was also shown that better performance obtained from either longer-range sensors, or a predictive, representation-based strategy. In either case, the robot was able to anticipate density changes and adjust its speed appropriately in advance, rather than just reacting to instantaneous obstacle density.

2. Qualitative localisation can be accurate even without robust models.

A simple qualitative representation of the robot's operating environment was proposed, based only on the goodness of match between previously mapped sensor data and current inputs. The sensors of a prototype mobile robot are treated as devices which partition its operating space qualitatively according to the sensor readings themselves, without any attempt to obtain from these measurements any information about the geometry of the robot's environment.

As with other qualitative spatial schemes discussed in the literature, the advantage of this approach is that it sidesteps the need for robust models of sensors. Unlike existing qualitative systems, however, the method proposed here does not restrict the robot to operating only along fixed boundaries within its environment, like walls. Furthermore, the work attempts to borrow some numerical tools from the quantitative designs. In a sense, it tries to be somewhat analytic about an otherwise synthetic operation.

Three of the robot's sensor systems were studied in the proposed framework. They were the sonic signal amplitude subsystem, the infrared beacon detector, and the ultrasonic module. Their respective utility to the robot in estimating its location were assessed, with the robot operating in a realistic (relatively uncontrived) laboratory setting. By means of a number of metrics from probability and information theory, sensory systems and combinations of systems were judged in terms of how accurately and confidently the robot was able to localise with them. Combining the estimates of the infrared and ultrasonic systems, in particular, led to position estimates accurate to within 10 cm (on average) of where the robot actually was.

The work shows that it is possible to go a long way towards a robust spatial competence with inexpensive sensors, naively modelled, and very basic procedures for combining them. It does not, of course, deny that if good models and ways of using

them are available, many advantages are conferred in so doing. In a sense, this qualitative investigation is a kind of modelling process, which could precede the implementation of a more analytical approach.

3. Accurate sensors with dead zones are preferable to inaccurate ones which can be used anywhere.

It was found that the sonic system, while able to operate everywhere within the environment, was not particularly useful for localisation. In contrast, both the infrared and ultrasonic systems worked very well, although both have regions of inutility (the infrared beacons are not visible universally, and the sonar system does not produce useful data when the robot is close to walls). It was suggested that these systems work well partly *because* of this natural segmentation of their data fields, rather than in spite of it.

4. Analysis and synthesis could be combined.

In summary, the major observations made in the course of this work are that (a) even for a relatively simple locomotor task (smooth goal-directed locomotion), a representation-free solution may lead to robust but not necessarily efficient behaviour, and that (b) qualitative localisation by means of a variety of sensor systems can be accomplished *without* extensive representations and sensor models, but still admits of quantitative performance characterisation (by means of statistical tests on map data).

Since these results are somewhat contrary to intuition, this suggests that deciding which aspects of robot behaviour are well-suited, and which are unsuited for representation versus representation-free implementation is not trivial. Thus, it seems fair to suggest that a combined approach, incorporating elements of both analysis- and synthesis-based philosophies is a potentially fruitful path for future exploration.

Appendix A

Robot Programming Language Syntax

The syntax of the object-based robot programming system is essentially that of Pascal, augmented by the object-oriented elements of Turbo Pascal® version 5.5 and later. These are documented in the language manual (Borland, 1988), but the basics are repeated in this appendix in order to explain the examples shown in Chapter 3.

Object types

Essentially, an "object" is a data type which combines records and procedures for manipulating them. Therefore, objects are added to conventional Pascal as a new kind of type, defined by:

```
type ObjectName = object (InheritedObject)
         { variables }
         { methods: procedures and functions }
       end;
```

The `(Inherited Object)` part is omitted if an object has no ancestors. Once the form of an object has been defined in this way, it is necessary to declare the implementation of its methods. For example, suppose that the following object has been defined:

```
type Point = object
         x, y: integer;
         procedure SetCoords (newx, newy: integer);
         procedure Plot;
       end;
```

This `Point` is an object having two internal variables (`x` and `y`), and a two methods, `SetCoords` and `Plot`. These methods are declared after the type definition, perhaps as:

```
procedure Point.SetCoords (newx, newy: integer);
  begin
    x := newx;
    y := newy;
  end;
```

```
procedure Point.Plot;
  begin
    PutPixel(x, y, white);
  end;
```

In other words, the first method tells the object to take on new values for its two local variables, and the second does something trivial like drawing a white dot at the coordinates (x, y).

An "object type" is not yet an object, just as an "integer type" is not an integer. The object itself comes into being only when a variable of type "object type" has been declared. For the above example, this would be done by:

```
var PointObject: Point;
```

This effects the transformation of the object from a syntactic form to an entity that occupies space in the processor's memory. It is then possible in the scope of the `PointObject` variable to make statements like the following:

```
begin
  PointObject.SetCoords(120, 315);
  PointObject.Plot;
end.
```

This piece of code can be thought of as saying to `PointObject` "set your coordinates to be (120, 315)" and then "show your coordinates as a white dot".

Inheritance

If one object has some characteristics in common with another, even if there are also differences, it is appropriate for the former to be defined as a "descendant" of the latter; that is, it inherits all of the latter's variables and methods. For example, suppose that we have:

```
type Circle = object (Point)
         radius: integer;
         procedure SetRadius (newr: integer);
         procedure Draw;
       end;
```

This is effectively identical to defining an entirely new object `Circle` as:

```
type Circle = object
         x, y: integer;
```

```
        radius: integer;
        procedure SetCoords (newx, newy: integer);
        procedure SetRadius (newr: integer);
        procedure Plot;
        procedure Draw;
      end;
```

However, the inheritance mechanism makes it unnecessary to declare implementations for `Circle.SetCoords`, or `Circle.Plot`, as these will be assumed to be identical to `Point.SetCoords` and `Point.Plot`. Only the new methods need be defined, perhaps as:

```
procedure Circle.SetRadius (newr: integer);
  begin
    radius := newr;
  end;

procedure Circle.Draw;
  begin
    PutCircle(x, y, radius, white);
  end;
```

These declarations mean that the original coordinate pair defined for a `Point` object have actually become the centre of a new `Circle` object, which adds a new local "radius" variable and a method for setting it, as well as a method that draws the circle the object is meant to represent.

Polymorphism

In the previous example, one thing to notice is that the new `Circle` object now has a redundant method `Circle.Plot` that draws a dot at the coordinates that are the circle's centre. One of the principal features of the object-oriented approach is that this kind of redundancy can be eliminated if the original `Plot` routine is redefined in such a way that the `Circle` object can replace it by a more appropriate routine. Consider the following new versions of `Point` and `Circle`:

```
type Point = object
        x, y: integer;
        constructor SetCoords (newx, newy: integer);
        procedure Plot; virtual;
      end;
     Circle = object (Point)
        radius: integer;
        procedure SetRadius (newr: integer);
        procedure Plot; virtual;
      end;
```

```
constructor Point.SetCoords (newx, newy: integer);
  begin
    x := newx;
    y := newy;
  end;

procedure Point.Plot;
  begin
    PutPixel(x, y, white);
  end;

procedure Circle.SetRadius (newr: integer);
  begin
    radius := newr;
  end;

procedure Circle.Plot;
  begin
    PutCircle(x, y, radius, white);
  end;
```

The principal change is that `Point.Plot` has now been defined as a "virtual" method. This means that when another object inherits from Point, it can redefine the Plot method to be something new. Therefore, the following is valid code:

```
var Spot: Point;
    Ring: Circle;

begin
  Spot.SetCoords(120, 315);
  Spot.Plot;
  Circle.SetCoords(120, 315);
  Circle.SetRadius(40);
  Circle.Plot;
end.
```

The `Spot.Plot` will use the original `Plot` method from `Point` to draw a pixel at coordinates (120, 315). However, `Circle.Plot` will use the new `Plot` method defined locally within the `Circle` object. Note that the `SetCoords` method has been defined as something called a "constructor". This is a technical requirement: since the appropriate version of a virtual method is bound to its name only at run-time, some initialisation operations are necessary before an object is first used. Rather than implementing some mechanism for detecting the first use of an object and then initialising it, Turbo Pascal® requires that one of the object's methods be called a "constructor", and that this method be called before any other.

Constructors are necessary whenever an object uses virtual methods, or if an object is to be declared not as a static variable (as in the examples above), but dynamically in heap memory. There is also a "destructor" mechanism which is only necessary in the latter case.

In summary, then, the syntax of the robot's programs and the examples in Chapter 3 is basically Pascal plus the Turbo Pascal® version 5.5 object extensions:

```
ObjectType  ::=  'object' [ Inheritance ]
                    [ Variables ] [ ';' Methods ] 'end'

Inheritance ::=  '(' ObjectTypeIdentifier ')'

Variables   ::=  VariableIdentifier : TypeIdentifier
                    [ ';' Variables ]

Methods     ::=  MethodType [ '; virtual' ] [ ';' Methods ]

MethodType  ::=  ( 'constructor' [ Params ] |
                   'destructor' [ Params ] |
                   'procedure' [ Params ] |
                   'function' [ Params ] ':' TypeIdentifier )

Params      ::=  '(' Variables ')'
```

Appendix B

Kinematic Model

Muir and Newman (1987a,b) propose that wheeled robots be modelled as multiple closed-link kinematic chains. The tractability of their approach depends upon several assumptions: the robot must be rigid, have at most one steering link per wheel, and have steering axes perpendicular to the floor. These assumptions are not unreasonable. However, Muir and Newman additionally require that the surface on which the robot moves be smooth and planar, that no slippage occur between the wheels and floor, and that a wheel be able to rotate about a vertical axis through the point of contact between it and the floor. These assumptions are not likely to be reasonable in practice, and their almost inevitable violation will lead to errors in the robot's dead-reckoned estimates of position.

The links of a wheeled robot are the floor, the wheel mounting structure, and the body. Muir and Newman use a convention for assigning coordinate frames where each link has a coordinate frame at both ends; i.e., there are two coordinate frames per joint. The joints are: the wheel, between the floor and the wheel mounting structure, modelled as a "planar pair" (two degrees of translation and one of rotation); the hip or steering axis, between the wheel mounting structure and the body, modelled as a revolute pair; and the robot mid-point, a "virtual" joint that is actually just the relationship between the body and the floor, modelled as a planar pair at the centre of the robot.

Each wheel has three links, three joints, and six coordinate frames. Using the notation of McKerrow (1991), we denote these as CF, CL, SL, SB, RB, and RF. Frame CF is anchored to the floor, but considered at any instant coincident with CL, which is attached to the wheel where it contacts the floor. This CF is an *instantaneously coincident coordinate system*, introduced for convenience, so that velocities and accelerations of the moving wheel can be calculated without reference to the vehicle's location. Frame SB is fixed to the body where the wheel mounting structure is attached (the "hip joint"); SL is attached to the steering link; the robot frame RB is at the centre of the robot; and RF is a

further instantaneous frame that allows the velocities and accelerations of the whole robot to be computed independently of its position. The floor frame F is a stationary reference frame for robot motion. Figure B.1 depicts in these terms the positions and relationships between the coordinate frames on the robot described in Chapter 3.

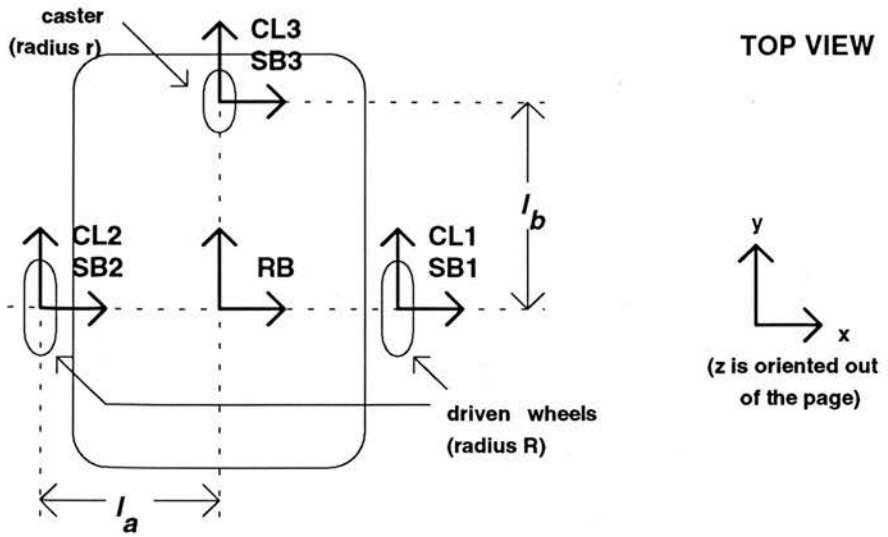

Figure B.1. Schematic representation of the robot, showing the positions of and relationships between the coordinate frames

In addition to l_b, the distance along the y-axis between the centres of the driven axles and the centre of the caster, and l_a, the distance along the x-axis between the two driven wheels, there is a third dimension, l_c. This is the height above the floor of the origin of the robot-centre frame RB. For the robot in Chapter 3, these dimensions are l_a = 101 mm, l_b = 127 mm, and l_c = 170 mm. The radius of the driven wheels, R, is 41 mm, and of the caster (r) is 25 mm.

All frames are oriented with the z axis pointing upwards, and since the robot is assumed to be moving on a horizontal plane, any rotations of the robot can be about z axes only. A standard transform between any two frames M and N is therefore

$$ {}^{M}\mathbf{T}_{N} = \begin{bmatrix} \cos{}^{M}\theta_{N} & -\sin{}^{M}\theta_{N} & 0 & p_x \\ \sin{}^{M}\theta_{N} & \cos{}^{M}\theta_{N} & 0 & p_y \\ 0 & 0 & 1 & p_x \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

where ${}^{M}\theta_{N}$ is the angle between the frames. The position of any point p, on the robot

can be computed for a new frame, N, given its specification in a reference frame, M, simply by

$$^{M}\mathbf{p} = {}^{M}\mathbf{T}_{N}\,{}^{N}\mathbf{p},$$

where $\mathbf{p}$ is the position vector of p.

The transform equations between the various frames for the three wheels then become

$$\begin{aligned} {}^{RF}\mathbf{T}_{RB} &= {}^{RB}\mathbf{T}_{SB1}\,{}^{SB1}\mathbf{T}_{SL1}\,{}^{SL1}\mathbf{T}_{CL1}\,{}^{CL1}\mathbf{T}_{CF1}\,{}^{CF1}\mathbf{T}_{F}\,{}^{F}\mathbf{T}_{RF} \\ &= {}^{RB}\mathbf{T}_{SB2}\,{}^{SB2}\mathbf{T}_{SL2}\,{}^{SL2}\mathbf{T}_{CL2}\,{}^{CL2}\mathbf{T}_{CF2}\,{}^{CF2}\mathbf{T}_{F}\,{}^{F}\mathbf{T}_{RF} \\ &= {}^{RB}\mathbf{T}_{SB3}\,{}^{SB3}\mathbf{T}_{SL3}\,{}^{SL3}\mathbf{T}_{CL3}\,{}^{CL3}\mathbf{T}_{CF3}\,{}^{CF3}\mathbf{T}_{F}\,{}^{F}\mathbf{T}_{RF}. \end{aligned}$$

Our robot does not have a steering link, so that some of the transformation matrices are actually equal to the identity matrix. In particular,

$$^{SB1}\mathbf{T}_{SL1} = {}^{SL1}\mathbf{T}_{CL1} = {}^{SB2}\mathbf{T}_{SL2} = {}^{SL2}\mathbf{T}_{CL2} = {}^{SB3}\mathbf{T}_{SL3} = {}^{SL3}\mathbf{T}_{CL3} = \mathbf{I},$$

so that

$$^{RB}\mathbf{T}_{SBn}\,{}^{SBn}\mathbf{T}_{SLn}\,{}^{SLn}\mathbf{T}_{CLn} = {}^{RB}\mathbf{T}_{CLn}, \quad n = 1,\ldots,3.$$

For a robot configured like ours, we have

$$^{RB}\mathbf{T}_{CL1} = \begin{bmatrix} 1 & 0 & 0 & l_a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$^{RB}\mathbf{T}_{CL2} = \begin{bmatrix} 1 & 0 & 0 & -l_a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and}$$

$$^{RB}\mathbf{T}_{CL3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_b \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Instantaneous Robot Velocity

The instantaneous velocity of the robot with respect to the coincident floor frame is related to the velocity vector for each wheel by a Jacobian matrix defined for the wheel. That is,

$$^{RF}\mathbf{v}_{RB} = \mathbf{J}_n \mathbf{v}_n = \begin{bmatrix} v_{xRB} & v_{yRB} & \omega_{RB} \end{bmatrix}^T, \tag{1}$$

where $n = 1,\ldots,3$, $\mathbf{J}_n$ is the Jacobian matrix for wheel n, $\mathbf{v}_n$ its velocity vector, and v_{xRB}, v_{yRB}, and ω_{RB} the x-, y-, and rotational-components of the whole vehicle's velocity. In our robot, wheel 3 is a caster, i.e., is not driven; furthermore, no sensing is provided at this wheel, so we consider only wheels 1 and 2 in the remainder of this appendix.

Muir and Newman (1987a,b) derive the "physical" Jacobian and velocity matrices by means of an intermediate step in which "pseudo" Jacobian matrices and "pseudo" velocity vectors for the wheels are defined; these express the kinematics of the wheel completely, but contain elements that cannot be measured or are redundant, and are therefore related to their "physical" analogues by "wheel matrices" ($\mathbf{W}_n$) that encode the mechanical configuration of each wheel. In particular,

$$\mathbf{v}_{n-\text{pseudo}} = \mathbf{W}_n \mathbf{v}_n, \tag{2}$$

and

$$\mathbf{J}_n = \mathbf{J}_{n-\text{pseudo}} \mathbf{W}_n. \tag{3}$$

The pseudo-velocity matrix is defined for n = 1, 2 as

$$\mathbf{v}_{n-\text{pseudo}} = \begin{bmatrix} ^{CF}v_{xCLn} \\ ^{CF}v_{yCLn} \\ ^{CF}\omega_{CLn} \\ ^{SB}\omega_{CLn} \end{bmatrix},$$

where $^{CF}\mathbf{v}_{xCLn}$ is the instantaneous linear velocity of wheel n with respect to the floor, $^{CF}\omega_{CLn}$ is the instantaneous angular velocity of wheel n about the contact point between it and the floor, and $^{SB}\omega_{CLn}$ is the angular velocity of the steering link about the hip joint. In our case, therefore,

$$\mathbf{v}_{1-\text{pseudo}} = \begin{bmatrix} 0 \\ {}^{\text{CF}}v_{y\text{CL1}} \\ {}^{\text{CF}}\omega_{CL1} \\ 0 \end{bmatrix}, \text{ and } \mathbf{v}_{2-\text{pseudo}} = \begin{bmatrix} 0 \\ {}^{\text{CF}}v_{y\text{CL2}} \\ {}^{\text{CF}}\omega_{\text{CL2}} \\ 0 \end{bmatrix}.$$

For the "physical" wheel n, we have only ω_{xn}, the angular velocity of the wheel about its axis, and ω_{zn}, the angular velocity of rotational slip of the wheel about the contact point between it and the floor. Thus,

$$\mathbf{v}_n = \begin{bmatrix} \omega_{xn} \\ \omega_{zn} \end{bmatrix}, \quad n = 1,2.$$

It follows from (2) that

$$\mathbf{W}_n = \begin{bmatrix} 0 & 0 \\ R & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

is the matrix that describes wheel mechanical configuration. For our robot, this is the matrix for both wheel 1 and wheel 2. The pseudo-Jacobian matrix is defined as

$$J_{n-pseudo} = \begin{bmatrix} \cos{}^{\text{RB}}\theta_{\text{CL}n} & -\sin{}^{\text{RB}}\theta_{\text{CL}n} & {}^{\text{RB}}p_{y\text{CL}n} & -{}^{\text{RB}}p_{y\text{SB}n} \\ \sin{}^{\text{RB}}\theta_{\text{CL}n} & \cos{}^{\text{RB}}\theta_{\text{CL}n} & -{}^{\text{RB}}p_{x\text{CL}n} & {}^{\text{RB}}p_{x\text{SB}n} \\ 0 & 0 & 1 & -1 \end{bmatrix},$$

where ${}^{\text{M}}\theta_{\text{N}}$ is the sum of the angles between the frames M and N, so that ${}^{\text{RB}}\theta_{\text{CL}} = {}^{\text{RB}}\theta_{\text{SB}} + {}^{\text{SB}}\theta_{\text{CL}}$, and ${}^{\text{M}}\mathbf{p}_{\text{N}}$ is a vector from the origin of one frame to the origin of the other. For our robot configuration,

$${}^{\text{RB}}\theta_{\text{SB}} = 0, \; {}^{\text{SB}}\theta_{\text{CL}} = 0, \; {}^{\text{RB}}\mathbf{p}_{\text{CL1}} = \begin{bmatrix} l_a \\ 0 \end{bmatrix}, \text{ and } {}^{\text{RB}}\mathbf{p}_{\text{CL2}} = \begin{bmatrix} -l_a \\ 0 \end{bmatrix}.$$

Thus we have,

$$\mathbf{J}_{1-\text{pseudo}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -l_a & l_a \\ 0 & 0 & 1 & -1 \end{bmatrix}, \text{ and } \mathbf{J}_{2-\text{pseudo}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & l_a & -l_a \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

From (3), we have that

$$\begin{aligned}\mathbf{J}_1 &= \mathbf{J}_{1-\text{pseudo}}\mathbf{W}_1 \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -l_a & l_a \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ R & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ R & -l_a \\ 0 & 1 \end{bmatrix},\end{aligned}$$

and

$$\begin{aligned}\mathbf{J}_2 &= \mathbf{J}_{2-\text{pseudo}}\mathbf{W}_2 \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & l_a & -l_a \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ R & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ R & l_a \\ 0 & 1 \end{bmatrix}.\end{aligned}$$

Now, returning to (1) we get

$$\begin{aligned}{}^{\text{RF}}\mathbf{v}_{\text{RB}} &= \begin{bmatrix} v_{x\text{RB}} \\ v_{y\text{RB}} \\ \omega_{\text{RB}} \end{bmatrix} \\ &= \mathbf{J}_1\mathbf{v}_1 \\ &= \begin{bmatrix} 0 & 0 \\ R & -l_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{x1} \\ \omega_{z1} \end{bmatrix} \\ &= \mathbf{J}_2\mathbf{v}_2 \\ &= \begin{bmatrix} 0 & 0 \\ R & l_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{x2} \\ \omega_{z2} \end{bmatrix}.\end{aligned}$$

Finally, it must be noted that this is the velocity with respect to the floor frame instantaneously coincident with the robot. With respect to the fixed frame, we rotate ${}^{RF}\mathbf{v}_{RB}$ through the angle between the robot body frame RB and the fixed frame F,

$$ {}^{F}\mathbf{v}_{RB} = \begin{bmatrix} \cos{}^{F}\theta_{RB} & -\sin{}^{F}\theta_{RB} & 0 \\ \sin{}^{F}\theta_{RB} & \cos{}^{F}\theta_{RB} & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{RF}\mathbf{v}_{RB}. $$

"Actuated" Inverse Kinematics

Muir and Newman define an "actuated" Jacobian matrix that relates the desired robot velocity vector to those velocity components that can actually be controlled, given the electromechanical configuration of the vehicle, as

$$ \mathbf{v}_a = \mathbf{J}_a \mathbf{v}_{RB}, \tag{4} $$

where $\mathbf{v}_a$ is a matrix of actuated wheel velocity components, and $\mathbf{J}_a$ is the actuated-wheel Jacobian. For our robot,

$$ \mathbf{J}_a = \begin{bmatrix} \left[\mathbf{J}_{1a}{}^{T}\Delta(\mathbf{J}_{1u})\mathbf{J}_{1a}\right]^{-1}\mathbf{J}_{1a}{}^{T}\Delta(\mathbf{J}_{1u}) \\ \left[\mathbf{J}_{2a}{}^{T}\Delta(\mathbf{J}_{2u})\mathbf{J}_{2a}\right]^{-1}\mathbf{J}_{2a}{}^{T}\Delta(\mathbf{J}_{2u}) \end{bmatrix}, $$

where

$$ \Delta(\mathbf{U}) = \begin{cases} -\mathbf{I} & \text{for } \mathbf{U} = \text{null} \\ \mathbf{U}(\mathbf{U}^{T}\mathbf{U})^{-1}\mathbf{U}^{T} - \mathbf{I} & \text{otherwise,} \end{cases} $$

and $\mathbf{J}_{na}$ and $\mathbf{J}_{nu}$ are the columns of the wheel Jacobians that correspond to actuated and unactuated velocity components, respectively. The rather complicated expression for $\mathbf{J}_a$ simply serves the purpose of expressing mathematically the act of partitioning and recombining the parts of the original wheel Jacobians $\mathbf{J}_1$ and $\mathbf{J}_2$ derived earlier, so that actuated velocity components are explicitly controlled, and unactuated components are programmed tacitly. In our robot, only the angular velocity of the wheels about their axles can be set (motor speed), i.e., only ω_{x1} and ω_{x2} are actuated. Therefore, we have

$$ \mathbf{J}_{1a} = \mathbf{J}_{2a} = \begin{bmatrix} 0 \\ R \\ 0 \end{bmatrix}, \quad \mathbf{J}_{1u} = \begin{bmatrix} 0 \\ -l_a \\ 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{J}_{2u} = \begin{bmatrix} 0 \\ l_a \\ 1 \end{bmatrix}. $$

Thus,

$$\begin{aligned}\Delta(\mathbf{J}_{1u}) &= \mathbf{J}_{1u}(\mathbf{J}_{1u}{}^{T}\mathbf{J}_{1u})^{-1}\mathbf{J}_{1u}{}^{T} - \mathbf{I} \\ &= \begin{bmatrix} 0 \\ -l_a \\ 1 \end{bmatrix}\left(\begin{bmatrix} 0 & -l_a & 1 \end{bmatrix}\begin{bmatrix} 0 \\ -l_a \\ 1 \end{bmatrix}\right)^{-1}\begin{bmatrix} 0 & -l_a & 1 \end{bmatrix} - \mathbf{I} \\ &= (1+l_a{}^2)^{-1}\begin{bmatrix} 0 & 0 & 0 \\ 0 & l_a{}^2 & -l_a \\ 0 & -l_a & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -\frac{1}{1+l_a{}^2} & -\frac{l_a}{1+l_a{}^2} \\ 0 & -\frac{l_a}{1+l_a{}^2} & -\frac{l_a{}^2}{1+l_a{}^2} \end{bmatrix}.\end{aligned}$$

Now,

$$\begin{aligned}\mathbf{J}_{1a}{}^{T}\Delta(\mathbf{J}_{1u})\mathbf{J}_{1a} &= \begin{bmatrix} 0 & R & 0 \end{bmatrix}\begin{bmatrix} -1 & 0 & 0 \\ 0 & -\frac{1}{1+l_a{}^2} & -\frac{l_a}{1+l_a{}^2} \\ 0 & -\frac{l_a}{1+l_a{}^2} & -\frac{l_a{}^2}{1+l_a{}^2} \end{bmatrix}\begin{bmatrix} 0 \\ R \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -\frac{R}{1+l_a{}^2} & -\frac{Rl_a}{1+l_a{}^2} \end{bmatrix}\begin{bmatrix} 0 \\ R \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -\dfrac{R^2}{1+l_a{}^2} \end{bmatrix},\end{aligned}$$

and,

$$\left[\mathbf{J}_{1a}{}^{T}\Delta(\mathbf{J}_{1u})\mathbf{J}_{1a}\right]^{-1} = -\frac{1+l_a{}^2}{R^2},$$

so,

$$\begin{aligned}\left[\mathbf{J}_{1a}{}^{T}\Delta(\mathbf{J}_{1u})\mathbf{J}_{1a}\right]^{-1}\mathbf{J}_{1a}{}^{T}\Delta(\mathbf{J}_{1u}) &= -\frac{1+l_a{}^2}{R^2}\begin{bmatrix} 0 & R & 0 \end{bmatrix}\Delta(\mathbf{J}_{1u}) \\ &= \begin{bmatrix} 0 & -\frac{1+l_a{}^2}{R} & 0 \end{bmatrix}\begin{bmatrix} -1 & 0 & 0 \\ 0 & -\frac{1}{1+l_a{}^2} & -\frac{l_a}{1+l_a{}^2} \\ 0 & -\frac{l_a}{1+l_a{}^2} & -\frac{l_a{}^2}{1+l_a{}^2} \end{bmatrix} \\ &= \begin{bmatrix} 0 & \dfrac{1}{R} & \dfrac{l_a}{R} \end{bmatrix}.\end{aligned}$$

Similarly,

$$\left[\mathbf{J}_{2a}{}^{T}\Delta(\mathbf{J}_{2u})\mathbf{J}_{2a}\right]^{-1}\mathbf{J}_{2a}{}^{T}\Delta(\mathbf{J}_{2u}) = \begin{bmatrix} 0 & \frac{1}{R} & -\frac{l_a}{R} \end{bmatrix}.$$

Thus,

$$\mathbf{J}_{a} = \frac{1}{R}\begin{bmatrix} 0 & 1 & l_a \\ 0 & 1 & -l_a \end{bmatrix},$$

and the actuated inverse kinematics relationship (4) becomes

$$\begin{bmatrix} \omega_{x1} \\ \omega_{x2} \end{bmatrix} = \frac{1}{R}\begin{bmatrix} 0 & 1 & l_a \\ 0 & 1 & -l_a \end{bmatrix}\begin{bmatrix} v_{x\mathrm{RB}} \\ v_{y\mathrm{RB}} \\ \omega_{\mathrm{RB}} \end{bmatrix}.$$

"Sensed" Forward Kinematics

Muir and Newman define a "sensed" Jacobian matrix that relates the desired robot velocity vector to those velocity components that can actually be measured by

$$\mathbf{v}_{\mathrm{RB}} = \mathbf{J}_{s}\mathbf{v}_{s}, \tag{5}$$

where $\mathbf{v}_{s}$ is a matrix of sensed wheel velocity components, and $\mathbf{J}_{s}$ is the sensed-wheel Jacobian. For our robot,

$$\mathbf{J}_{s} = \left[\Delta(\mathbf{J}_{1n}) + \Delta(\mathbf{J}_{2n})\right]^{-1}\left[\Delta(\mathbf{J}_{1n})\mathbf{J}_{1s} \quad \Delta(\mathbf{J}_{2n})\mathbf{J}_{2s}\right].$$

where $\mathbf{J}_{ks}$ and $\mathbf{J}_{kn}$ are the columns of the wheel Jacobians that correspond to sensed and non-sensed velocity components, respectively.

In our robot, only the angular velocity of the wheels about their axles can be measured (shaft encoders), i.e., only ω_{x1} and ω_{x2} are sensed. Therefore, we have

$$\mathbf{J}_{1s} = \mathbf{J}_{2s} = \begin{bmatrix} 0 \\ R \\ 0 \end{bmatrix}, \quad \mathbf{J}_{1n} = \begin{bmatrix} 0 \\ -l_a \\ 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{J}_{2n} = \begin{bmatrix} 0 \\ l_a \\ 1 \end{bmatrix}.$$

Thus,

$$\Delta(\mathbf{J}_{1n}) + \Delta(\mathbf{J}_{2n}) = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -\frac{2}{1+l_a{}^2} & 0 \\ 0 & 0 & -\frac{2l_a{}^2}{1+l_a{}^2} \end{bmatrix}, \quad \text{and}$$

$$\left[\Delta(\mathbf{J}_{1n})+\Delta(\mathbf{J}_{2n})\right]^{-1} = \begin{bmatrix} -\frac{1}{2} & 0 & 0 \\ 0 & -\frac{1+l_a^2}{2} & 0 \\ 0 & 0 & -\frac{1+l_a^2}{2l_a^2} \end{bmatrix}, \quad \text{and}$$

$$\left[\Delta(\mathbf{J}_{1n})\mathbf{J}_{1s} \quad \Delta(\mathbf{J}_{2n})\mathbf{J}_{2s}\right] = \begin{bmatrix} 0 & 0 \\ -\frac{R}{1+l_a^2} & -\frac{R}{1+l_a^2} \\ -\frac{Rl_a}{1+l_a^2} & -\frac{Rl_a}{1+l_a^2} \end{bmatrix}.$$

Therefore,

$$\mathbf{J}_s = \begin{bmatrix} -\frac{1}{2} & 0 & 0 \\ 0 & -\frac{1+l_a^2}{2} & 0 \\ 0 & 0 & -\frac{1+l_a^2}{2l_a^2} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -\frac{R}{1+l_a^2} & -\frac{R}{1+l_a^2} \\ -\frac{Rl_a}{1+l_a^2} & -\frac{Rl_a}{1+l_a^2} \end{bmatrix}$$

$$= \frac{R}{2l_a} \begin{bmatrix} 0 & 0 \\ l_a & l_a \\ 1 & -1 \end{bmatrix},$$

and the sensed forward kinematics relationship (5) becomes

$$\begin{bmatrix} v_{x\mathrm{RB}} \\ v_{y\mathrm{RB}} \\ \omega_{\mathrm{RB}} \end{bmatrix} = \frac{R}{2l_a} \begin{bmatrix} 0 & 0 \\ l_a & l_a \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \omega_{x1} \\ \omega_{x2} \end{bmatrix}.$$

Dead Reckoning Equations

The vehicle's position after n sampling periods, T, in terms of its position at the previous sampling instant, and its velocity currently and at the previous sample is given by the following relationship:

$$\mathbf{p}_{\mathrm{RB}}(nT) = \mathbf{p}_{\mathrm{RB}}[(n-1)T] + \frac{T}{2}\mathbf{V}[(n-1)T]\{\mathbf{v}_{\mathrm{RB}}[(n-1)T] + \mathbf{v}_{\mathrm{RB}}(nT)\},$$

where $\mathbf{V}$ is the orthogonal transformation matrix

$$\mathbf{V} = \begin{bmatrix} \cos\theta_{\mathrm{RB}} & -\sin\theta_{\mathrm{RB}} & 0 \\ \sin\theta_{\mathrm{RB}} & \cos\theta_{\mathrm{RB}} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The notation $\mathbf{x}[(n\text{-}1)T]$ is meant to denote the value of vector or matrix $\mathbf{x}$ after n-1 sampling periods of length T. That is, where the components of vector or matrix $\mathbf{x}$ change through time, in $\mathbf{x}[(n\text{-}1)T]$ the time-varying components have the values that obtain $(n\text{-}1)T$ seconds after sampling began: in the position vector $\mathbf{p}$, these values are the robot's x- and y-displacement and orientation terms; in velocity vector $\mathbf{v}$, these are the angular velocities of the robot's wheels about their axles; finally, in the transformation matrix $\mathbf{V}$, these are the sines and cosines of the robot's orientation, as this angle itself changes through time.

This expression is adequate for the kinds of robot motion control implemented on the robot described in Chapter 3. This robot moves forwards in straight lines exclusively. When it wishes to change direction, it turns on the spot, and carries on in the new direction.

So, expanding the position expression given above, and substituting the sensed forward kinematics relationship derived in the previous section for $\mathbf{v}_{RB}$, we get

$$\begin{bmatrix} p_{x\mathrm{RB}} \\ p_{y\mathrm{RB}} \\ \theta_{\mathrm{RB}} \end{bmatrix}(nT) = \begin{bmatrix} p_{x\mathrm{RB}} \\ p_{y\mathrm{RB}} \\ \theta_{\mathrm{RB}} \end{bmatrix}[(n-1)T] +$$
$$+ \frac{T}{2}\begin{bmatrix} \cos\theta_{\mathrm{RB}} & -\sin\theta_{\mathrm{RB}} & 0 \\ \sin\theta_{\mathrm{RB}} & \cos\theta_{\mathrm{RB}} & 0 \\ 0 & 0 & 1 \end{bmatrix}[(n-1)T]\frac{R}{2l_a}\begin{bmatrix} 0 & 0 \\ l_a & l_a \\ 1 & -1 \end{bmatrix}\left(\begin{bmatrix} \omega_{x1} \\ \omega_{x2} \end{bmatrix}[(n-1)T] + \begin{bmatrix} \omega_{x1} \\ \omega_{x2} \end{bmatrix}(nT)\right).$$

The x- and y-displacement terms simplify to

$$\begin{bmatrix} p_{x\mathrm{RB}} \\ p_{y\mathrm{RB}} \end{bmatrix}(nT) = \begin{bmatrix} p_{x\mathrm{RB}} \\ p_{y\mathrm{RB}} \end{bmatrix}[(n-1)T] +$$
$$+ \frac{TR}{4}\begin{bmatrix} -\sin\theta_{\mathrm{RB}} & -\sin\theta_{\mathrm{RB}} \\ \cos\theta_{\mathrm{RB}} & \cos\theta_{\mathrm{RB}} \end{bmatrix}[(n-1)T]\left(\begin{bmatrix} \omega_{x1} \\ \omega_{x2} \end{bmatrix}[(n-1)T] + \begin{bmatrix} \omega_{x1} \\ \omega_{x2} \end{bmatrix}(nT)\right).$$

On the other hand, the orientation term can be shown to reduce to

$$\theta_{\mathrm{RB}}(nT) = \frac{R}{2l_a}[\theta_{1x}(nT) - \theta_{2x}(nT)] + \theta_{\mathrm{RB}}(0) - \frac{R}{2l_a}[\theta_{1x}(0) - \theta_{2x}(0)],$$

where θ_{kx} is the total angular (rotational) displacement of wheel k about its axle. In other words, the x- and y-displacement terms of the robot's position after n sampling periods are

a function of its displacement, velocity, and orientation after n-1 sampling periods. However, the orientation of the robot after n sampling periods can be determined directly from the difference between the total displacement of the two wheels about their axles; since this is measured by shaft encoders, the robot's orientation (subject to errors due to slippage and friction) can be found at any time simply by subtracting the two encoder counts, and multiplying the difference by a suitable constant (and offsetting by the difference between the counts when sampling began and the vehicle's orientation at that time).

References

Alexander, J.C. and J.H. Maddocks, "On the Kinematics of Wheeled Mobile Robots", Int. J. Rob. Res. **8** (5), 1989, pp. 15-27.

Arkin, R.C., "Motor Schema-Based Mobile Robot Navigation", Int. J. Rob. Res. **8** (4), 1989, pp. 92-112.

Arkin, R.C. and D.T. Lawton, "Reactive Behavioural Support for Qualitative Visual Navigation", Proc. IEEE Workshop on Intelligent Motion Control, Istanbul, Turkey, August 1990, pp. IP21-IP27.

Arkin, R.C. and R.R. Murphy, "Autonomous Navigation in a Manufacturing Environment", IEEE J. Rob. Autom. **6** (4), 1990, pp. 445-454.

Ayache, N. and O. Faugeras, "Maintaining representations of the environment of a mobile robot", IEEE Trans. Rob. Autom. **5** (6), 1989, pp. 804-819.

Baader, A. and J. Hallam, "Mapbuilding and Control Algorithms for Mobile Robots", unpublished manuscript, 1992.

Barto, A.G. and R.S. Sutton, "Landmark learning: an illustration of associative search", Biol. Cybern. **42**, 1981, pp. 1-8.

Borenstein, J. and Y. Koren, "Real-time obstacle avoidance for fast mobile robots in cluttered environments", Proc. IEEE Int. Conf. Rob. Autom., Cincinnati, Ohio, May 1990, pp. 572-577.

Borland International, Inc., *Turbo Pascal® Object-oriented programming guide*, Scotts Valley, CA: Borland International, Inc., 1988.

Brooks, R.A., "Symbolic Error Analysis and Robot Planning", Int. J. Rob. Res. **1** (4), 1982, pp. 29-68.

Brooks, R.A., "Solving the Find-Path Problem by Good Representations of Free Space", IEEE Trans. Sys. Man Cyb. **SMC-13** (3), 1983, pp. 190-197.

Brooks, R.A., "Achieving Artificial Intelligence Through Building Robots", MIT AI Lab Memo 899, May 1986(a).

Brooks, R.A., "A Robust Layered Control System for a Mobile Robot", IEEE J. Rob. Autom. **RA-3**, 1986(b), pp. 14-23.

Brooks, R.A., "Intelligence without representation", Artificial Intelligence **47**, 1991(a), pp. 139-159.

Brooks, R.A., "Intelligence without reason", MIT AI Lab Memo 1293, April 1991(b).

Carlton, R.E. and S.J. Bartholet, "The evolution of the application of mobile robotics to nuclear facility operations and maintenance", Proc. IEEE Int. Conf. Rob. Autom., Raleigh, North Carolina, March 1987, pp. 720-726.

Cartright, B.A. and T.S. Collett, "Landmark Learning in Bees", J. Comp. Physiol. A **151**, 1983, pp. 521-543.

Connell, J.H., "A Colony Architecture for an Artificial Creature", Ph.D. Thesis, MIT, published as MIT AI Lab Tech. Rep. 1151, Sept. 1989.

Crowley, J.L., "Asynchronous Control of Orientation and Displacement in a Robot Vehicle", Proc. IEEE Int. Conf. Rob. Autom., Scottsdale, Arizona, May 1989(a), pp. 1277-1282.

Crowley, J.L., "World Modelling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging", Proc. IEEE Int. Conf. Rob. Autom., Scottsdale, Arizona, May 1989(b), pp. 674-681.

Donnett, J. and B. McGonigle, "Evolving Speed Control in Mobile Robots: From Blindness to Kinetic Vision", Proc. Vision Interface '91, Calgary, Canada, June 1991, pp. 35-41.

Donnett, J. and T. Smithers, "Lego Vehicles: A Technology for Studying Intelligent Systems", Proc. 1st Int. Conf. on the Simulation of Adaptive Behaviour, Paris, France, September 1990, pp. 540-549.

Durieu, C., H. Clergeot, and F. Monteil, "Localization of a mobile robot with beacons taking erroneous data into account", Proc. IEEE Int. Conf. Rob. Autom., Scottsdale, Arizona, May 1989, pp. 1062-1068.

Durrant-Whyte, H.F., "Uncertain Geometry in Robotics", IEEE J. Rob. Autom. **4** (1), Feb. 1988(a), pp. 23-31.

Durrant-Whyte, H.F., "Sensor Models and Multisensor Integration", Int. J. Rob. Res. **7** (6), 1988(b), pp. 97-113.

Elfes, A., "Sonar-Based Real-World Mapping and Navigation", IEEE. J. Rob. Autom. **RA-3** (3), 1987, pp. 249-265.

Elfes, A., "Dynamic control of robot perception using stochastic spatial models", Int. Workshop on Information Processing in Mobile Robots, Munich, Germany, March 1991.

Ferguson, I.A., "TouringMachines: Autonomous Agents with Attitudes", IEEE Computer **25** (5), May 1992, pp. 51-55.

Flynn, A.M., "Combining Sonar and Infrared Sensors for Mobile Robot Navigation", Int. J. Rob. Res. **7** (6), 1988, pp. 5-14.

Fodor, J.A., *The Modularity of Mind*, Cambridge, Massachusetts: MIT Press, 1983.

Gallistel, C.R., *The Organization of Learning*, Cambridge, Massachusetts: MIT Press, 1990.

Gibson, J.J., *The Perception of the Visual World*, 1950, Boston: Houghton Mifflin.

Gibson, J.J., *The Ecological Approach to Visual Perception*, 1979, Boston: Houghton Mifflin.

Giralt, G., R. Chatila, and M. Vaisset, "An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots", in *The First International Symposium on Robotics Research*, eds. M. Brady and R. Paul, Cambridge, Massachusetts: MIT Press, 1984, pp. 191-214.

Goldstein, M., F.G. Pin, G. de Saussure, and C.R. Weisbin, "3-D world modeling based on combinatorial geometry for autonomous robot navigation", Proc. IEEE Int. Conf. Rob. Autom., Raleigh, North Carolina, March 1987, pp. 727-733.

Goto, Y. and A. Stentz, "The CMU System for Mobile Robot Navigation", Proc. IEEE Int. Conf. Rob. Autom., Raleigh, North Carolina, March 1987, pp. 99-105.

Hallam, J.C.T., "Intelligent automatic interpretation of active marine sonar", Ph.D. Thesis, Dept. of Artificial Intelligence, Univ. Edinburgh, 1985.

Hallam, J., "Analysing specular echoes in active acoustic range data", in *Artificial Intelligence and its Applications*, eds. A.G. Cohn and J.R. Thomas, London: John Wiley & Sons Ltd., 1986, pp. 165-177.

Hallam, J., P.Forster, and J. Howe, "Map-Free Localisation in a Partially Moving 3D World: the Edinburgh Feature-Based Navigator", Intell. Auton. Sys. 2 Conf., Amsterdam, December 1989, pp. 726-736.

Hayes, G.M., "A real-time kinetic depth system", M.Sc. Dissertation, Dept. of Artificial Intelligence, Univ. Edinburgh, 1990.

Khinchin, A.I., *Mathematical Foundations of Information Theory*, New York: Dover Publications, Inc., 1957; translated by R.A. Silverman and M.D. Friedman, originally appearing in Russian in Uspekhi Matematicheskikh Nauk **VII** (3), 1953, pp. 3-20 ("The Entropy Concept in Probability Theory"), and **XI** (1), pp. 17-75 ("On the Fundamental Theorems of Information Theory").

Kimura, I, O. Ohkuma, K. Ohsubo, and J. Tadano, "Guide and carry robot for hospital use", Robot **53**, August 1986, pp. 116-121.

Kirsh, D., "Today the earwig, tomorrow man?", Artificial Intelligence **47**, 1991, pp. 161-184.

Klein, F., *Vorlesungen Über Höhere Geometrie*, (Lectures in Higher Geometry), Berlin: Springer Verlag, 1926; edited by W. Blaschke from "Vergleichende Betrachtungen über neuere geometrische Forschungen", (Comparative observations on recent geometric research), Math. Annalen **43** (1893) and Klein's Erlangen lecture of 1872.

Klinowska, M., "Cetacean 'Navigation' and the Geomagnetic Field", J. Navigation **41** (I), 1988, pp. 52-71.

Koenderink, J.J. and A.J. van Doorn, "How an Ambulent Observer Can Construct a Model of the Environment From the Geometrical Structure of the Visual Inflow", in *Kybernetik*, eds. G. Hauske and E. Butenandt, 1977, pp. 224-247.

Kriegman, D.J., E. Triendl, and T.O. Binford, "A Mobile Robot: Sensing, Planning, and Locomotion", Proc. IEEE Int. Conf. Rob. Autom., Raleigh, North Carolina, March 1987, pp. 402-408.

Kuc, R. and M.W. Siegel, "Physically based simulation model for acoustic sensor robot navigation", IEEE Trans. Patt. Anal. Mach. Intell. **PAMI-9** (6), 1987, pp. 766-778.

Kuipers, B.J. and Y.T. Byun, "A Qualitative Approach to Robot Exploration and Map-Learning", Proc. Workshop on Spatial Reasoning and Multi-Sensor Fusion, 1987, pp. 390-404.

Kuipers, B.J. and T.S. Levitt, "Navigation and Mapping in Large-Scale Space", AI Magazine **9** (2), 1988, pp. 25-43.

Kuritsky, M.M. and M.S. Goldstein, "Inertial Navigation", Proc. IEEE **71** (10), 1983, pp. 1156-1176.

Lawton, D.T., T.S. Levitt, C.C. McConnell, P.Nelson, and J. Glicksman, "Terrain Models for an Autonomous Land Vehicle", Proc. IEEE Conf. on Rob. Autom., San Francisco, California, April 1986, pp. 2043-2051.

Layman, J., "Hospitals convert supply transport", Robotics World **4** (13), December 1986, pp. 30-32.

Leonard, J.J., "Directed Sonar Sensing for Mobile Robot Navigation", Ph.D. Thesis, Dept. of Engineering Science, Univ. Oxford, 1990.

Leonard, J.J., H.F. Durrant-Whyte, and I.J. Cox, "Dynamic Map Building for an Autonomous Mobile Robot", Int. J. Rob. Res. **11** (4), 1992, pp. 286-298.

Levitt, T.S., D.T. Lawton, D.M. Chelberg, and P. Nelson, "Qualitative Landmark-Based Path Planning and Following", Proc. AAAI-87 Conf. on Artif. Intell., September 1987, pp. 689-694.

Levitt, T.S. and D.T. Lawton, "Qualitative Navigation for Mobile Robots", Artificial Intelligence **44**, 1990, pp. 305-360.

Lozano-Pérez, T., "Spatial Planning: A Configuration Space Approach", IEEE Trans. Computers **C-32** (2), 1983, pp. 108-120.

Malcolm, C.A., "A Behavioural Approach to Robot Task Planning", M.Sc. Dissertation, Dept. of Artificial Intelligence, Univ. Edinburgh, 1987.

Malcolm, C.A., T. Smithers, and J. Hallam, "An Emerging Paradigm in Robot Architecture", Intell. Auton. Sys. 2 Conf., Amsterdam, December 1989, pp. 545-564.

Mann, R.C., W.R. Hamel, and C.R. Weisbin, "The development of an intelligent nuclear maintenance robot", Proc. IEEE Int. Conf. Rob. Autom., Philadelphia, Pennsylvania, April 1988, pp. 621-623.

Mardia, K.V., *Statistics of Directional Data*, London: Academic Press, 1972.

Mataric, M.J., "Navigation with a Rat Brain: A Neurobiologically-Inspired Model for Robot Spatial Representation", Proc. 1st Int. Conf. on the Simulation of Adaptive Behaviour, Paris, France, September 1990, pp. 169-175.

McGillem, C.D. and G.R. Cooper, *Continuous and Discrete Signal and System Analysis*, New York, NY: Holt, Rinehart, and Winston Inc., 1984.

McKerrow, P.J., *Introduction to Robotics*, Sydney, Australia: Addison-Wesley Ltd., 1991.

Milberg, J. and P. Luts, "Integration of autonomous mobile robots in the industrial production environment", Proc. 1st Int. Conf. on Automated Guided Vehicle Systems, Stratford-upon-Avon, June 1981, pp. 209-214.

Minsky, M., "Steps Toward Artificial Intelligence", in *Computers and Thought*, eds. E.A. Feigenbaum and J. Feldman, New York, NY: McGraw Hill, 1963.

Moravec, H.P., "The Stanford Cart and the CMU Rover", Proc. IEEE **71** (7), 1983, pp. 872-884.

Moravec, H.P. and A. Elfes, "High resolution maps from wide angle sonar", Proc. IEEE Int. Conf. Rob. Autom., St. Louis, Missouri, March 1985, pp. 116-121.

Muir, P.F. and C.P. Newman, "Kinematic Modelling for Feedback Control of an Omnidirectional Wheeled Mobile Robot", Proc. IEEE Int. Conf. Rob. Autom., Raleigh, North Carolina, March 1987(a), pp. 1772-1778.

Muir, P.F. and C.P. Newman, "Kinematic modelling of wheeled mobile robots", J. Rob. Sys. **4** (2), 1987(b), pp. 281-340.

Müller, T., *Automated Guided Vehicles*, Berlin: Springer-Verlag, 1983.

Nehmzow, U., J. Hallam, and T. Smithers, "Really Useful Robots", Intell. Auton. Sys. 2 Conf., Amsterdam, December 1989, pp. 284-293.

Nehmzow, U. and T. Smithers, "Map-Building using Self-Organising Networks", Proc. 1st Int. Conf. on the Simulation of Adaptive Behaviour, Paris, France, September 1990, pp. 152-159.

Nelson, W.L. and I.J. Cox, "Local Path Control for an Autonomous Vehicle", Proc. IEEE Int. Conf. Rob. Autom., Philadelphia, Pennsylvania, April 1988, pp. 1504-1510.

Nilsson, N.J., "A Mobile Automaton: An Application of Artificial Intelligence Techniques", Proc. 1st Int. Joint Conf. Artif. Intell., Washington, D.C., May 1969, pp. 509-520.

O'Keefe, J., "Place Units in the Hippocampus of the Freely Moving Rat", Experimental Neurology **51**, 1976, pp. 78-109.

O'Keefe, J. and J. Dostrovsky, "The hippocampus as a spatial map", Brain Research **34**, 1971, pp. 171-175.

O'Keefe, J. and L. Nadel, *The Hippocampus as a Cognitive Map*, Oxford: Clarendon Press, 1978.

Rosenschein, S.J. and L.P. Kaelbling, "The Synthesis of Digital Machines with Provable Epistemic Properties", in *Theoretical Aspects of Reasoning about Knowledge*, ed. J.Y. Halpern, Los Altos, California: Morgan Kaufmann, 1986.

Roth-Tabak, Y. and R. Jain, "Building an Environment Model Using Depth Information", IEEE Computer **22** (6), 1989, pp. 85-90.

Sarachik, K.B., "Characterising an Indoor Environment with a Mobile Robot and Uncalibrated Stereo", Proc. IEEE Int. Conf. Rob. Autom., Scottsdale, Arizona, May 1989, pp. 984-989.

Sekiguchi, M., S. Nagata, and K. Asakawa, "Behaviour Control for a Mobile Robot by Multi-Hierarchical Neural Network", Proc. IEEE Int. Conf. Rob. Autom., Scottsdale, Arizona, May 1989, pp. 1578-1583.

Shannon, C.E. and W. Weaver, *The Mathematical Theory of Communication*, Urbana, Illinois: The University of Illinois Press, 1964; reprint with minor corrections of Shannon, C.E., "The Mathematical Theory of Communication", Bell System Technical Journal, July and October 1948, and expanded version of Weaver, W., "Recent Contributions to the Mathematical Theory of Communications", Sci. Am., July 1949.

Smith, R. and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty", Int. J. Rob. Res. **5** (4), 1986, pp. 56-68.

Smith, R., M. Self, and P. Cheeseman, "Estimating Uncertain Spatial Relationships in Robotics", in *Uncertainty in Artificial Intelligence 2*, eds. J.F. Lemmer and L.N. Kanal, Amsterdam: Elsevier Science Publishers B.V. (North-Holland), 1988.

Squire, L.R., "Memory and the Hippocampus: A Synthesis From Findings With Rats, Monkeys, and Humans", Psychol. Rev. **99** (2), 1992, pp. 195-231.

Suga, N., "Biosonar and Neural Computation in Bats", Sci. Am., June 1990, pp. 34-41.

Tolman, E.C., "Cognitive Maps in Rats and Men", Psychol. Rev. **55** (4), 1948, pp. 189-208.

Tolman, E.C., B.F. Ritchie, and D. Kalish, "Studies in spatial learning. I. Orientation and the short-cut", J. Exp. Psychol. **36**, 1946, pp. 13-24.

White, J.R., H.W. Harvey, and K.A. Farnstrom, "Testing of mobile surveillance robot at a nuclear power plant", Proc. IEEE Int. Conf. Rob. Autom., Raleigh, North Carolina, March 1987, pp. 714-719.

Winkler, R.L., *Introduction to Bayesian Inference and Decision*, New York: Holt, Rinehart, and Winston, Inc., 1972.