1 *Introduction*

1.1 Uncertainty in Robotics

Robotics is the science of perceiving and manipulating the physical world through computer-controlled devices. Examples of successful robotic systems include mobile platforms for planetary exploration, industrial robotics arms in assembly lines, cars that travel by themselves, and manipulators that assist surgeons. Robotics systems are situated in the physical world, perceive information on their environments through sensors, and manipulate through physical forces.

While much of robotics is still in its infancy, the idea of "intelligent" manipulating devices has an enormous potential to change society. Wouldn't it be great if all our cars were able to safely steer themselves, making car accidents a notion of the past? Wouldn't it be great if robots, and not people, would clean up nuclear disaster sites like Chernobyl? Wouldn't it be great if our homes were populated by intelligent assistants that take care of all domestic repair and maintenance tasks?

To do these tasks, robots have to be able to accommodate the enormous uncertainty that exists in the physical world. There is a number of factors that contribute to a robot's uncertainty.

First and foremost, *robot environments* are inherently unpredictable. While the degree of uncertainty in well-structured environments such as assembly lines is small, environments such as highways and private homes are highly dynamic and in many ways highly unpredictable. The uncertainty is particularly high for robots operating in the proximity of people.

Sensors are limited in what they can perceive. Limitations arise from several factors. The range and resolution of a sensor is subject to physical limitations. For example, cameras cannot see through walls, and the spatial resolution of a camera image is limited. Sensors are also subject to noise, which perturbs sensor measurements in unpredictable ways and hence limits the information that can be extracted. And finally, sensors can break. Detecting a faulty sensor can be extremely difficult.

Robot actuation involves motors that are, at least to some extent, unpredictable. Uncertainty arises from effects like control noise, wear-and-tear, and mechanical failure. Some actuators, such as heavy-duty industrial robot arms, are quite accurate and reliable. Others, like low-cost mobile robots, can be extremely flaky.

Some uncertainty is caused by the robot's software. All *internal models* of the world are approximate. Models are abstractions of the real world. As such, they only partially model the underlying physical processes of the robot and its environment. Model errors are a source of uncertainty that has often been ignored in robotics, despite the fact that most robotic models used in state-of-the-art robotics systems are rather crude.

Uncertainty is further created through *algorithmic approximations*. Robots are real-time systems. This limits the amount of computation that can be carried out. Many popular algorithms are approximate, achieving timely response through sacrificing accuracy.

The level of uncertainty depends on the application domain. In some robotic applications, such as assembly lines, humans can cleverly engineer the system so that uncertainly is only a marginal factor. In contrast, robots operating in residential homes or on other planets will have to cope with substantial uncertainty. Such robots are forced to act even though neither their sensors, nor their internal models, will provide it with sufficient information to make the right decisions with absolute certainty. As robotics is now moving into the open world, the issue of uncertainty has become a major stumbling block for the design of capable robot systems. Managing uncertainty is possibly the most important step towards robust real-world robot systems.

Hence this book.

1.2 Probabilistic Robotics

This book provides a comprehensive overview of *probabilistic robotics*. Probabilistic robotics is a relatively new approach to robotics that pays tribute to the uncertainty in robot perception and action. The key idea in probabilistic robotics is to represent uncertainty explicitly using the calculus of probability theory. Put differently, instead of relying on a single "best guess" as to what might be the case, probabilistic algorithms represent information by probability distributions over a whole space of guesses. By doing so, they can represent ambiguity and degree of belief in a mathematically sound way. Control choices can be made robust relative the uncertainty that remains, and probabilistic robotics can even actively chose to reduce their uncertainty when this appears to be the superior choice. Thus, probabilistic algorithms degrade gracefully in the face of uncertainty. As a result, they outperform alternative techniques in many real-world applications.

We shall illustrate probabilistic robotics with two motivating examples: one pertaining to robot perception, and another to planning and control.

MOBILE ROBOT LOCALIZATION

Our first example is *mobile robot localization*. Robot localization is the problem of estimating a robot's coordinates relative to an external reference frame. The robot is given a map of its environment, but to localize itself relative to this map it needs to consult its sensor data. Figure 1.1 illustrates such a situation. The environment is known to possess three indistinguishable doors. The task of the robot is to find out where it is, through sensing and motion.

This specific localization problem is known as *global localization*. In global localization, a robot is placed somewhere in a known environment and has to localize itself from scratch. The probabilistic paradigm represents the robot's momentary *belief* by a probability density function over the space of all locations. This is illustrated in diagram (a) in Figure 1.1. This diagram shows a uniform distribution over all locations. Now suppose the robot takes a first sensor measurement and observes that it is next to a door. Probabilistic techniques exploit this information to update the belief. The 'posterior' belief is shown in diagram (b) in Figure 1.1. It places an increased probability at places near doors, and lower probability near walls. Notice that this distribution possesses three peaks, each corresponding to one of the indistinguishable doors in the environment. Thus, by no means does the robot *know* where it is. Instead, it now has three, distinct hypotheses which are each equally plausible given the sensor data. We also note that the robot assigns positive probability to places *not* next to a door. This is the natural result of the inherent uncertainty in sensing: With a small, non-zero probability, the robot might have erred in its assessment of seeing a door. The ability to maintain low-probability hypotheses is essential for attaining robustness.

Now suppose the robot moves. Diagram (c) in Figure 1.1 shows the effect on a robot's belief. The belief has been shifted in the direction of motion. It also possesses a larger spread, which reflects the uncertainty that is intro-

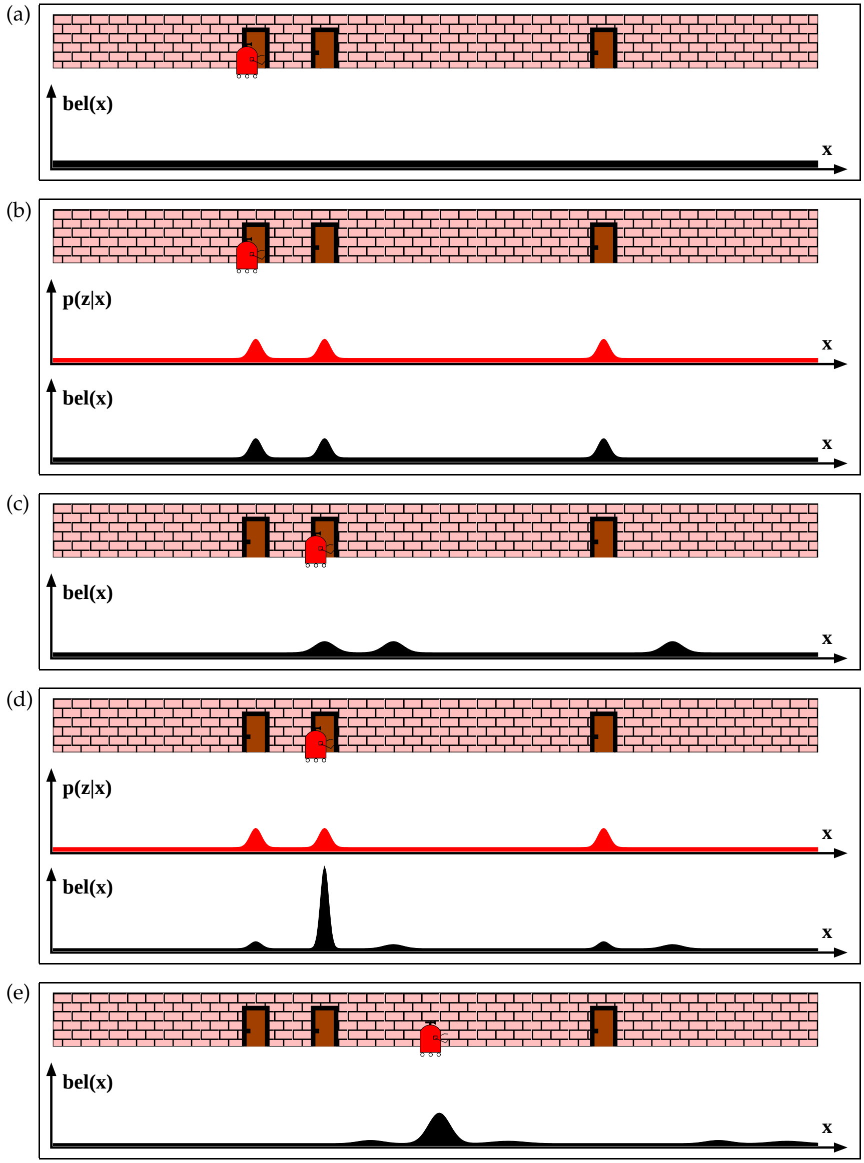

Figure 1.1 The basic idea of *Markov localization*: A mobile robot during global localization. Markov localization techniques will be investigated in Chapters 7 and 8.

(a)

(b)

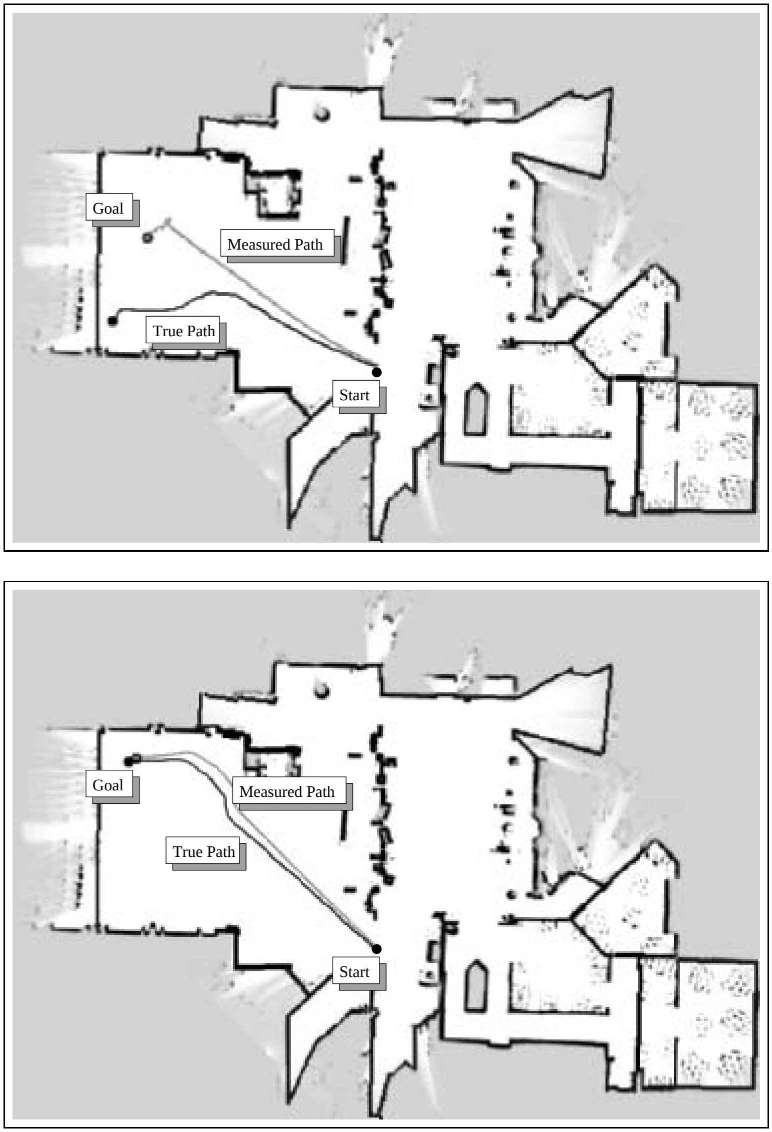

Figure 1.2 Top image: a robot navigating through open, featureless space may lose track of where it is. Bottom: This can be avoided by staying near known obstacles. These figures are results of an algorithm called *coastal navigation*, which will be discussed in Chapter 16. Images courtesy of Nicholas Roy, MIT.

duced by robot motion. Diagram (d) in Figure 1.1 depicts the belief after observing another door. This observation leads our algorithm to place most of the probability mass on a location near one of the doors, and the robot is now quite confident as to where it is. Finally, Diagram (e) shows a belief as the robot travels further down the corridor.

This example illustrates many aspects of the probabilistic paradigm. Stated probabilistically, the robot perception problem is a state estimation problem, and our localization example uses an algorithm known as *Bayes filter* for posterior estimation over the space of robot locations. The representation of information is a probability density function. The update of this function represents the information gained through sensor measurements, or the information lost through processes in the world that increase a robot's uncertainty.

BAYES FILTER

Our second example brings us into the realm of robotic planning and control. As just argued, probabilistic algorithms can compute a robot's momentary uncertainty. But they can also anticipate future uncertainty, and take such uncertainty into consideration when determining the right choice of control. One such algorithm is called *coastal navigation*. An example of coastal navigation is shown in Figure 1.2. This figure shows a 2-D map of an actual building. The top diagram compares an estimated path with an actual path: The divergence is the result of the uncertainty in robot motion that we just discussed. The interesting insight is: not all trajectories induce the same level of uncertainty. The path in Figure 1.2a leads through relatively open space, deprived of features that could help the robot to remain localized. Figure 1.2b shows an alternative path. This trajectory seeks a distinct corner, and then "hugs" the wall so as to stay localized. Not surprisingly, the uncertainty will be reduced for the latter path, hence chances of arriving at the goal location are actually higher.

COASTAL NAVIGATION

This example illustrates one of the many ways proper consideration of uncertainty affects the robot's controls. In our example, the anticipation of possible uncertainty along one trajectory makes the robot prefer a second, longer path, just so as to reduce the uncertainty. The new path is better, in the sense that the robot has a much higher chance of actually being at the goal when believing that it is. In fact, the second path is an example of active information gathering. The robot has, through its probabilistic consideration, determined that the best choice of action is to seek information along its path, in its pursuit to reach a target location. Probabilistic planning techniques anticipate uncertainty and can plan for information gathering, and probabilistic control techniques realize the results of such plans.

1.3 Implications

Probabilistic robotics seamlessly integrates models with sensor data, overcoming the limitations of both at the same time. These ideas are not just a matter of low-level control. They cut across all levels of robotic software, from the lowest to the highest.

In contrast with traditional programming techniques in robotics—such as model-based motion planning techniques or reactive behavior-based approaches—probabilistic approaches tend to be more robust in the face of sensor limitations and model limitations. This enables them to scale much better to complex real-world environments than previous paradigms, where uncertainty is of even greater importance. In fact, certain probabilistic algorithms are currently the only known working solutions to hard robotic estimation problems, such as the localization problem discussed a few pages ago, or the problem of building accurate maps of very large environments.

In comparison to traditional model-based robotic techniques, probabilistic algorithms have weaker requirements on the accuracy of the robot's models, thereby relieving the programmer from the insurmountable burden to come up with accurate models. Probabilistic algorithms have weaker requirements on the accuracy of robotic sensors than those made by many reactive techniques, whose sole control input is the momentary sensor input. Viewed probabilistically, the *robot learning problem* is a long-term estimation problem. Thus, probabilistic algorithms provide a sound methodology for many flavors of robot learning.

However, these advantages come at a price. The two most frequently cited limitations of probabilistic algorithms are *computational complexity*, and a *need to approximate*. Probabilistic algorithms are inherently less efficient than their non-probabilistic counterparts. This is due to the fact that they consider entire probability densities instead of a single guess. The need to approximate arises from the fact that most robot worlds are continuous. Computing exact posterior distributions tends to be computationally intractable. Sometimes, one is fortunate in that the uncertainty can be approximated tightly with a compact parametric model (e.g., Gaussians). In other cases, such approximations are too crude to be of use, and more complicated representations must be employed.

Recent developments in computer hardware has made an unprecedented number of FLOPS available at bargain prices. This development has certainly aided the field of probabilistic robotics. Further, recent research has successfully increased the computational efficiency of probabilistic algorithms, for a range of hard robotics problems—many of which are described in depth in this book. Nevertheless, computational challenges remain. We shall revisit this discussion at numerous places, where we investigate the strengths and weaknesses of specific probabilistic solutions.

1.4 Road Map

This book is organized in four major parts.

- Chapters 2 through 4 introduce the basic mathematical framework that underlies all of the algorithms described in this book, along with key algorithms. These chapters are the mathematical foundation of this book.

- Chapters 5 and 6 present probabilistic models of mobile robots. In many ways, these chapters are the probabilistic generalization of classical robotics models. They form the robotic foundation for the material that follows.

- The mobile robot localization problem is discussed in Chapters 7 and 8. These chapters combine the basic estimation algorithms with the probabilistic models discussed in the previous two chapters.

- Chapters 9 through 13 discuss the much richer problem of robotic mapping. As before, they are all based on the algorithms discussed in the foundational chapters, but many of them utilize tricks to accommodate the enormous complexity of this problem.

- Problems of probabilistic planning and control are discussed in Chapters 14 through 17. Here we begin by introducing a number of fundamental techniques, and then branch into practical algorithms for controlling a robot probabilistically. The final chapter, Chapter 17, discusses the problem of robot exploration from a probabilistic perspective.

The book is best read in order, from the beginning to the end. However, we have attempted to make each individual chapter self-explanatory. Frequent sections called *"Mathematical Derivation of . . . "* can safely be skipped on first reading without compromising the coherence of the overall material in this book.

1.5 Teaching Probabilistic Robotics

When used in the classroom, we do *not* recommend to teach the chapters in order—unless the students have an unusually strong appreciation of abstract mathematical concepts. Particle filters are easier to teach than Gaussian filters, and students tend to get more excited by mobile robot localization problems than abstract filter algorithms. In our own teachings, we usually begin with Chapter 2, and move directly to Chapters 7 and 8. While teaching localization, we go back to the material in Chapters 3 through 6 as needed. We also teach Chapter 14 early, to expose students to the problems related to planning and control early on in a course.

As a teacher, feel free to use slides and animations from the book's Web site

`www.probabilistic-robotics.org`

to illustrate the various algorithms in this book. And feel free to send us, the authors, pointers to your class Web sites and any material that could help others in teaching Probabilistic Robotics.

The material in this book is best taught with hands-on implementation assignments. There is nothing more educational in robotics than programming an actual robot. And nobody can explain the pitfalls and challenges in robotics better than Nature!

1.6 Bibliographical Remarks

MODEL-BASED PARADIGM

The field of robotics has gone through a series of paradigms for software design. The first major paradigm emerged in the mid-1970s, and is known as the *model-based paradigm*. The model-based paradigm began with a number of studies showing the hardness of controlling a high-DOF robotic manipulator in continuous spaces (Reif 1979). It culminated in text like Schwartz et al.'s (1987) analysis of the complexity of robot motion, a first singly exponential general motion planning algorithm by Canny (1987), and Latombe's (1991) seminal introductory text into the field of model-based motion planning (additional milestone contributions will be discussed in Chapter 14). This early work largely ignored the problem of uncertainty—even though it extensively began using randomization as a technique for solving hard motion planning problems (Kavraki et al. 1996). Instead, the assumption was that a full and accurate model of the robot and the environment be given, and the robot be deterministic. The model had to be sufficiently accurate that the residual uncertainty was managed by a low-level motion controller. Most motion planning techniques simply produced a single reference trajectory for the control of a manipulator, although ideas such as *potential fields* (Khatib 1986) and *navigation functions* (Koditschek 1987) provided mechanisms for reacting to the unforeseen—as long as it could be sensed. Applications of these early techniques, if any, were confined to environments where every little bit of uncertainty could be engineered away, or sensed with sufficient accuracy.

BEHAVIOR-BASED ROBOTICS

The field took a radical shift in the mid-1980s, when the lack of sensory feedback became the focus of an entire community of researchers within robotics. With strong convictions, the field of *behavior-based robotics* rejected the idea of any internal model. Instead, it was the interaction with a physical environment of a *situated agent* (Kaelbling and Rosenschein 1991) that created the complexity in robot motion (a phenomena often called *emergent behavior* (Steels 1991)). Consequently, sensing played a paramount role, and internal models were rejected (Brooks 1990).

The enthusiasm in this field was fueled by some early successes that were far beyond the reach of traditional model-based motion planning algorithms. One of them was "Genghis," a hexapod robot developed by Brooks (1986). A relatively simple finite state automaton was able to control the gait of this robot even in rugged terrain. The key to success of such techniques lay in sensing: the control was entirely driven by environment interaction, as perceived through the robot's sensors. Some of the early work impressed by creating a seemingly complex robot through clever exploitation of environment feedback (Connell 1990). More recently, the paradigm enjoyed commercial success through a robotic vacuum cleaning robot (IRobots Inc. 2004), whose software follows the behavior-based paradigm.

Due to the lack of internal models and a focus on simple control mechanism, most robot systems were confined to relatively simple tasks, where the momentary sensor information was sufficient to determine the right choice of control. Recognizing this limitation, more recent work in this field embraced *hybrid control* architectures (Arkin 1998), in which behavior-based technique provided low-level control, whereas a model-based planner coordinated the robot's actions at a high, abstract level. Such hybrid architectures are commonplace in robotics today. They are not dissimilar to the seminal work on three-layered architectures by Gat (1998), which took its origins in "Shakey the Robot" (Nilsson 1984).

Modern probabilistic robotics has emerged since the mid-1990s, although its roots can be traced back to the invention of the Kalman filter (Kalman 1960). In many ways, probabilistic robotics falls in between model-based and behavior-based techniques. In probabilistic robotics, there are models, but they are assumed to be incomplete and insufficient for control. There are also sensor measurements, but they too are assumed to be incomplete and insufficient for control. Through the integration of both, models and sensor measurements, a control action can be devised. Statistics provides the mathematical glue to integrate models and sensor measurements.

Many of the key advances in the field of probabilistic robotics will be discussed in future chapters. Some of the cornerstones in this field include the advent of Kalman filter techniques for high-dimensional perception problems by Smith and Cheeseman (1986), the invention of occupancy grid maps by (Elfes 1987; Moravec 1988), and the re-introduction of partially observable planning techniques due to Kaelbling et al. (1998). The past decade has seen an explosion of techniques: Particle filters have become vastly popular (Dellaert et al. 1999), and researchers have developed new programming methodologies focused on Bayesian information processing (Thrun 2000b; Lebeltel et al. 2004; Park et al. 2005). This development went hand in hand with the deployment of physical robot systems driven by probabilistic algorithms, such as industrial machines for cargo handling by Durrant-Whyte (1996), entertainment robots in museums (Burgard et al. 1999a; Thrun et al. 2000a; Siegwart et al. 2003), and robots in nursing and health care (Pineau et al. 2003d). An open-source software package for mobile robot control that heavily utilizes probabilistic techniques is described in Montemerlo et al. (2003a).

The field of commercial robotics is also at a turning point. In its annual World Robotics Survey, the *United Nations and the International Federation of Robotics* 2004 finds a 19% annual increase in the size of the robotic market worldwide. Even more spectacular is the change of market segments, which indicates a solid transition from industrial applications to service robotics and consumer products.