The International Journal of Robotics Research

http://ijr.sagepub.com/

An Objective-Based Framework for Motion Planning under Sensing and Control Uncertainties

Steven M. LaValle and Seth A. Hutchinson

The International Journal of Robotics Research 1998 17: 19
DOI: 10.1177/027836499801700104

Steven M. LaValle
Computer Science Department
Stanford University
Stanford, California 94305 USA

Seth A. Hutchinson
Department of Electrical and Computer Engineering
University of Illinois
Urbana, Illinois 61801 USA

An Objective-Based Framework for Motion Planning under Sensing and Control Uncertainties

Abstract

The authors consider the problem of determining robot motion plans under sensing and control uncertainties. Traditional approaches are often based on methodology known as preimage planning, which involves worst-case analysis. The authors have developed a general framework for determining feedback strategies by blending ideas from stochastic optimal control and dynamic game theory with traditional preimage planning concepts. This generalizes classical preimages to performance preimages and preimage planning for designing motion strategies with information feedback. For a given problem, one can define a performance criterion that evaluates any executed trajectory of the robot. The authors present methods for selecting a motion strategy that optimizes this criterion under either nondeterministic uncertainty (resulting in worst-case analysis) or probabilistic uncertainty (resulting in expected-case analysis). The authors present dynamic programming-based algorithms that numerically compute performance preimages and optimal strategies; several computed examples of forward projections, performance preimages, and optimal strategies are presented.

1. Introduction

We present an objective-based motion planning framework that addresses sensing and control uncertainties. A preliminary version of this work appeared in LaValle and Hutchinson (1994), and more details and related concepts are presented in LaValle (1995, 1997). In an objective-based motion planning framework, motion strategies are chosen by minimizing a criterion that evaluates a trajectory by taking into account quantities such as the path length or execution time. Many of the concepts introduced here are borrowed from stochastic optimal control theory and dynamic game theory (see Başar and Olsder 1982; Kumar and Varaiya 1986) and build on previous preimage planning research (see Erdmann 1984; Latombe, Lazanas, and Shekhar 1991; Lozano-Pérez, Mason, and Taylor 1984). Our approach generalizes classical preimages to performance preimages and preimage planning for designing motion strategies with information feedback. Constructing a traditional termination criterion becomes equivalent to formulating an optimal stopping problem (Bertsekas 1987) within our framework.

Several variations of the motion planning problem have been considered in past research (e.g., see Latombe 1991). One basic form of motion planning, which we refer to as collision-free planning, consists of computing a continuous path in the robot's configuration space that will bring the robot from some initial configuration to a goal configuration while avoiding collisions with obstacles. This is often referred to as the gross-motion planning problem (Hwang and Ahuja 1992). Alternatively, interactions between the robot and objects can be allowed for operations such as compliant motions, pushing, and grasping. These interactions are incorporated into the motion plan of a robot, enabling it to accomplish some specified task. Many researchers have considered the problem of achieving a goal configuration under uncertainty while permitting compliant motions; this has been referred to as the fine-motion planning problem (Canny 1989; Erdmann 1984; Lozano-Pérez, Mason, and Taylor 1984) and the manipulation planning problem (Brost and Christiansen 1996). In other research, manipulation planning has been used to refer to problems that involve the transfer of objects in the workspace (Alami, Siméon, and Laumond 1989; Latombe 1991).

We use the term manipulation planning in this paper to refer to the problem of achieving a goal configuration under uncertainty while permitting compliant motions. For the examples in this paper, we use the manipulation planning model, since it is often considered more difficult than collision-free planning; this is due to the characterization of motions when the robot is in contact with obstacles. The methods presented in this paper can also be adapted to collision-free planning (we have computed results for many cases); however, in this paper, we focus only on manipulation planning, since the

The International Journal of Robotics Research,
Vol. 17, No. 1, January 1998, pp. 19–42,

basic collision-free planning issues under sensing and control uncertainties are included in the manipulation planning model.

Section 2 provides some literature background and states the key contributions of our research. Section 3 provides the general definitions and concepts that form the basis of our approach. Some of the notation, although not previously used to characterize manipulation planning problems, is borrowed from modern control theory. After the general concepts and definitions have been presented, they are used in Sections 4, 5, and 6 to present forward projections, performance preimages, and the determination of optimal motion strategies, respectively. Computed examples are presented at the conclusion of each of these three sections.

The concepts in each of Sections 4 through 6 can be logically divided into four components that represent different types of uncertainty, based on two choices: (1) using probabilistic representations versus nondeterministic (or bounded-set) representations and (2) modeling control errors and assuming perfect sensing versus modeling both control and sensing errors. Each of the probabilistic and nondeterministic representations offers distinct advantages; hence, both are included in our framework. Probabilistic representations lead to expected-case analysis, and nondeterministic representations lead to worst-case analysis; relevant issues are discussed in Section 2. Perfect sensing results are applicable when reasonable state estimation (i.e., configuration estimation) can be performed. The addition of sensing uncertainty can be considered as an extension in which the configuration space is replaced by an information space that is derived from the sensing and action history.

Section 7 discusses several issues regarding our current approach, and Section 8 summarizes our contributions.

2. Background and Motivation

2.1. *Prior Research*

Preimage planning constitutes a large body of research that assumes that sensing and control errors lie within bounded sets. The approach was first conceptualized in Lozano-Pérez, Mason, and Taylor (1984). Using geometric reasoning techniques, a plan is constructed that guarantees that the robot will terminate in a specified subset of configuration space regardless of where the errors might lie within the bounded sets. This plan is generally constructed using recursive subgoals as a form of backchaining. For each subgoal, a preimage is formed that allows the robot to achieve the subgoal for a fixed command, starting from the subset of the configuration space attained from the previous subgoal. Figure 1 shows an example of a preimage. In general, a preimage is defined as the set of all configurations from which a robot is guaranteed to halt in the goal region.

Two basic representations of sensing and control uncertainty have been proposed in the manipulation planning literature; consequently, we will provide a unified treatment of both. We refer to these as nondeterministic uncertainty and probabilistic uncertainty, as done in Erdmann (1992). Under nondeterministic uncertainty, it is assumed that parameter uncertainties lie in a bounded set. Worst-case analysis is performed to yield a motion plan that is guaranteed to be successful regardless of the true value of uncertain parameters within the bounded set. This uncertainty representation is the most common in previous manipulation planning research (e.g., Erdmann 1984; Latombe 1991; Latombe, Lazanas, and Shekhar 1991; Lozano-Pérez, Mason, and Taylor 1984). Under probabilistic uncertainty, probability densities are used to represent uncertainty associated with parameters. This approach has been advocated for manipulation planning by Brost and Christiansen (Brost 1991; Brost and Christiansen 1993, 1996). Each uncertainty representation offers advantages. Nondeterministic models do not require a statistical representation of the errors and, hence, are often easier to specify. If the uncertainty model is correct, the guarantee that the goal is achieved is useful, particularly when the penalty is severe for not achieving it. As noted in Brost and Christiansen (1996) and Erdmann (1992), for many tasks a guaranteed motion plan does not exist; however, a plan can alternatively be computed that achieves the goal with some probability. Since either uncertainty model might be appropriate in a given context, both are considered in this paper.

Several other planning problems and approaches are related to the context developed in this paper. In a series of papers by Donald (1987, 1988, 1990), it was shown how a planner that is capable of recognizing failure (in addition to success) can be used to implement error detection and recovery strategies. Under this model, the robot is allowed to try a new plan after realizing that a failure has occurred, as opposed to continuing the failed plan. This represents an important use of sensor information, and expands the previous notion of reachability to include failure. Goldberg applied preimage planning ideas to construct manipulation plans that orient an object using a parallel gripper without sensors (Goldberg 1990, 1993). Methods for computing backprojections using visual constraint rays that result from the correspondence between edges in the workspace and the image plane are developed in Fox and Hutchinson (1995). Algorithms for computing strategies in the presence of probabilistic uncertainty for mobile robots have been developed in Dean and Wellman (1991); Hu, Brady, and Probert (1991); and Kirman, Basye, and Dean (1991). In Takeda, Facchinetti, and Latombe (1994), a sensory uncertainty field was introduced that indicates positional sensing accuracy for a mobile robot as a function of configuration.

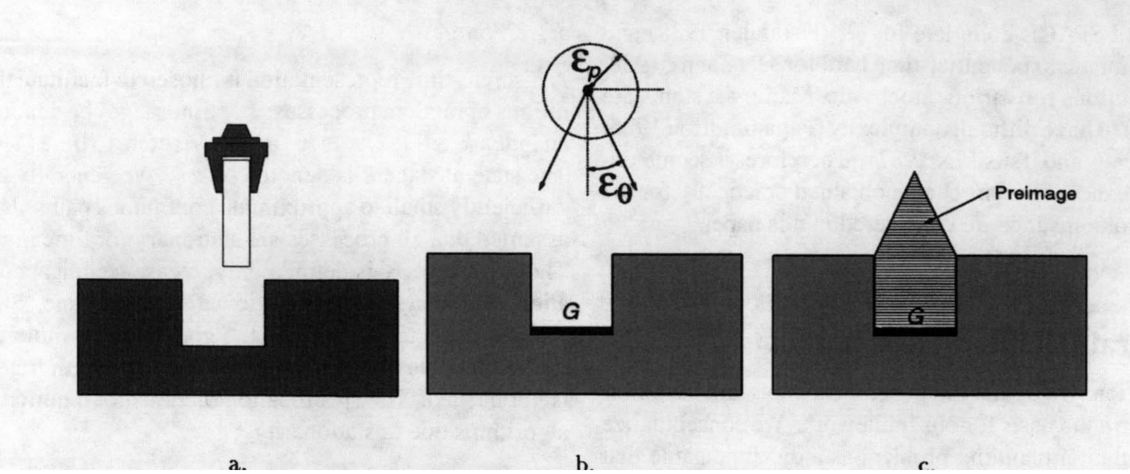


Fig. 1. (a) A classic two-dimensional peg-in-hole insertion task without rotation. (b) Such a task can be represented in configuration space with bounded uncertainty in commanded velocity and sensed configuration. (c) The classical preimage.

2.2. *Motivation*

Previous research in motion planning under sensing and control uncertainties has led to many interesting concepts and algorithms. Similar uncertainty issues have been studied extensively in stochastic optimal control and dynamic game theory, and significant advantages can be obtained by combining previous motion planning concepts (in particular, preimage planning) with modern control concepts. We are therefore motivated to address several issues in this paper.

1. Probabilistic uncertainty is often handled quite differently from nondeterministic uncertainty. We show that the distinction between the two can be reduced to the way in which parameters are chosen by an additional decision maker referred to as *nature*. These parameters potentially interfere with configuration prediction and sensing. As an example of the close connection that we establish between the two uncertainty forms, we show how probabilistic hyperstates, as considered in Brost and Christiansen (1996) and Goldberg (1990), can be obtained by sequentially applying Bayes's rule to the history, whereas nondeterministic knowledge states (Erdmann, 1993) can analogously be obtained by replacing Bayes's rule with appropriate set operations applied to the history. By drawing such connections, methods can potentially be applied to either uncertainty form.

2. For many problems that involve probabilistic or nondeterministic uncertainty, it becomes crucial to optimize a performance criterion such as path length or execution time. In much previous work in motion planning under uncertainty, the task is only to find a strategy that achieves the goal; however, consider the limiting case in which there is severe prediction uncertainty to the point that pure Brownian motion results from a fixed motion command. Many goals will be achieved with probability 1, regardless of the motion strategy. Although it may seem that a system with more uncertainty is easier to control, the execution time, which is not being measured, can be arbitrarily high. Thus, for many problems, we can expect a performance criterion to be crucial in establishing the desired behavior.

3. Sensing uncertainty problems have repeatedly appeared in robotics contexts and are closely related to imperfect observability in control theory. By using the concept of an information space, as considered in stochastic control and dynamic game theory, we provide a general characterization of the fundamental relationship between sensor and action history and decision making. In the case of imperfect information regarding the current configuration, the information space becomes a replacement for the configuration space. Although the current configuration will generally be unknown, an information state (i.e., a point in the information space) will always be known. Thus, a motion planning problem with sensing uncertainty can be transformed into a problem with perfect sensing in an information space. We show that reachability, recognizability, and the termination concepts from preimage planning research can be naturally described in terms of information space concepts.

4. One important difficulty in motion planning under uncertainty is the determination of appropriate uncertainty models. One advantage of having a common mathematical framework is that error models can be changed without altering the general computational approach, facilitating the improvement of error models that are appropriate for a particular robotic system. Configuration space concepts have provided this type of advantage for the basic path planning problem (without uncertainty), and an expanded mathematical formulation can provide similar benefits for incorporating uncertainties.

5. The focus in much previous motion planning research has been on computing exact, geometric solutions. We provide approximate, numerical solutions, since the computational complexity of the exact manipulation planning

problem is PSPACE-complete for $\Re^2$ (Natarajan 1988) and nondeterministic exponential-time hard for $\Re^3$ (Canny 1988, 1989). Solutions for various stochastic, Markov systems are also known to have difficult complexity (Papadimitriou 1985; Papadimitriou and Tsitsiklis 1987). Therefore, it seems unlikely that exact solutions can be obtained practically for the types of problems that are considered in this paper.

3. General Definitions and Concepts

In this section, we define the general concepts and terminology that form the basis for our framework. We conceptualize the manipulation planning problem as a dynamic game that is played between two players: the robot and nature. Both players will be considered as decision-making agents that influence the general state of the system. The robot has a general plan to achieve some goal, whereas nature makes some decisions that potentially interfere with the robot. At an abstract level, this general view of robotic manipulation tasks has been advocated in Taylor, Mason, and Goldberg (1987).

Most of the concepts are expressed in a general form, and a specific modeling example that forms the basis for the experiments is given in Section 3.5.

3.1. States, Stages, and Actions

3.1.1. State Space

Position and orientation information is represented by a point in an n-dimensional configuration space $\mathcal{C}$, for which n is the number of degrees of freedom of the robot. The configuration space could, for example, represent joint angles of a manipulator that is engaged in a manipulation task, such as peg insertion. As another example, the configuration space could specify the position and orientation of a rigid mobile robot. For manipulation planning, a subset of $\mathcal{C}$, denoted as $\mathcal{C}_{\text{valid}}$, is usually defined (see Latombe 1991 for configuration space details). This corresponds to points in $\mathcal{C}$ at which either (1) the robot does not touch an obstacle or (2) the boundary of the robot is in contact with the boundary of some obstacle, but the interiors do not overlap. The second condition enables the possibility of guarded motion and compliance (Whitney 1977), which, for instance, allows the robot to execute a motion along the tangent of an obstacle boundary.

We associate a state space X with a given problem. For the examples in this paper, we take $X = \mathcal{C}_{\text{valid}}$ (for collision-free planning, we would use $X = \mathcal{C}_{\text{free}}$, which is the interior of $\mathcal{C}_{\text{valid}}$). In general, parameters such as velocities or environment characteristics could be included in the state space, while many of the general concepts remain unchanged (LaValle 1995).

3.1.2. Stages

A discrete-time representation is chosen to facilitate the definitions of random processes. Let time stages be denoted with an index $k \in \{1, 2, \dots, K\}$. Stage k refers to time $(k-1)\Delta t$. The state at stage k is denoted by x_k. We generally take Δt sufficiently small to approximate continuous paths. It will be assumed that all processes are stationary (or time invariant). The final stage K is defined only to ease technical considerations as the system evolves toward infinite time. Since the robot is expected to achieve the goal in some finite time (if it is achievable), consideration of infinite-length trajectories is not needed. The specification of K is not required by our algorithms due to stationarity.

3.1.3. Actions

An action (or command), which is denoted by u_k, can be issued to the robot at each stage k. We let U denote the action space for the robot, requiring that $u_k \in U$. The effect of uncertainties will be modeled with an extra decision maker referred to as nature. Let θ_k denote an action for nature, which is chosen from a set Θ. Let θ_k be a vector quantity that is divided into two subvectors θ_k^a and θ_k^s (i.e., $\theta_k = [\theta_k^a \theta_k^s]$). As will be seen shortly, θ_k^a affects the outcome of the robot's actions, and θ_k^s affects the sensor observations of the robot.

3.1.4. State Transition Equation

The effect of a robot action with respect to state is characterized by a state transition equation:

$$x_{k+1} = f(x_k, u_k, \theta_k^a). \tag{1}$$

Hence, given a robot action, nature's action, and the current state, the next state is deterministically specified. During execution, however, the robot will not know the action of nature. A specific example of a state transition equation is given in Section 3.5.

If nondeterministic uncertainty is considered, the state transition equation can be applied to obtain the following subset of X:

$$F_{k+1}(x_k, u_k) = \{f(x_k, u_k, \theta_k^a) \in X | \theta_k^a \in \Theta^a\}. \tag{2}$$

This set represents the possible next states that can result from a single application of the state transition equation.

Under probabilistic uncertainty, a probability density function (pdf) $p(\theta_k^a)$, is assumed to be known. For this probability density and the remaining probability densities in the paper, we implicitly assume there is some underlying probability space, and random variables with densities are constructed using appropriate measurability conditions. By using the state transition equation, a pdf for x_{k+1} can be inferred, which is represented by $p(x_{k+1} \mid x_k, u_k)$.

In general, let F refer to minimal subsets of X that can be inferred from the arguments. The role of F in our expression for nondeterministic uncertainty can be considered analogous to the role of p in probabilistic expressions. Thus, F is a generic representation for a subset of X, whereas p is a generic representation for a density on X. This will help clarify the relationships between nondeterministic and probabilistic uncertainties.

3.2. Imperfect Sensing and Information Spaces

In this section, we consider uncertainty in sensing, which implies that the current state is not known by the robot, and actions must be chosen on the basis of imperfect information. Therefore, actions taken by the robot will be conditioned on an information space as opposed to the state space. This information space concept has been adapted from stochastic control (Kumar and Varaiya 1986) and dynamic game theory (Başar and Olsder 1982) to fit our particular context.

3.2.1. Sensor Observations

We begin by defining a general model of robot sensing. A sensor can be viewed as a mapping from states onto sensor values with potential interference that is caused by nature. At every stage k, the robot makes an observation that is governed by the equation

$$y_k = h_k(x_k, \theta_k^s), \tag{3}$$

which we term the observation equation. A specific example of an observation equation is given in Section 3.5. An extension of this model could alternatively be defined that allows delayed and correlated measurements (LaValle 1995), but this will not be addressed in this paper.

The values y_k belong to a sensor space, denoted by Y. This model indicates that the robot receives information at every possible stage; however, this assumption can be relaxed. For example, in visual servo control applications, the servo rate for the robot joint controllers is typically much faster than the sampling rate of the vision system (e.g., see Mahadevamurty, Tsao, and Hutchinson 1994). It might also be the case that a sensor only provides information at randomly chosen stages (as is the case for the visual servo system reported in Feddema and Mitchell 1989, in which the vision system's sampling rate varies according to the amount of processing required to track moving objects in the scene).

For nondeterministic uncertainty, the set of possible values for x_k after only observing y_k can be determined from the observation equation as

$$F_k(y_k) = \{x_k \in X | y_k = h(x_k, \theta_k^s), \theta_k^s \in \Theta^s\}. \tag{4}$$

Under probabilistic uncertainty, we assume that $p(\theta_k^s)$ is known, and a pdf for x_k can be inferred, which is represented by $p(x_k \mid y_k)$. As a simple example, h could represent a position sensor that measures x_k with Gaussian noise. If $h(x_k, \theta_k^s) = x_k + \theta_k^s$, and $p(\theta_k^s)$ is a Gaussian density, then $p(x_k \mid y_k)$ is Gaussian.

If $Y = X$, and h_k is reduced to the identity map from X to Y, then the sensing model reduces to perfect state information. Equation (3) represents the output equation used in control theory; it is also similar to the projection of world states onto sensor values used in previous robotics contexts (e.g., Donald and Jennings 1991).

3.2.2. History and Information

Several approaches to manipulation planning have incorporated sensor information. Erdmann (1993) presented an approach that yields motion strategies that are conditioned on knowledge states. These knowledge states are inferred from the sensing history and at a high level correspond closely to using information feedback. In Latombe, Lazanas, and Shekhar (1991), sensing and control history is used to infer a subset of configuration space defined as a goal kernel in which the robot can successfully switch between commands in a multiple step plan or terminate in the goal region. In Goldberg (1990), it is assumed that no sensor information is available, and squeezing operations are conditioned on the history of previous operations (which is equivalent to the control history). Donald and Jennings (1991) have defined perceptual equivalence classes that determine distinguishable scenarios for a mobile robot based on its sensing history and the projection from the configuration space onto the sensor space.

The following definitions precisely describe the sensing and action history available to the robot. For a given stage k, let η_k denote some subset

$$\eta_k \subseteq \{u_1, u_2, \ldots, u_{k-1}, y_1, y_2, \ldots, y_k\}. \tag{5}$$

The value η_k is a set of past actions and observations that are known to the robot at stage k and is termed the information state. As an example, we could define a sensorless robot as considered in Goldberg (1990) in which $\eta_k = \{u_1, \ldots, u_{k-1}\}$.

The set of values that η_k can assume is denoted by N_k and is termed the information space. We define an information structure as the set of N_k for all $1 \leq k \leq K$.

3.3. Representations of the Information State

In this section, we present alternative ways to interpret the information space. Consider the case in which the robot has perfect memory. Each η_k then corresponds to a complete history of previous robot actions and observations. If U is n_1-dimensional and Y is n_2-dimensional, then in general the dimension of N_k will be $[k(n_1 + n_2) + n_2]$. A space that grows significantly with each stage (and becomes infinite-dimensional when $K = \infty$) is very unappealing for designing strategies.

3.3.1. The Case of Nondeterministic Uncertainty

An alternative to maintaining a growing history is to consider subsets of X that represent the possible current states x_k for a given information space value η_k. We will represent the minimal subset of X that can be inferred from η_k as $F_k(\eta_k)$. In other words, η_k as $F_k(\eta_k)$ represents the set of all states x_k that could possibly be the true system state, given the history η_k.

The information state subset $F_{k+1}(\eta_{k+1})$ can be determined from $F_k(\eta_k)$ when u_k and y_{k+1} are given. Initially, we have $F_1(\eta_1) \subseteq X$. Suppose inductively that we have $F_k(\eta_k)$. Recall from (2) that $F_{k+1}(x_k, u_k)$ represents the possible values x_{k+1} that could be obtained through a single application of the state transition equation. We can define

$$F_{k+1}(\eta_k, u_k) = \bigcup_{x_k \in F_k(\eta_k)} F_{k+1}(x_k, u_k). \qquad (6)$$

Recall that η_{k+1} can be specified with η_k, u_k, and y_{k+1}. Recall from (4) that $F_k(y_k)$ denotes the set of possible values for x_k after only observing y_k. By maintaining consistency with the observation of y_{k+1}, the following can be obtained:

$$F_{k+1}(\eta_{k+1}) = F_{k+1}(y_{k+1}) \bigcap F_{k+1}(\eta_k, u_k), \qquad (7)$$

which depends on (2) and (4). If the robot does not have perfect memory, then the condition $\{\eta_k, u_k\}$ is replaced in (7) by the appropriate subset of history.

3.3.2. The Case of Probabilistic Uncertainty

Under probabilistic uncertainty, the information state can be considered as a conditional density on the state space, denoted as $p(x_k \mid \eta_k)$. By using this approach, the information state density $p(x_{k+1} \mid \eta_{k+1})$, can be determined from $p(x_k \mid \eta_k)$ when u_k and y_{k+1} are given. This observation allows the development of several well-known stochastic control results, such as the Kalman filter, when all densities in the information space take some parametric form of fixed, low dimension.

We briefly indicate how the information state density is obtained. These equations can be considered as probabilistic versions of the nondeterministic results. Initially, we have $p(x_1 \mid \eta_1)$. We can derive an expression for $p(x_{k+1} \mid \eta_{k+1})$ in terms of $p(x_k \mid \eta_k)$, u_k, and y_{k+1}. Suppose inductively that we have $p(x_k \mid \eta_k)$. First, consider the effect on the state space of using the action u_k. Using the density from the state transition equation, we obtain, through marginalization with respect to X_k,

$$p(x_{k+1}|\eta_k, u_k) = \int p(x_{k+1}|x_k, u_k)p(x_k|\eta_k)dx_k. \qquad (8)$$

Recall from Section 3.1 that $p(x_{k+1} \mid x_k, u_k)$ is inferred from the state transition equation. Note that η_{k+1} can be specified with η_k, u_k, and y_{k+1}. By using Bayes's rule on X_{k+1} and Y_{k+1}, the following can be obtained:

$$p(x_{k+1}|\eta_{k+1}) = \frac{p(y_{k+1}|x_{k+1})p(x_{k+1}|\eta_k, u_k)}{\int p(y_{k+1}|x_{k+1})p(x_{k+1}|\eta_k, u_k)dx_{k+1}}, \quad (9)$$

which is a function of $p(y_{k+1} \mid x_{k+1})$, as defined in Section 3.2. A more detailed discussion of (9) can be found in Kumar and Varaiya (1986). If the robot does not have perfect memory, then the condition $\{\eta_k, u_k\}$ is replaced in (9) by the appropriate subset of history.

3.4. Strategy Concepts

3.4.1. Motion Strategies

At first it might seem appropriate to define some action u_k for each stage. In general, due to the control uncertainty, it is not possible to predict the trajectory of the robot for given motion commands. It is therefore advantageous to allow the robot to respond to information that becomes available during execution.

We consider robot strategies for two cases: perfect information and imperfect information. Suppose that the robot has perfect state information. We can implement a state-feedback strategy at stage k as a function $g_k : X \rightarrow U$. For each state x_k, a strategy yields an action $u_k = g_k(x_k)$. The set of mappings $\{g_1, g_2, \ldots, g_K\}$ is denoted by g and termed a (robot) strategy.

If the robot does not have direct access to state information, its actions are instead conditioned on the information state. In this case, we define a strategy at state k of the robot as a function $g_k : N_k \rightarrow U$. For each information state η_k, a strategy yields an action $u_k = g_k(\eta_k)$. In a sense, the "planning" actually occurs in this information space. These strategy concepts are equivalent to a feedback control law (Bertsekas 1987) and are similar to a conditional multistep plan in manipulation planning (Latombe, Lazanas, and Shekhar 1991).

We also define a strategy γ^θ for nature. Since nature is considered as a decision maker that can interfere with the robot, we allow nature's actions to depend in general on the state x_k and the action u_k of the robot. We can define a pure or deterministic strategy for nature as a deterministic mapping at each stage as $\gamma_k^\theta : X \times U \rightarrow \Theta^a$. Under nondeterministic uncertainty, we will assume that nature implements a deterministic strategy that is unknown to the robot. We will use the notation Γ^Θ to refer to the space of strategies that are available to nature under nondeterministic uncertainty.

Under probabilistic uncertainty, we consider a randomized or mixed strategy for nature in which the action of nature is represented by a pdf, $p(\theta_k)$ (or we can more generally consider $p(\theta_k \mid x_k, u_k)$). The specific action of nature at stage k is denoted by θ_k, sampled from the random variable Θ_k.

Therefore, the robot is given a pdf $p(\theta_k)$ that characterizes the action taken by nature at stage k. Although the randomized strategy is known by the robot, the actions that will be chosen are sampled from a random variable at each stage.

3.4.2. Termination Conditions

The decision to halt the robot has been given careful attention in manipulation planning research, particularly when there is uncertainty in the sensed current configuration. A motion plan might bring the robot into a goal region (reachability), but the robot may not halt if it does not realize that it is in the goal region (recognizability) (Erdmann 1984). Under nondeterministic uncertainty, it is said that a plan achieves a goal if the robot is guaranteed to halt in the goal region. The same concept is needed in our context; hence, we define a termination condition TC_k at each stage by a binary-valued mapping

$$TC_k : N_k \rightarrow \{true,\ false\}, \tag{10}$$

which is similar to the general form for the termination condition in Latombe, Lazanas, and Shekhar (1991). With perfect state information, N_k is simply replaced by X in (10). We require that if $TC_k = true$, then $TC_{k+1} = true$.

Let TC denote the complete specification of TC_k for all k. The termination condition is implemented so that the robot terminates at some stage $k \leq K+1$, making the specific choice of K not important except that it be sufficiently large. We will use the notation γ to denote the pair (g, TC), which can be considered as a strategy with termination condition. This pairing is similar to the concept of a motion command as defined in Latombe, Lazanas, and Shekhar (1991). We will use the notation Γ to denote the set of all γ that are available to the robot. It can also been seen that the use of this termination condition in the determination of an optimal strategy is equivalent to an optimal stopping rule (from optimal control theory; Bertsekas 1987).

3.4.3. Loss Functionals

We encode the objectives that are to be achieved by a nonnegative real-valued functional

$$L(x_1, \ldots, x_{K+1}, u_1, \ldots, u_K, TC),$$

called the loss functional, which is assumed to be of a stage-additive form that is often used in optimal control theory:

$$L(x_1, \ldots, x_{K+1}, u_1, \ldots, u_K, TC) = \sum_{k+1}^{K} l_k(x_k, u_k, TC_k) + l_{K+1}(x_{K+1}). \tag{11}$$

The first K terms correspond to costs that are received at each step during execution of the strategy. The final term l_{K+1} is a final cost that can be used to indicate the importance of terminating in the goal. This form is quite general, and facilitates the application of the dynamic programming principle, as discussed in Section 6.

Next, we present two useful loss functionals. Let $G \subset X$ represent a goal region in the state space. The following loss functional distinguishes only between success and failure to achieve the goal:

$$L(x_1, \ldots, x_{K+1}, u_1, \ldots, u_K, TC) = \begin{cases} 0 & \text{if } x_{K+1} \in G \\ 1 & \text{otherwise} \end{cases}. \tag{12}$$

Under probabilistic uncertainty, the evaluation of this criterion under a given strategy yields the probability of success (in the same manner that a 0-1 loss results in the probability of an incorrect decision in Bayesian decision theory; Devijver and Kittler (1982)).

Often, we will want to consider the cumulative cost of executing motions. Under the bounded velocity assumption, the following loss functional can measure the length of the executed trajectory:

$$L(x_1, \ldots, x_{K+1}, u_1, \ldots, u_k, TC) = \begin{cases} \sum_{k=1}^{K+1} l(u_k, TC_k) & \text{if } x_{K+1} \in G \\ C_f & \text{otherwise} \end{cases}. \tag{13}$$

Above, $l(u_k, TC_k)$ denotes the cost associated with taking action u_k, and we require that $l(u_k, TC_k) = 0$ if $TC_k = true$. Hence, loss does not accumulate after the robot has terminated. We use C_f to express how important it is to achieve the goal. If C_f becomes less than a typical aggregate action cost that achieves the goal, then strategies will be preferred that do not even expect to achieve the goal.

3.5. Specific Model Details

In this section, we present specific definitions of a state transition equation and an observation equation. These models are inspired by those used in previous manipulation planning research, and are used for our examples throughout this paper. Variants of this model have been considered for object manipulation tasks (Erdmann 1984) and mobile robot navigation (Latombe, Lazanas, and Shekhar 1991). In general, a variety of other types of models could be defined; however, the present model facilitates the comparison with previous motion planning research.

3.5.1. A Control Model

Suppose the robot is a polygon translating in the plane amid polygonal obstacles. The action set of the robot is a set of commanded velocity directions which can be specified by an orientation, yielding $U = [0, 2\pi)$. The robot will attempt to move a fixed distance $|| \ v \ || \ \Delta t$ (expressed in terms of a constant velocity modulus $|| \ v \ ||$) in the direction specified by u_k. The action space of nature is a set of angular displacements θ_k^a such that $-\epsilon_\theta \leq \theta_k^a \leq \epsilon_\theta$ for some maximum angle ϵ_θ. Under nondeterministic uncertainty, any action $\theta_k^a \in [-\epsilon_\theta, \epsilon_\theta]$ can be chosen by nature. When using probabilistic uncertainty, $p(\theta_k^a)$ could be a continuous pdf, which is zero outside of $[-\epsilon_\theta, \epsilon_\theta]$.

There are several cases to consider in defining the state transition equation f. Force compliant control is usually permitted in manipulation planning, which enables the robot to move along obstacle boundaries (Mason 1982; Paul and Shimano 1976; Raibert and Craig 1981; Shan and Koren 1995; Whitney 1985), and we use the generalized damping model, which has been considered most often in this context (Latombe 1991; Whitney 1977). First, consider the state transition equation when $x_k \in \mathcal{C}_{\text{free}}$ at a distance of at least $|| \ v \ || \ \Delta t$ away from the obstacles. If the robot chooses action u_k from state x_k, and nature chooses θ_k^a, then x_{k+1} is given by

$$f(x_k, u_k, \theta_k^a) = x_k + ||v||\Delta t \begin{bmatrix} \cos(u_k + \theta_k^a) \\ \sin(u_k + \theta_k^a) \end{bmatrix}. \quad (14)$$

Let $\mathcal{C}_{\text{contact}}$ represent the boundary of $\mathcal{C}_{\text{free}}$ (hence, $\mathcal{C}_{\text{contact}} = \mathcal{C}_{\text{valid}} - \mathcal{C}_{\text{free}}$). If $x_k \in \mathcal{C}_{\text{contact}}$ with a distance of at least $|| \ v \ || \ \Delta t$ from the edge endpoints, then a compliant motion is generated by using the generalized damper model (e.g., see Whitney 1977) for certain choices of u_k. If u_k points into the obstacle edge with a sufficient angle to overcome friction, then the robot moves a fixed distance parallel to the edge. Otherwise, the robot either remains fixed or moves away into $\mathcal{C}_{\text{free}}$. The remaining cases describe when the robot moves from $\mathcal{C}_{\text{free}}$ to $\mathcal{C}_{\text{contact}}$, from $\mathcal{C}_{\text{contact}}$ to $\mathcal{C}_{\text{free}}$, or from one edge in $\mathcal{C}_{\text{valid}}$ to another. Note that in our model, nature repeatedly acts at each time Δt, as opposed to only once as in Brost and Christiansen (1996); Latombe, Lazanas, and Shekhar (1991); and Lozano-Pérez, Mason, and Taylor (1984).

3.5.2. A Sensing Model

We now present a sensing model that is similar to that used in Brost and Christiansen (1996); Erdmann (1984); and Latombe, Lazanas, and Shekhar (1991). This sensing model will be used in Section 6.6. The robot is equipped with a position sensor and a force sensor. Assume that the position sensor is calibrated in the configuration space, yielding values in $\Re^2$. The force sensor provides values in $[0, 2\pi) \cup \{\emptyset\}$, indicating either the direction of force or no force (represented by $\emptyset$).

We consider independent portions of the observation equation: h^p for the position sensor and h^f for the force sensor (which together form a three-dimensional vector-valued function). We partition the sensing action of nature θ_k^s into subvectors $\theta_k^{s,p}$ and $\theta_k^{s,f}$, which act on the position sensor and force sensor, respectively. The observation for the position sensor is $y_k^p = h^p(x_k, \theta_k^{s,p}) = x_k + \theta_k^{s,p}$. Under nondeterministic uncertainty, $\theta_k^{s,p}$ could be any value in $\Theta_k^{s,p}$. If probabilistic uncertainty is used, we could provide a density for nature as

$$p(\theta_k^{s,p}) = \begin{cases} \frac{2}{\pi\epsilon_p^2} & \text{for } ||\theta_k^{s,p}|| < \epsilon_p \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

for some prespecified radius ϵ_p, and $\theta_k^{s,p}$ is two dimensional.

For the force sensor, we obtain either (1) a value in $[0, 2\pi)$, governed by $y_k^f = h^f(x_k, \theta_k^{s,f}) = \alpha(x_k) + \theta_k^{s,f}$, in which $x_k \in \mathcal{C}_{\text{contact}}$, and the true normal is given by $\alpha(x_k)$, or (2) an empty value $\emptyset$ when the robot is in $\mathcal{C}_{\text{free}}$. When the robot configuration lies in $\mathcal{C}_{\text{contact}}$ and probabilistic uncertainty is in use, then we might choose

$$p(\theta_k^{s,p}) = \begin{cases} \frac{1}{2\epsilon_f} & \text{for } |\theta_k^{s,f}| < \epsilon_f \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

for some positive prespecified constant $\epsilon_f < \frac{1}{2}\pi$. We consider the random variables of $\theta_k^{s,p}$ and $\theta_k^{s,f}$ to be independent and identically distributed over all stages. In some applications, sensing errors might not be stage independent, in which case these random variables become conditioned on sensing and action history. Additional state parameters could also be included that allow the sensing model to vary, although this leads to higher complexity.

4. Forward Projections

In this section, we present forward projections for each of the four uncertainty cases that are considered in this paper. A forward projection is used to characterize the possible future states under the implementation of a strategy from an initial state. The forward projection concepts presented here are based on forward projection concepts that have appeared in manipulation planning research (Brost and Christiansen 1996; Erdmann 1984). In our work, the forward projections result from the implementation of a strategy γ. We conclude this section by presenting some computed examples of forward projections.

4.1. Nondeterministic Forward Projections

4.1.1. The Perfect Information Case

We use the notation $F_j(x_i, g)$ to denote the minimal subset of X that is guaranteed to contain x_j if the system begins in state x_i at stage i, and strategy g is implemented up to stage j.

Assume that some g is given, and that at stage k, the state x_k is known. The action taken by the robot at stage k is known to be $u_k = g_k(x_k)$. Therefore, we can write

$$F_{k+1}(x_k, g) = F_{k+1}(x_k, g_k(x_k)) = F_{k+1}(x_k, u_k), \quad (17)$$

in which $F_{k+1}(x_k, u_k)$ is given by (2). Although the action is known, the resulting next state x_{k+1} is nondeterministic because of nature, $\theta_k^a \in \Theta^a$.

Suppose that we wish to determine the outcome at stage x_{k+2} if we know x_k. From (2), we already know that $x_{k+1} \in F_{k+1}(x_k, u_k)$. The nondeterministic action of nature at stage $k+1$ must be taken into account to yield

$$F_{k+2}(x_k, g) = \{f(x_{k+1}, u_{k+1}, \theta_{k+1}^a) \in X | x_{k+1} \in F_{k+1}(x_k, g), \theta_{k+1}^a \in \Theta^a\}. \quad (18)$$

This forward projection can also be expressed with a set union as

$$F_{k+2}(x_k, g) = \bigcup_{x_k+1 \in F_{k+1}(x_k, g)} F_{k+2}(x_{k+1}, g). \quad (19)$$

The forward projection for a finite number of stages from stage 1 can be considered as an iterated union

$$F_k(x_1, g) = \bigcup_{x_2 \in F_2(x_1, g)} \bigcup_{x_3 \in F_3(x_2, g)} \cdots \bigcup_{x_{k-1} \in F_{k-1}(x_{k-2}, g)} F_k(x_{k-1}, g), \quad (20)$$

which is an extension of (19). The projection from any stage k to stage $k+N$ can be similarly defined.

The next step is to include the termination condition to determine a forward projection for γ as opposed to g. Assume that the robot remains motionless after TC becomes *true*. Hence, the effect of the termination condition is equivalent to considering the resulting location of the robot at stage $K+1$ (assuming that under all possible trajectories, the termination condition was met before $K+1$). This results in $F_{K+1}(x_1, \gamma)$, which can be constructed by replacing g with γ in eqs. (17) to (20).

The classical reachability and recognizability concepts (Erdmann 1993) can be defined using our framework. We say that the goal is reachable at stage k under γ if $F_k(x_1, g) \subseteq G$. In other words, if the strategy is guaranteed to bring the robot into the goal region for some k, then reachability at stage k holds. We can also define a reachability that does not depend on k. We can say that the goal is reachable if for every possible state trajectory $\{x_1, \ldots, x_{K+1}\}$ (under the implementation of a given g), there exists a k such that $x_k \in G$.

A stronger condition is that the goal is recognizably achieved under γ, which means $F_{K+1}(x_1, \gamma) \subseteq G$. This condition implies that the robot is guaranteed to terminate in the goal region.

4.1.2. *The Imperfect Information Case*

The previous forward projection (20) provided a subset of X in which the system state will lie after the execution of a strategy. With imperfect sensing, we can consider the motions to occur in the information space. In fact, we can consider the information space as a new "state space" in which there is perfect "state" information. For this reason, a forward projection can also be defined directly on the information space.

It is assumed for the forward projection that the history has not yet been given. Suppose that an information state $\eta_k \in N_k$ is given. Under the implementation of g, the action $u_k = g_k(\eta_k)$ is known.

We now define the information forward projection for a single stage. This will be an intermediate concept that is used to define the forward projection as a subset of the state space. We have previously used F to represent a subset of X, and we will use $\tilde{F}$ to refer to a subset of the information space. After applying an action u_k and receiving sensor observation y_{k+1}, we obtain

$$\begin{aligned}\tilde{F}_{k+1}(\eta_k, g(\eta_k)) &= \tilde{F}_{k+1}(\eta_k, u_k) \\ &= \{\eta_{k+1} \in N_{k+1} | \eta_k \cup \{u_k, y_{k+1}\} \\ &\subset \eta_{k+1}, x_{k+1} \in F_{k+1}(x_k, u_k) \\ &\cap F_{k+1}(y_{k+1}), x_k \in F_k(\eta_k)\},\end{aligned} \quad (21)$$

which depends on (2) and (4), and $F_k(\eta_k) \subseteq X$ is the subset representation of the information state from Section 3.2. This yields the set of all next information states that share a common history with the current information state, and are consistent with the state transition equation and next sensor observation.

To obtain the information forward projection from stage 1 to some stage k, we can iteratively apply (21).

The information forward projection can be mapped to subsets of the state space. For a given $\tilde{F}_k(\eta_1, g)$, the subset of X in which the system state will lie is

$$\bigcup_{\eta_k \in \tilde{F}_k(\eta_1, g)} F_k(\eta_k). \quad (22)$$

The goal is reachable at stage k if the set defined in (22) is a subset of G. As in Section 4.1.1, we can replace g with γ in the above expressions to yield the forward projection with termination condition $F_{K+1}(\eta_1, \gamma)$. Hence, recognizability can also be defined.

4.2. *Probabilistic Forward Projections*

We use the notation $p(x_j \mid x_i, g)$ in this section to represent the density that is obtained if the system begins at state x_i at stage i and strategy g is implemented. This density follows directly from the state transition equation and the densities for nature of the form $p(\theta_k^a)$.

4.2.1. *The Perfect Information Case*

The following development parallels the development of the forward projection in Section 4.1.1. Assume that some g is given, and that at stage k, the state x_k is known. The action taken by the robot at stage k is known to be $u_k = g_k(x_k)$. Therefore, we can write

$$p(x_{k+1}|x_k, g) = p(x_{k+1}|x_k, g_k(x_k)) = p(x_{k+1}|x_k, u_k). \tag{23}$$

Recall from Section 3.1 that $p(x_{k+1} \mid x_k, u_k)$ can be determined from the state transition equation.

Next, consider predicting the outcome at stage $k+2$ if we begin at stage k and apply g:

$$p(x_{k+2}|x_k, g) = \int p(x_{k+2}|x_{k+1}, g_{k+1}(x_{k+1})) \times p(x_{k+1}|x_k, g_k(x_k))dx_{k+1}. \tag{24}$$

The result after applying two actions is a posterior density on X.

The forward projection for a finite number of stages from stage 1 results in the posterior

$$p(x_k|x_1, g) = \int p(x_k|x_{k-1}, g_{k-1}(x_{k-1})) \times p(x_{k-1}|x_{k-2}, g_{k-2}(x_{k-2})) \cdots p(x_2|x_1, g_1(x_1))dx_2dx_3 \cdots dx_{k-1}. \tag{25}$$

The projection from any stage k to stage $k+N$ can be similarly defined.

The next step is to include the termination condition to determine a forward projection for γ as opposed to g. We can replace g with γ in the conditions above and define $p(x_{K+1} \mid x_1, \gamma)$ by using the assumption that the robot remains motionless after the termination condition becomes *true*.

We can now define probabilistic notions of reachability and recognizability. The probability that the goal is reached at stage k is given by

$$\int_G p(x_k|x_1, g)dx_k, \tag{26}$$

in which the region of integration is the goal region $G \subseteq X$.

The probability that the goal is recognizably achieved is

$$\int_G p(x_{K+1}|x_1, \gamma)dx_{K+1}. \tag{27}$$

4.2.2. The Imperfect Information Case

In this section, we develop the forward projections for the case in which there is probabilistic uncertainty in both sensing and control. The forward projection for this case will be considered as a density on X, which is conditioned on a particular strategy and initial state (either x_1 or η_1). This density indicates where the robot will be likely to end up when a fixed γ is implemented, either after TC is satisfied or at some specified stage. Note that we could also derive $p(\eta_k \mid \eta_1, \gamma)$, resulting in a pdf on the information space.

At stage k, the density on X after starting at η_1 is given by

$$p(x_k|\eta_1, g) = \int p(x_k|\eta_{k-1}, g_{k-1}(\eta_{k-1})) \times p(\eta_{k-1}|\eta_{k-2}, g_{k-2}(\eta_{k-2})) \cdots p(\eta_2|\eta_1, g_1(\eta_1))d\eta_{k-1} \cdots d\eta_2. \tag{28}$$

The first term in the integrand can be determined using (8). Each of the remaining terms can be reduced to

$$p(\eta_{k+1}|\eta_k, g_k(\eta_k)) = p(y_1, \ldots, y_{k+1}, u_1, \ldots, u_k|y_1, \ldots, y_k, u_1, \ldots, u_k) = p(y_{k+1}, u_k|\eta_k, u_k). \tag{29}$$

This reduction occurs because most of the sensing and action history appears on both sides of the density expression. The right side of (29) can be further reduced to

$$p(y_{k+1}|\eta_k, u_k) = \int p(y_{k+1}|x_{k+1})p(x_{k+1}|\eta_k, u_k)dx_{k+1} = \int\int p(y_{k+1}|x_{k+1})p(x_{k+1}|x_k, u_k) \times p(x_k|\eta_k)dx_kdx_{k+1} \tag{30}$$

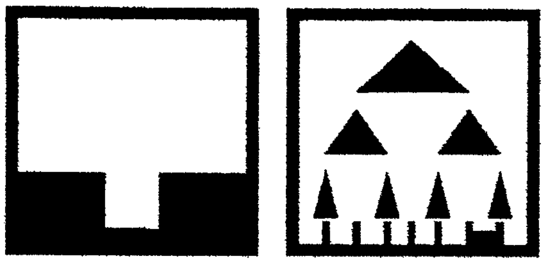
in which all three terms in the final integrand are known. The density $p(y_{k+1} \mid x_{k+1})$ is inferred from the sensing model; $p(x_{k+1} \mid x_k, u_k)$ is inferred from the control model; $p(x_k \mid \eta_k)$ is the density representation of the current information state.

To include the termination condition, we replace g by γ above to obtain $p(x_{K+1} \mid \eta_1, \gamma)$. Reachability and recognizability can be defined in a manner similar to that in Section 4.2.1.

4.3. Computed Examples

In this section, we present computed examples that illustrate the forward projection concepts. These forward projections are provided under the assumption that constant motion commands are given to the robot. In other words, some $u \in U$ is chosen, and a strategy is defined as $\gamma_k \equiv u$ for all $k \in \{1, \ldots, K\}$. This will make the comparison of our forward projections to previous research more clear. In Section 6.6, we will present forward projections that are obtained under the implementation of the optimal strategies, as computed by our algorithms.

We have computed forward projection examples in a straightforward way by using a discretized array representation for the state space. Under nondeterministic uncertainty, this can be considered as a bitmap representation of the forward projection. Under probabilistic uncertainty, the representation approximates a pdf on X by using a fine grid.



a. b.

Fig. 2. Two examples that are considered in this section. The obstacles in the workspace are indicated by gray regions, and the black region represents the goal.

In the first step of the computation, the array is initialized to reflect the uncertainty associated with the initial state. At each additional step, the forward projection for the next stage is represented in a new array, which is determined by applying the given strategy to the elements in the previous array and using linear interpolation (in the implementation, only two copies of the array are needed at any given time). We have found this computational technique to produce reasonable representations of forward projections.

For the examples that are considered in this paper, we assume a two-dimensional, bounded state space (i.e., $\mathcal{C}_{\text{valid}} \subseteq \Re^2$), in which each coordinate is constrained to lie in the interval [0, 100]. This could, for example, represent the configuration space of a planar Cartesian manipulator with two translational degrees of freedom. The obstacles in the workspace will be indicated in figures by gray regions, and a black region will represent the goal.

The first example is depicted in Figure 2a and can be considered as a configuration space representation of the classical peg-in-hole problem (e.g., see Erdmann 1984; Brost and Christiansen 1996; Latombe, Lazanas, and Shekhar 1991; Lozano-Pérez, Mason, and Taylor 1984). The second example is depicted in Figure 2b and is designed to spread the possible locations of the robot over a large portion of the state space. The initial configuration x_1 for these two examples is (50,96). We use the control model that is discussed in Section 3.5 and assume that $\| v \| \Delta t = 3$, which implies that the robot is capable of moving three units at each stage. We assume that the maximum angular displacement that can be caused by nature is $\epsilon_\theta = 48.8°$. The given strategy is $\gamma_k \equiv \frac{3}{2}\pi$ for all $k \in \{1, \ldots, K\}$ (i.e., move down).

Figures 3 and 4 show forward projections under probabilistic uncertainty. For these examples, we assume that $p(\theta^a)$ is uniform on the interval $[-\epsilon_\theta, \epsilon_\theta]$. Initially, the pdf is sharply peaked; however, as control uncertainty accumulates, the density becomes more diffuse. Whenever compliance is possible, the density becomes narrower in the direction perpendicular to the edge. The compliant motions have the effect of funneling the probability mass into smaller regions. The pdf values become larger, since the density must integrate to 1. This effect can be seen in Figure 4, as a triangular obstacle causes the probability mass to divide. In the final stages, there is also a peaking effect; this corresponds to the robot sticking at some final state. Maximizing the probability that the goal will be achieved can be thought of as causing as much of the probability mass to stay in the goal as possible.

5. Performance Preimages

In this section, we present performance preimages for each of the four uncertainty cases considered in this paper. A performance preimage describes a region in the information space or state space from which the loss in achieving the goal lies within a set of values. This concept generalizes the notion of classical preimages to arbitrary performance measures, although the preimages are defined in discretized time in our framework. In the same way that classical preimages are useful for evaluating a motion command, the performance preimage is useful for evaluating a strategy. We conclude this section by presenting some computed examples of performance preimages and relating them to those in previous literature.

5.1. *Nondeterministic Performance Preimages*

In this section, we assume that nature implements a deterministic, unknown strategy γ^θ, as defined in Section 3.4.

5.1.1. The Perfect Information Case

Suppose that the strategy for nature γ^θ was given to the robot; then, the loss that would be received by choosing robot strategy γ could be expressed as $L(x_1, \gamma, \gamma^\theta)$, since the state trajectory can be deterministically predicted once x_1, γ, and γ^θ are given. This in turn implies that the action sequence $u_1, u_2, \ldots, u_K$ can also be predicted. Since the strategy of nature is not known by the robot, we define

$$\check{L}(x_1, \gamma) = \sup_{\gamma^\theta \in \Gamma^\theta} L(x_1, \gamma, \gamma^\theta), \tag{31}$$

which represents the maximum loss (given by the loss functional) that the robot could receive under the implementation of γ from x_1. This corresponds to modeling nature as an opponent, as is done in minimax design (Başar and Kumar 1987).

Recall that the classical preimage is a subset of X from which the robot is guaranteed to achieve the goal for a fixed motion command. Suppose that we are evaluating the trajectory of the robot with the loss functional (12) for a strategy that consists of a fixed action repeated at every stage (which

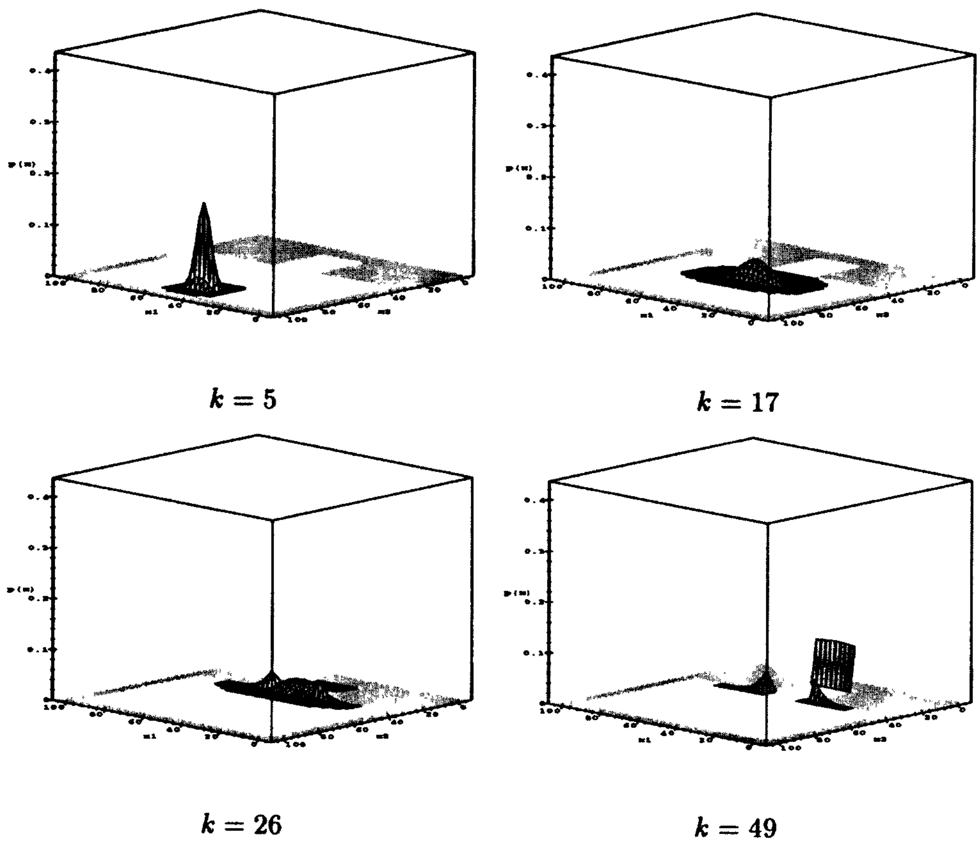


Fig. 3. The forward projection at several stages, under probabilistic uncertainty.

is equivalent to a fixed motion command). Elements $x_1 \in X$ such that $\check{L}(x_1, \gamma) = 0$ correspond to locations in the state space from which the robot is guaranteed to achieve the goal, and hence lie in the classical preimage.

Next, we generalize this classical preimage. Note that $\check{L}(x_1, \gamma)$ can be considered as a real-valued function of x_1 for a fixed γ. Consider some subset of the reals $R \subseteq \Re$. We define the performance preimage on X as a subset of X that is given by

$$\check{\pi}_x(\gamma, R) = \{x_1 \in X \,|\, \check{L}(x_1, \gamma) \in R\}. \tag{32}$$

The set $\check{\pi}_x(\gamma, R) \subseteq X$ indicates places in the state space from which if the robot begins, the loss will lie in R.

We can consider partitioning X into isoperformance classes by defining an equivalence class $\check{\pi}_x(\gamma, \{r\})$ for each $r \in [0, \infty)$. To shorten notation, we denote an isoperformance class $\check{\pi}_x(\gamma, \{r\})$ by $\check{\pi}_x(\gamma, r)$.

For the loss functional (12), $\check{\pi}_x(\gamma, 0)$ yields the classical preimage. Under (13) and $R = [0, m)$, we obtain a performance preimage that indicates all $x_1 \in X$ from which the loss is guaranteed to be less than m.

If we replace γ with g and replace the condition "if $x_{k+1} \in G$" in (12) with "if $x_k \in G$ for some k," then $\check{\pi}(g, 0)$ yields a backprojection that is similar to that appearing in Erdmann (1984).

5.1.2. The Imperfect Information Case

Let $L(\eta_1, \gamma, \gamma^\theta)$ represent the loss that is obtained if the robot implements γ and nature implements γ^θ. If x_1 cannot be recovered from η_1, γ, and γ^θ, then $L(\eta_1, \gamma, \gamma^\theta)$ must be considered as the supremum over all initial states that are consistent with η_1, γ, and γ^θ. The maximum amount of loss that the robot could receive under the implementation of γ while starting from η_1 is obtained by replacing x_1 with η_1 in (31). Note that in this case, γ^θ represents both control and sensing actions.

We define the performance preimage on N_1 as a subset of N_1, denoted by $\check{\pi}(\gamma, R)$, which is given by replacing x_1 with η_1 in (32).

5.2. Probabilistic Performance Preimages

In this section, we assume that nature chooses actions by sampling from a known pdf $p(\theta)$ which corresponds to a mixed strategy.

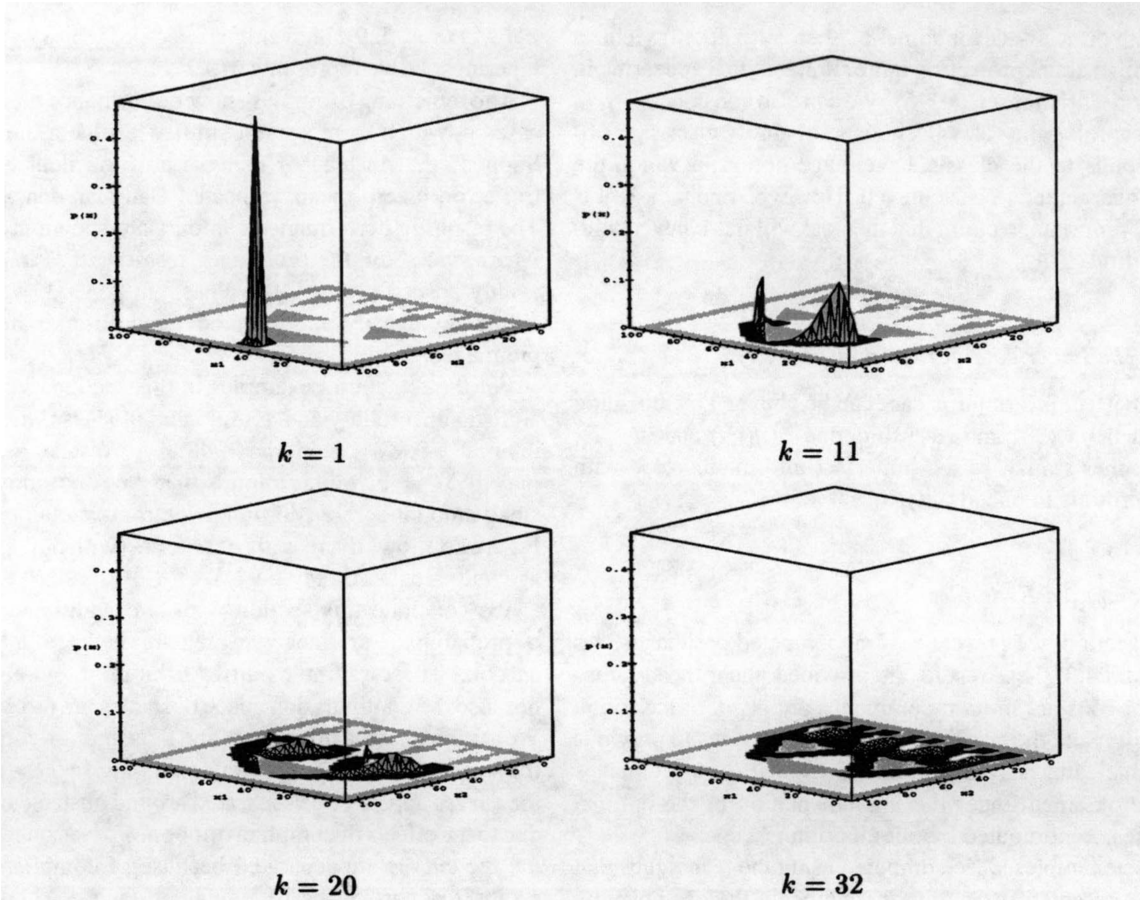


Fig. 4. The forward projection at several stages, under probabilistic uncertainty.

5.2.1. The Perfect Information Case

Suppose that we wish to evaluate some $\gamma = (g, TC)$ with a given initial state x_1. If θ is given along with x_1 and a strategy γ, the entire state trajectory $x_1, x_2, \ldots, x_{K+1}$ can be deterministically specified. We can therefore specify the loss for this trajectory as a function $L(x_1, \gamma, \theta)$. This is true because (1), (3), and x_k can be determined for every state when the value of nature's action θ is given.

The expected loss that we incur if γ is implemented can be expressed as

$$\bar{L}(x_1, \gamma) = \int L(x_1, \gamma, \theta) p(\theta) d\theta, \qquad (33)$$

in which θ represents the actions taken by nature over all stages. The integral considers each possible action sequence for nature θ weighted by the probability density value $p(\theta)$. For any given θ (along with γ and x_1), the action sequence $\{u_1, \ldots, u_K\}$, and state trajectory $\{x_1, \ldots, x_{K+1}\}$ can be completely determined, allowing the evaluation of the loss functional.

We observe for a fixed γ that $\bar{L}(x_1, \gamma)$ can be considered as a real-valued function of x_1. Consider some subset of the reals $R \subseteq \Re$. We define the performance preimage on X as a subset of X,

$$\bar{\pi}_x(\gamma, R) = \{x_1 \in X | \bar{L}(x_1, \gamma) \in R\}. \qquad (34)$$

The set $\bar{\pi}_x(\gamma, R) \subseteq X$ indicates places in the state space from which if the robot begins, the expected loss lies within R.

We now describe some particular choices for R. Suppose that $R = [0, r]$ for some $r \geq 0$ (recall that L is nonnegative). The performance preimage yields places in X from which the expected performance will be better than or equal to r. If $R = \{r\}$, for some point $r \geq 0$, we obtain places in X in which equal expected performance will be obtained. We can consider partitioning X into isoperformance classes by defining an equivalence class $\bar{\pi}_x(\gamma, \{r\})$ for each $r \in [0, \infty)$.

The loss functional (12) implies that we are only interested in achieving thc goal without any notion of efficiency in the actual robot trajectory. The loss $\bar{L}(x_1, \gamma)$ in this case represents the probability that the goal will not be achieved using γ. We consider some $\bar{\pi}_x(\gamma, [0, r])$ for $r \in [0, 1]$ as a probabilistic preimage on X. The probabilistic preimage thus indicates places in X from which the goal will be achieved with probability of at least $1 - r$. Furthermore, if we replace γ with g, and replace the condition "if $x_{k+1} \in G$" in

(12) with "if $x_k \in G$ for some k," then $\bar{\pi}_x(g, [0, r])$ yields a probabilistic backprojection quite similar to that appearing in Brost and Christiansen (1996). We can also consider $\bar{\pi}_x(\gamma, r)$ as an isoprobability class. The isoprobability class $\bar{\pi}_x(\gamma, 0)$ corresponds to the classical preimage notion in which the goal is guaranteed to be achieved. However, in our case, it is more appropriate to claim that the goal will be achieved with probability 1.

5.2.2. *The Imperfect Information Case*

The imperfect information case can be obtained by replacing x_1 with η_1 in (33) and (34) to define $\bar{L}(\eta_1, \gamma)$ and $\bar{\pi}(\gamma, R)$ in a manner similar to the imperfect information case with nondeterministic uncertainty.

5.3. Computed Examples

In this section, we present several computed preimages. As in Section 4.3, these results are provided under the assumption that constant motion commands are given to the robot. This will make the comparison of our preimages to previous research clearer, and in Section 6.6 we will show preimages that are obtained under the implementation of the optimal strategies, as computed by our algorithms.

These examples were computed using the techniques that will be presented in Section 6. We begin the examples by returning to the peg-in-hole problem, which was discussed in Section 4.3. Suppose that the fixed action is $\frac{3}{2}\pi$ and that the maximum angular displacement ϵ_θ is 14.3°. We will use (12) for all of the examples in this section, since we have observed that (13) produces very similar curves under fixed motion commands; when considering optimal strategies, however, the difference between the two loss functionals becomes much more significant.

Figure 5a shows a performance preimage under nondeterministic uncertainty. The subset of the state space that is below the curve corresponds to places in the state space from which the goal is guaranteed to be achieved. Note that this result does not depend on $\| v \| \Delta t$; this is because with nondeterministic uncertainty, the robot configuration can lie anywhere within the cone generated from the initial state and $\pm\epsilon_\theta$. The curve shown in Figure 5a corresponds closely to the classical preimage that has been determined for this problem in previous manipulation planning research (Erdmann 1984; Latombe 1991).

Figure 5b shows probabilistic backprojections which are quite similar to those in Brost and Christiansen (1996). We assume that $p(\theta^a)$ is uniform on the interval $[-\epsilon_\theta, \epsilon_\theta]$, and $\epsilon_\theta = 48.8°$. We assume that $\| v \| \Delta t = 200$, and let $K = 1$. There is only one decision-making stage, and the robot can move enough distance to accomplish the goal in a single stage. The figure shows isoprobability curves from $\bar{\pi}_x(0.2)$ to $\bar{\pi}_x(0.9)$, at evenly spaced probability values. The innermost curve represents $\bar{\pi}(0.2)$.

The remaining examples show performance preimages for cases in which there are not similar results in the literature. Suppose that instead of using a uniform density for control error, a zero-mean, truncated Gaussian density is used. The resulting performance contours are shown in Figure 5c. Again, we show the preimages from $\bar{\pi}_x(0.2)$ to $\bar{\pi}_x(0.9)$ at evenly spaced probability values. In general, we can substitute any density into the model and observe the resulting preimages.

For the remaining examples in this section, we use probabilistic uncertainty and assume that $p(\theta^a)$ is uniform on the interval $[-\epsilon_\theta, \epsilon_\theta]$, and $\epsilon_\theta = 48.8°$. We also assume that $\| v \| \Delta t = 3$, which implies that the robot moves only a small amount before additional control uncertainty is added. Figure 5d shows the resulting contours (with the same preimage values as used previously).

We conclude this section with two additional examples of probabilistic preimages, which are depicted in Figures 6a and 6b. These examples differ from the previous example because the configuration space obstacles are more complex. We assume that the initial state and fixed motion command are the same as used in the previous example. In these examples, the curves appear to be separated around obstacle boundaries due to the effects of compliant motion. For instance, in Figure 6a, the curves are separated because of compliant motions on the top part of the triangular obstacle. Although there is significant uncertainty in control, the edge of the triangle guides the robot into the goal region, significantly reducing the expected loss.

6. Designing Optimal Strategies

Sections 4 and 5 have presented methods that evaluate a given strategy. In this section, we define concepts of optimality, present a computational approach that selects an optimal strategy, and show computed examples.

6.1. Defining Optimality

The design problem is to select the most desirable strategy γ^* from the space of allowable strategies Γ. Under nondeterministic uncertainty, nature is considered as an opponent with diametrically opposed interests; therefore, a strategy is selected that minimizes the maximum amount of loss that could result from the strategy of nature. Under probabilistic uncertainty, the actions of nature can be characterized with probability densities; hence, a strategy is selected that minimizes the expected amount of loss.

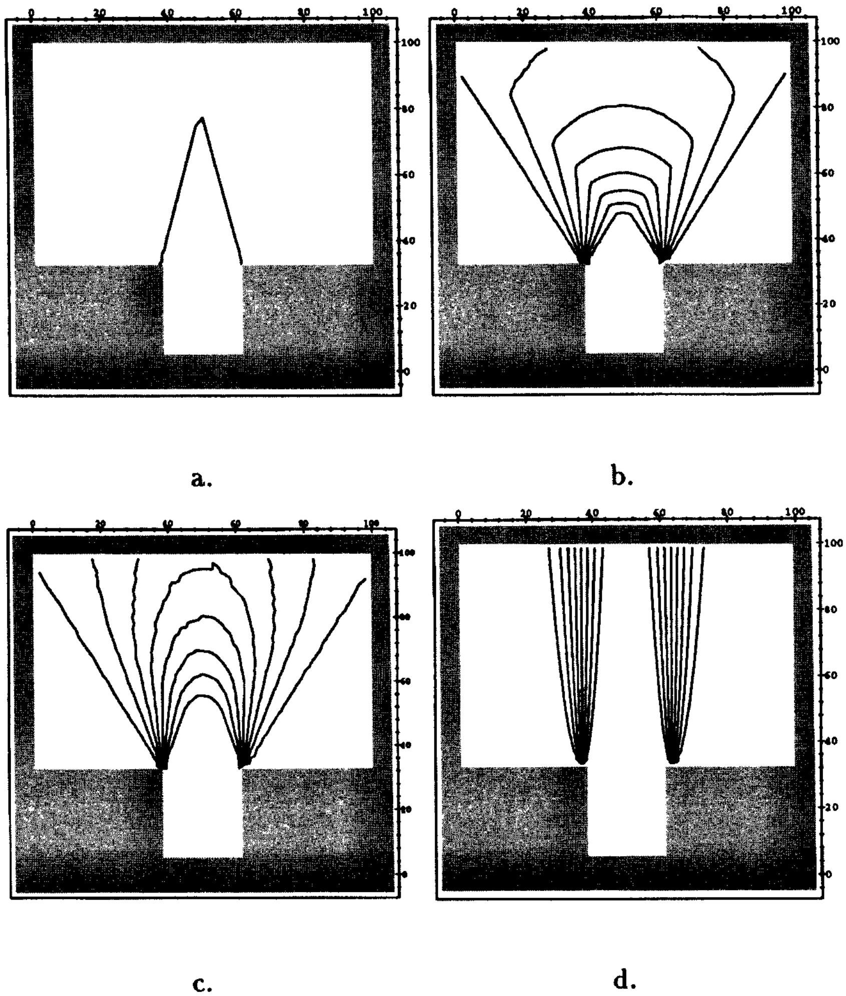


Fig. 5. Several computed performance preimages.

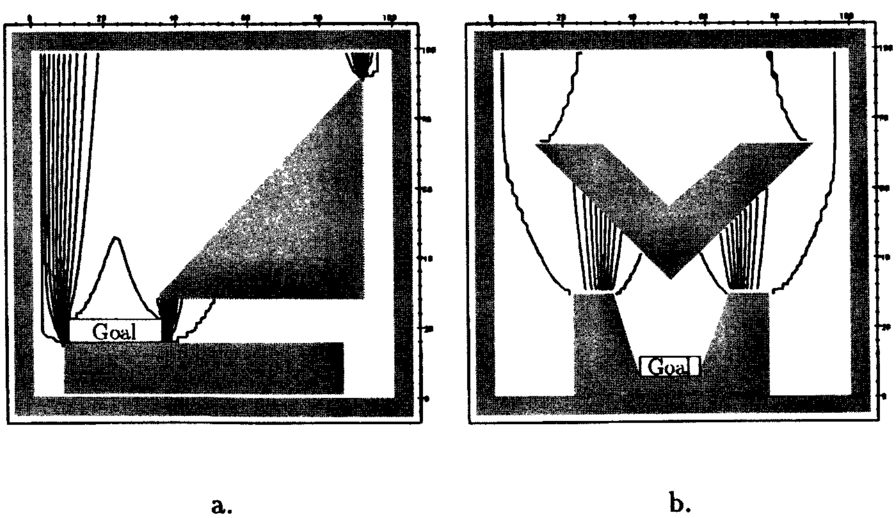


Fig. 6. Two computed performance preimages. In these examples, the obstacles displace the curves.

6.1.1. Optimality under Nondeterministic Uncertainty

6.1.1.1 Perfect Information

Recall that Γ^θ denotes the space of deterministic strategies for nature when nondeterministic uncertainty is considered. Under perfect information and nondeterministic control uncertainty, the ideal choice for a strategy $\gamma^* \in \Gamma$ satisfies

$$\check{L}(x_1, \gamma^*) = \inf_{\gamma \in \Gamma} \check{L}(x_1, \gamma) = \inf_{\gamma \in \Gamma} \sup_{\gamma^\theta \in \Gamma^\theta} L(x_1, \gamma, \gamma^\theta) \quad (35)$$

for all $x_1 \in X$. This indicates that from any initial state, the strategy will guarantee the least possible loss given the worst-case actions of nature. This concept has been used previously to design controllers based on worst-case analysis (Başar and Kumar 1987).

If we use the loss functional (12), the space of strategies Γ can be partitioned into two equivalence classes: those that are guaranteed to achieve the goal (resulting in a worst-case loss of zero) and those that may fail to achieve the goal (resulting in a worst-case loss of one). Any strategy in the first equivalence class satisfies (35) and directly corresponds to the common approach in previous manipulation planning research of selecting a strategy that is guaranteed to achieve the goal. By using another loss functional, such as (13), (35) can be considered as partitioning Γ into many more classes; this induces preferences on the set of strategies that achieve the goal.

6.1.1.2 Imperfect Information

The imperfect information case can be obtained by replacing x_1 with η_1 in (35) to define $\check{L}(\eta_1, \gamma^*)$.

6.1.2. Optimality under Probabilistic Uncertainty

6.1.2.1 Perfect Information

Recall that the actions of nature can be partially predicted through the specification of a pdf $p(\theta)$ in which θ represents the action θ_k of nature at every stage.

Under perfect information and probabilistic control uncertainty, the design task is to select a strategy $\gamma^* \in \Gamma$ such that

$$\bar{L}(x_1, \gamma^*) = \inf_{\gamma \in \Gamma} \bar{L}(x_1, \gamma) = \inf_{\gamma \in \Gamma} \int L(x_1, \gamma, \theta) p(\theta) d\theta \quad (36)$$

for all $x_1 \in X$. This corresponds to selecting a strategy that minimizes the loss in the expected sense, as considered in stochastic optimal control theory.

6.1.2.2 Imperfect Information

Under imperfect uncertainty, x_1 is replaced with η_1 (36) to define $\bar{L}(\eta_1, \gamma^*)$.

6.2. The Principle of Optimality

One powerful tool that underlies many of the solution techniques for dynamic decision-making problems is dynamic programming. The key is the principle of optimality, which states that an optimal solution can be recursively decomposed into optimal parts. In general, this optimization concept has been useful in a variety of contexts, both for producing analytical solutions and for numerical computation procedures.

The class of problems that can be analytically solved by using the principle of optimality is fairly restrictive, and there has been much focus on numerical dynamic programming procedures (Bertsekas 1987; Larson 1968; Larson and Casti 1982). Bertsekas (1987) considered solvable problems to typically be an exception in applications. In both control theory and game theory, the classic set of problems that can be solved are those with a linear state transition equation and quadratic loss functional (Anderson and Moore 1990; Başar and Olsder 1982; Bryson and Ho 1975). As an example, the principle of optimality forms the basis of the analytic solution to the linear-quadratic Gaussian (LQG) optimal control problem.

6.2.1. The Nondeterministic Case

6.2.1.1 Perfect Information

Suppose that for some k, the optimal strategy is known for each stage $i \in \{k, \ldots, K\}$. The optimal worst-case loss obtained by starting from stage k and implementing the portion of the optimal strategy $\{\gamma_k^*, \ldots, \gamma_K^*\}$ can be represented as

$$\check{L}_k^*(x_k) = \sup_{\gamma^\theta \in \Gamma^\theta} \left\{ \sum_{i=k}^{K} l_i(x_i, \gamma_i^*(x_i)) + l_{K+1}(x_{K+1}) \right\}, \quad (37)$$

in which Γ^θ refers to the set of possible choices for γ^θ. We use $\gamma_i^*(x_i)$ to represent the simultaneous choice of u_i^* and TC_i^*. Recall that under the implementation of a strategy, the state trajectory depends on the actions chosen by nature; therefore, the expression in the sup depends on nature. The function $\check{L}_k^*(x_k)$ is sometimes referred to as the cost-to-go function in dynamic optimization literature (Bertsekas 1987).

The principle of optimality (Başar and Olsder 1982) states that $\check{L}_k^*(x_k)$ can be obtained from $\check{L}_{k+1}^*(x_{k+1})$ by the following recurrence:

$$\check{L}_k^*(x_k) = \inf_{\gamma_k \in \Gamma_k} \sup_{\theta_k^a \in \Theta_k^a} \{ l_k(x_k, \gamma_k(x_k)) + \check{L}_{k+1}^*(f(x_k, u_k, \theta_k^a)) \}. \quad (38)$$

Recall that $f(x_k, u_k, \theta_k^a)$ represents x_{k+1}, and hence this defines a recursion. We assume that termination is implicitly included as a possible choice.

The goal is to determine the optimal action u_k and termination condition TC_k for every value of x_k and every stage $k \in \{1, \ldots, K\}$. One can begin with stage $K+1$ and repeatedly apply (38) to obtain the optimal actions. At stage $K+1$, we can use the last term of (12) to obtain $\check{L}^*_{K+1}(x_{K+1}) = l_{K+1}(x_{K+1})$. The cost-to-go $\check{L}^*_K$ can be determined from $\check{L}^*_{K+1}$ through (38). Using the $u_K \in U$ and TC_k that minimize (38) at x_K, we define $\gamma^*_K(x_K) = \{u_K, TC_K\}$. We then apply (38) again, using $\check{L}^*_K$ to obtain $\check{L}^*_{K-1}$ and γ^*_{K-1}. These iterations continue until $k = 1$. Finally, we take $\gamma^* = \{\gamma^*_1, \ldots, \gamma^*_K\}$.

The cost-to-go function $\check{L}^*_1$ shares similarities with the concept of a global navigation function in motion planning (Latombe 1991; Rimon and Koditschek 1992), as both represent functions on the configuration space that can be used to control the robot. Also, various forms of dynamic programming have been successfully applied in several other motion planning contexts (Barraquand and Ferbach 1994; Hu, Brady, and Probert 1991; Miura and Shirai 1991; Suh and Shin 1988); for instance, the wavefront expansion method that is described in Latombe (1991) can be viewed as a specific form of dynamic programming.

Recall that because of stationarity, the strategy γ_k does not depend on the stage index k for the problems that we consider.

6.2.1.2 Imperfect Information

We now describe how the dynamic programming equation is applied under nondeterministic sensing and control uncertainties. From a given information state, we wish to evaluate a partial strategy from stage k to stage K. Previously, we used the notation $\check{L}^*_k(x_k)$ to evaluate a part of an optimal strategy from a given state. Using the information state representation $F_k(\eta_k)$, which was defined in Section 3.3, we have

$$\check{L}^*_k(\eta_k) = \sup_{x_k \in F_k(\eta_k)} \check{L}^*_k(x_k). \quad (39)$$

We want to consider the effect of selecting $\gamma_k(\eta_k)$ in the information space η_k. This results in $F_{k+1}(\eta_{k+1})$, as defined in (17). We additionally assume that the per-stage loss does not depend on state $l(x_k, u_k, TC_k) = l(u_k, TC_k)$, which encompasses the loss functionals that we have considered thus far (this assumption is not required in general).

The dynamic programming principle states that $\check{L}^*_k(\eta_k)$ can be obtained from $\check{L}^*_{k+1}(\eta_{k+1})$ by the following recurrence:

$$\check{L}^*_k(\eta_k) = \inf_{\gamma_k \in \Gamma_k} \sup_{\eta_{k+1} \in \bar{F}_{k+1}(\eta_k, u_k)} \left\{ l(\gamma_k(\eta_k)) + \check{L}^*_{k+1}(\eta_{k+1}) \right\}, \quad (40)$$

in which $\check{L}^*_k(\eta_k)$ represents the optimal worst-case loss, obtained by implementing the optimal strategy γ^* from stage k to stage $K+1$.

At stage $K+1$, we can use the last term of (11) to obtain

$$\bar{L}^*_{K+1}(\eta_{K+1}) = \sup_{x_{K+1} \in F_{K+1}(\eta_{K+1})} l_{K+1}(x_{K+1}). \quad (41)$$

6.2.2. *The Probabilistic Case*

6.2.2.1 Perfect Information

Next, we present the principle of optimality under probabilistic control uncertainty. The resulting equation can be applied in the same iterative manner to obtain an optimal solution.

The expected loss obtained by starting from stage k and implementing the portion of the optimal strategy $\{\gamma^*_k, \ldots, \gamma^*_K\}$ can be represented as

$$\bar{L}^*_k(x_k) = E\left\{ \sum_{i=k}^{K} l_i(x_i, \gamma^*_i(x_i)) + l_{K+1}(x_{K+1}) \right\}, \quad (42)$$

in which $E\{\}$ denotes expectation taken over the actions of nature.

The principle of optimality (Kumar and Varaiya 1986) states that $\bar{L}^*_k(x_k)$ can be obtained from $\bar{L}^*_{k+1}(x_{k+1})$ by the following recurrence:

$$\bar{L}^*_k(x_k) = \min_{\gamma_k \in \Gamma_k} \left\{ l_k(x_k, u_k) + \int \bar{L}^*_{k+1}(x_{k+1}) p(x_{k+1} | x_k, u_k) dx_{k+1} \right\}. \quad (43)$$

Note that the integral is taken over states that can be reached using (1).

6.2.2.2 Imperfect Information

We now describe how the dynamic programming equation is applied under probabilistic sensing and control uncertainties. From a given information state, we wish to evaluate a partial strategy from stage k to stage K. Previously, we used the notation $\bar{L}^*_k(x_k)$ to represent the expected loss of executing a partial, optimal strategy from a given state. Using the information state density $p(x_k \mid \eta_k)$ on X, we have

$$\bar{L}^*_k(\eta_k) = \int \bar{L}^*_k(x_k) p(x_k | \eta_k) dx_k. \quad (44)$$

We also consider the one-stage expected loss associated with taking an action, from a given information state, η_k:

$$\bar{l}(\eta_k, \gamma_k(\eta_k)) = \int l(x_k, \gamma_k(\eta_k)) p(x_k | \eta_k) dx_k. \quad (45)$$

This is the expected loss that will be incurred if an action u_k and TC_k are taken from state η_k, resulting in some η_{k+1}. The integral of (45) is determined from (11) and (9). Using the previous notation, the dynamic programming principle states that $\bar{L}^*_k(\eta_k)$ can be obtained from $\bar{L}^*_{k+1}(\eta_{k+1})$ by the following recurrence:

$$\bar{L}^*_k(\eta_k) = \inf_{\gamma_k(\eta_k)} \left\{ \bar{l}(\eta_k, \gamma_k(\eta_k)) + \int \bar{L}^*_{k+1}(\eta_{k+1}) p(\eta_{k+1}|\eta_k, \gamma_k(\eta_k)) d\eta_{k+1} \right\} \quad (43)$$

Above, $p(\eta_{k+1} \mid \eta_k, \gamma_k(\eta_k))$ is determined by replacing g with γ in (29).

At stage $K + 1$, we can use the last term of (11) to obtain

$$\bar{L}^*_{K+1}(\eta_{K+1}) = \int l_{K+1}(x_{K+1}) p(x_{K+1}|\eta_{K+1}) d\eta_{K+1}. \quad (47)$$

6.3. Approximating the State Space

We determine optimal strategies numerically by successively building approximate representations of $\bar{L}^*_k$ over the state space. A uniform grid is used to closely approximate $\bar{L}^*_k$. (Note that the use of a hierarchical representation is more difficult in this context than in traditional path planning.) We obtain the value for $\bar{L}^*_k(x_k)$ by computing the right side of (38) (or the appropriate dynamic programming equation) for various values of u_k and TC_k and using linear interpolation. Other schemes, such as quadratic interpolation, can be used to improve numerical accuracy (Larson and Casti 1982).

Note that the $\bar{L}^*_K$ represents the cost of the optimal one-stage strategy from each state x_K. More generally, $\bar{L}^*_{K-i}$ represents the cost of the optimal $i + 1$ stage strategy from each state x_{K-i}. For a motion planning problem, we are concerned only with strategies that require a finite number of stages before terminating in the goal region, and assume that stationarity holds, as discussed in Section 3.4. We select a positive $\delta \approx 0$ and terminate the dynamic programming iterations when $|\bar{L}^*_k(x_k) - \bar{L}^*_{k+1}(x_{k+1})| < \delta$ for all values in the state space. The resulting strategy is formed from the optimal actions and termination conditions in the final iteration. Note that no choice of K is necessary. Also, at each iteration of the dynamic programming algorithm, we retain only the representation of $\bar{L}^*_{k+1}$ while constructing $\bar{L}^*_k$; earlier representations can be discarded.

To execute a strategy, the robot uses the final cost-to-go representation, which we call $\bar{L}^*_1$. The robot is not confined to move along the quantization grid that is used for determining the cost-to-go functions. The optimal action can be obtained from any real-valued location $x \in X$ through the use of (38) (or the appropriate dynamic programming equation), interpolation, and the approximate representation of $\bar{L}^*_1$. A real-valued initial state is given. The application of the optimal action will yield a new real-valued configuration for the robot. This form of iteration continues until $TC_k = true$.

6.4. Approximating the Information Space

With sensing uncertainty, planning occurs in the information space, which is generally of a much higher dimension than the state space. This section discusses trade-offs between the computational expense and the quality of information space approximations. Many issues exist that may cause one method to be preferable over another. In the current implemented examples, which are presented in Section 6.6, we limit the history to the past sensor observation. This results in sensor feedback, which is similar to the approach used in Erdmann (1993).

These approximation techniques can be used in combination with interpolation, which was discussed in Section 6.3. Strategies are determined, however, by successively building cost-to-go functions on the information space, as opposed to X.

6.4.1. Limiting History

As defined in Section 3.2, η_k is defined as a subset of the sensing and action history. One straightforward way to keep the information space dimension fixed is to limit the amount of history that is remembered. For instance, we can maintain i stages of history to obtain

$$\eta_k = \{u_{k-i+1}, u_{k-i+2}, \ldots, u_{k-1}, y_{k-i}, y_{k-i+1}, \ldots, y_k\}. \quad (48)$$

If $i = 0$, then only the last sensor observation can be retained for decision making, which results in a sensor-feedback strategy. If position sensing is used along with a directional force sensing, then the information space is reduced to having one more dimension than the state space.

6.4.2. Introducing Statistics

A more general way to reduce the information space complexity is to transform the history into a lower dimensional space. This technique encompasses the history limiting approach. An ideal situation exists when an information space can be transformed using a low-dimensional sufficient statistic (Duda and Hart 1973; Kumar and Varaiya 1986). A sufficient statistic implies that any decision that is based on the complete history can equivalently be made by considering only the statistic.

In general, a transformation of the form

$$\eta_k = z_k(u_1, \ldots, u_{k-1}, y_1 \ldots, y_k) \quad (49)$$

is applied to the history. The information space N_k as defined in previous sections can be replaced by the statistic space Z_k for which $z_k \in Z_k$. Strategies are then defined Z_k, and dynamic programming can again be applied to yield solutions.

6.4.3. *Functional Approximation with Moments*

One special type of statistic that can be used to approximate the information spaces is the set of moments. Suppose there is probabilistic uncertainty and the information space is represented as a function space of pdfs. Moments can be used as statistics that summarize the information in the pdfs. Consider, as an example, a second-order approximation. Recall from Section 4.2.2 that from any sensing and action history (i.e., $\{u_1, u_2, \ldots, u_{k-1}, y_1, y_2, \ldots, y_k\}$), the pdf on the state space can be inferred, $p(x_k \mid \eta_k)$. Let μ_k and Σ_k represent the mean vector and covariance matrix, respectively, of $p(x_k \mid \eta_k)$. From any history, we can now obtain the moments μ_k and Σ_k.

If η_k are $\gamma_k(\eta_k)$ given, recall that a density for the next information state $p(\eta_{k+1} \mid \eta_k, \gamma_k(\eta_k))$ will be obtained. This was used in (46) as part of the principle of optimality, which can be used to compute an optimal strategy. When using moments, we can replace $p(\eta_{k+1} \mid \eta_k, \gamma_k(\eta_k))$ with $p(\mu_{k+1}, \Sigma_{k+1} \mid \mu_k, \Sigma_k, \gamma_k(\mu_k, \Sigma_k))$ in (46). This can be used to determine optimal strategies from moments (optimal on the approximated information space).

An information feedback strategy $\gamma_k(\eta_k)$ is then replaced by a moment feedback strategy $\gamma_k(\mu_k, \Sigma_k)$. The hope in using moment approximation is that $\gamma_k^*(\eta_k) \approx \gamma_k^*(\mu_k, \Sigma_k)$ for all $\eta_k \in N_k$.

6.5. *Computational Performance*

We briefly discuss the computational performance of the dynamic programming computations. Let Q denote the number of cells per dimension in the representation of $\mathcal{C}_{\text{free}}$. Let n denote the dimension of the information space (which becomes the dimension of the state space in the case of perfect information). Let $\mid U \mid$ denote the number of actions that are considered. Let $\mid \Theta \mid$ denote the number of actions that are considered by nature. The space complexity of the algorithm is $O(Q^n)$, which is proportional to the size of the state space. For each iteration of the dynamic programming, the time complexity is $O(Q^n \mid U \mid \mid \Theta \mid)$, and the number of iterations is proportional to the robot velocity and the complexity of the solution strategy. The number of iterations required is directly proportional to the number of stages required for the longest (in terms of stages) optimal strategy that reaches the goal. The computation at each cell (in the application of (38)) has time complexity $O(\mid U \mid \mid \Theta \mid)$, with n fixed. This analysis assumes constant time for interpolation; typically, however, the interpolation complexity itself is exponential in dimension.

The computational cost of dynamic programming increases exponentially in the dimension of the state space for perfect information and the information space for imperfect sensing; however, most algorithms that solve the basic motion planning problem without sensing and control uncertainty have exponential complexity in the dimension of the configuration space (for surveys and comparisons, see Hwang and Ahuja 1992; Latombe, 1991). We consider the current approach to be reasonable for a few dimensions, which includes many interesting motion planning problems. For more difficult problems, some additional computational techniques may need to be developed.

In our simulation experiments, we have considered problems in which $X \subseteq \Re^2$. We typically divide the state space into 50×50 cells, 64 quantized actions to approximate translational motion. We have considered similar quantizations of the information space under sensor feedback.

The computation times vary dramatically depending on the resolutions of the representation. For the examples that we present in this paper, the computation times vary from about a few minutes to a few hours on a SPARC 10 workstation. It is important to note that the dynamic programming equations are highly parallelizable. For example, under probabilistic uncertainty with perfect state information, the computation of the optimal action at each location x_k depends only on a very local portion of the representation of $\bar{L}_{k+1}^*(x_{k+1})$ and on no portion of $\bar{L}_k^*(x_k)$. A parallelized implementation of the algorithm would significantly improve performance.

6.6. *Computed Examples*

In this section, we present computed examples of optimal strategies that were determined by the computational methods discussed in Section 6. For these strategies, we show forward projections and preimages that can be compared to the results in Sections 4.3 and 5.3.

For the results in this section, we used the loss functional (13) with $l(u_k, TC_k) = \| v \| \Delta t$ and $C_f = 10,000$. We have found the loss functional (12) to not be as useful for determining optimal strategies. For most problems, the cost-to-go is zero at every state from which it is possible to achieve the goal. Therefore, there are many strategies that are considered equivalent, while in reality the expected time (or worst-case time) for some of the strategies to achieve the goal may be arbitrarily longer. For fixed motion commands, however, (12) provided useful information because a strategy was chosen that in many possible trajectories did not achieve the goal.

For the first example, we refer back to the peg-in-hole problem that was introduced in Section 4.3. We assume, as considered previously, that $\| v \| \Delta t = 3$ and $\epsilon_\theta = 48.8°$. Figures 7a and 7b show computed results that were obtained under probabilistic uncertainty with perfect state information. Figure 7a depicts the optimal strategy by showing the direction of the motion command $u_k = \gamma_k^*(x_k)$ at different locations in the state space. Figure 7b shows isoperformance classes for every six units (i.e., there is a contour for every two expected stages of motion). This can be compared to the preimage results from Section 5.3; under the implementation

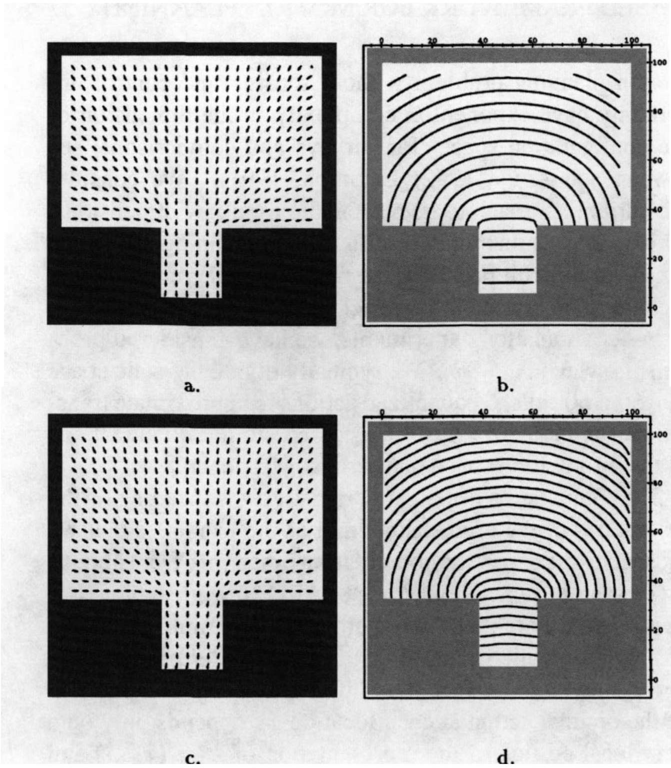


Fig. 7. Optimal strategies and performance preimages for the peg-in-hole problem under probabilistic control uncertainty and nondeterministic control uncertainty.

of the optimal strategy, the curves emanate radially from the goal region.

Figures 7c and 7d show computed results that were obtained under nondeterministic uncertainty with perfect state information. The isoperformance curves are closer together because worst-case analysis causes the computed loss to be greater.

Figure 8 shows several more computed optimal strategies for probabilistic uncertainty with perfect state information. We assume for each of these examples that $\| v \| \Delta t = 3$ and $\epsilon_\theta = 48.8°$.

Figure 9 shows the computed optimal strategies for probabilistic uncertainty with imperfect state information. We assume for each of these examples that $\| v \| \Delta t = 3$ and $\epsilon_\theta = 48.8°$. We used the sensing model from Section 3.5 and let $\epsilon_p = 5$ and $\epsilon_f = 0$. Without perfect sensing, the expected time to reach the goal increases, which causes the isoperformance curves to be closer together. In addition, the sample paths under the implementation of the optimal strategies involve more variations.

The strategy representation and the isoperformance curves in Figure 9 do not align completely with the obstacles in the workspace because the optimal actions and isoperformance curves are defined in the information space. For these examples, the information space is represented by the set of possible sensor values. Sensed force values are not shown in the figures.

Figure 10 shows computed optimal strategies for nondeterministic uncertainty with imperfect state information. A solution strategy could not be found using the same uncertainty models as for the probabilistic case. This occurs because worst-case analysis eliminates the consideration of many reasonable strategies, as mentioned in Section 2. We therefore use $\| v \| \Delta t = 10$, $\epsilon_\theta = 0.8$, $\epsilon_p = 2.5$, and $\epsilon_f = 0$. The isoperformance curves are shown for every 30 units of loss.

7. Discussion

In this section, we briefly discuss some aspects of the current framework and future directions that could be taken with this research.

7.1. Randomized Strategies

In Erdmann (1989, 1992, 1993), useful manipulation planning methods were developed around randomizing the actions of the robot. It is important to note that in our work the robot strategy is deterministic even though the execution of the strategy can be considered as a random process. Erdmann has argued that two important benefits result from using randomized strategies: (1) robustness with respect to incorrect models can be obtained and (2) multiple attempts can be made to solve a task instead of requiring a guaranteed solution.

By conditioning our strategies on state feedback or information feedback, the robot is capable of making multiple attempts to solve a task. In a motion planning context, one can imagine a robot that attempts to execute a motion plan, reports failure, and then replans to make another attempt; this behavior is exhibited in the error detection and recovery strategies in Donald (1987, 1988, 1990). An "attempt" is not as distinct in our approach, however, because the robot responds dynamically to its information. Rather than recomputing a new strategy, the response corresponds to the optimal behavior that was determined through global analysis of the motion planning problem and its uncertainties.

Robustness with respect to incorrect modeling represents a useful feature, which has not been considered by our framework thus far. In the approach that we present, the assumption is made that the models are correct. Under nondeterministic uncertainty, the correctness of the model can become critical since it then becomes impossible to "guarantee" a particular loss (unless the model truly represents an upper bound on the uncertainty). Under probabilistic uncertainty, the effect of modeling errors appears to be less drastic. One difficulty with introducing randomization is that it can arbitrarily increase the loss required to complete the goal even though robustness is strengthened. In the limiting case, pure Brownian

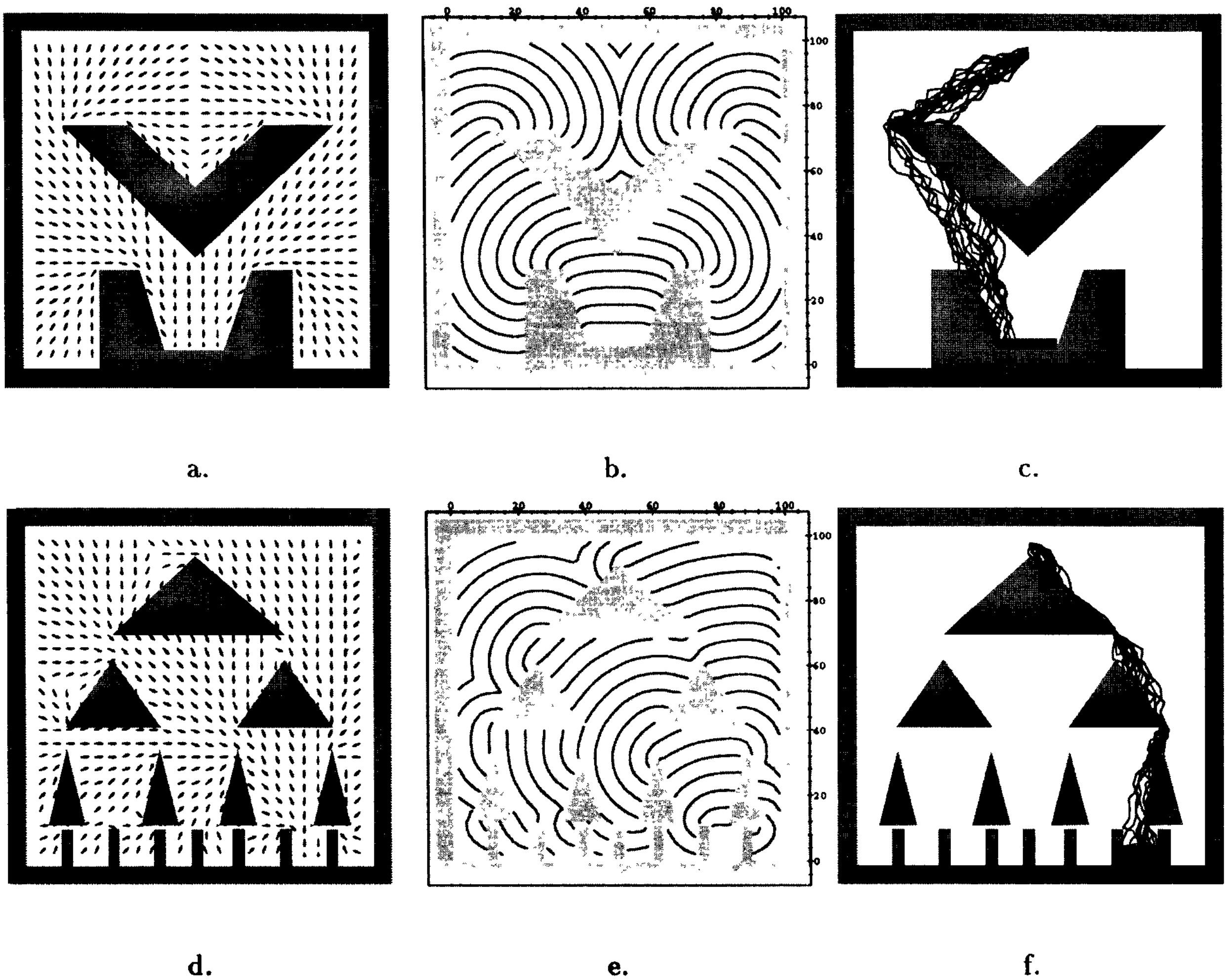


Fig. 8. Examples that were computed under probabilistic uncertainty and perfect state information.

motion can be executed. This essentially makes no modeling assumptions and will achieve the goal, but the loss can be extremely high. It remains to be seen whether randomized actions can be incorporated into our framework to provide a reasonable trade-off between the distance of a strategy from optimality and the potential incorrectness of the models.

7.2. *Hierarchical Strategies*

In the methods developed in this paper, the robot executes a fixed command at each Δt. In traditional preimage planning, however, a fixed action is executed until the termination condition is met. If the goal is not yet reached, another action is executed. In general, a sequence of fixed actions with termination conditions is executed until the goal is reached.

Recall that the performance preimage can be used to evaluate a particular strategy. One interesting approach would be to implement preimage backchaining and subgoals by performance preimages. We can define G_1 as a subgoal for a larger problem, and define a g and TC that achieves G_1 in a satisfactory way. The resulting posterior density $p(x_{K+1})$ would be used as the initial information state for the achievement of a second goal G_2. We can consider abstract actions of the form $\{G_i, \gamma\}$ that attempt to achieve some original goal. Backchaining from G under explicit performance measures and a given set of choices for abstract actions is another form of dynamic programming. The relationship between standard preimage planning and dynamic programming is discussed in Erdmann (1993). The reason for considering abstract actions and subgoals is the hope that a simple set of abstract actions exists that can be composed to provide quick and efficient solutions for a wide class of problems (as was the case with backprojection planning (Erdmann 1984).

7.3. *Determining Accurate Uncertainty Models*

The flexibility of our approach permits the use of a variety models for sensing and control uncertainty. In many previ-

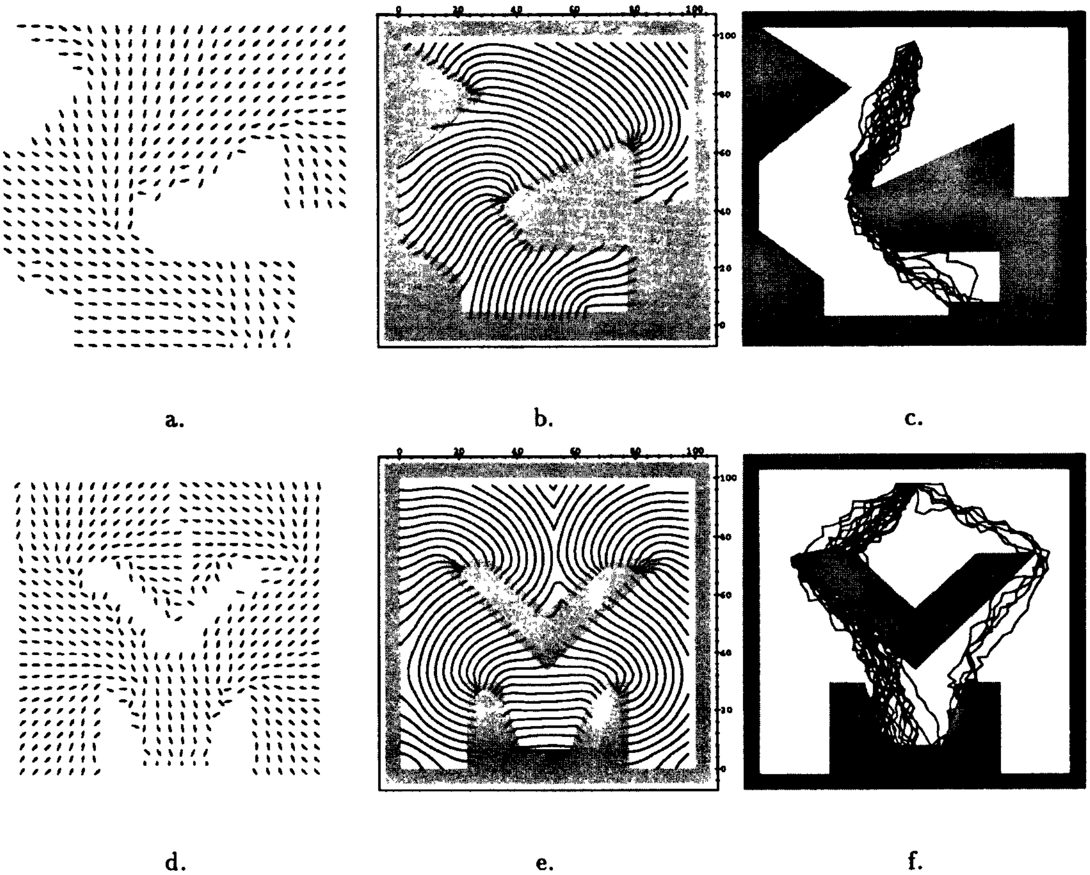


Fig. 9. Examples that were computed under probabilistic uncertainty and imperfect state information.

ous approaches, the results were strongly dependent on the particular model chosen. For instance, worst-case analysis in the backchaining approach has often used bounded disk uncertainty for position and bounded angular error for control uncertainty. With our approach, one important area of future research is to develop models that accurately reflect the uncertainty involved in a particular manipulation task. This is particularly true for the case of probabilistic uncertainty. The densities hold a large amount of expressive power; however, simple models are often chosen to obtain reasonable results. Within our framework, different uncertainty models can be substituted and, through simulations or repeated execution trials, better uncertainty models could be developed for a particular context. This direction of research was also advocated in Brost and Christiansen (1996) to determine valid error distributions for the computations of appropriate probabilistic backprojections.

7.4. Sampling Issues

One important issue that has received little attention in manipulation planning literature is the sampling rates that are available for sensing and control. In the typical preimage planning formulation, the robot is allowed to issue a new command at any point in time, implying continuous-time controllability of the robot. The robot command is changed, however, only during the few occurrences of meeting the termination condition. In this paper, we have assumed a sampling rate that essentially approximates continuous-time control and sensing. By allowing the motion command to change at any discrete stage, we obtain a significant amount of control over the robot in the face of uncertainty. One useful approach might be to consider a much lower sampling rate. This models the situation in which fine motions are performed before additional sensing or a new control input can be applied. This seems to appropriately reflect a situation in which the planning workspace is very small, such as in a part-mating operation.

8. Conclusion

We presented a flexible framework for manipulation planning under uncertainty in which motion strategies are selected to optimize a loss functional. We have indicated through the discussion and simulation experiments that the efficiency of a robot motion strategy is crucial in planning under uncertainty.

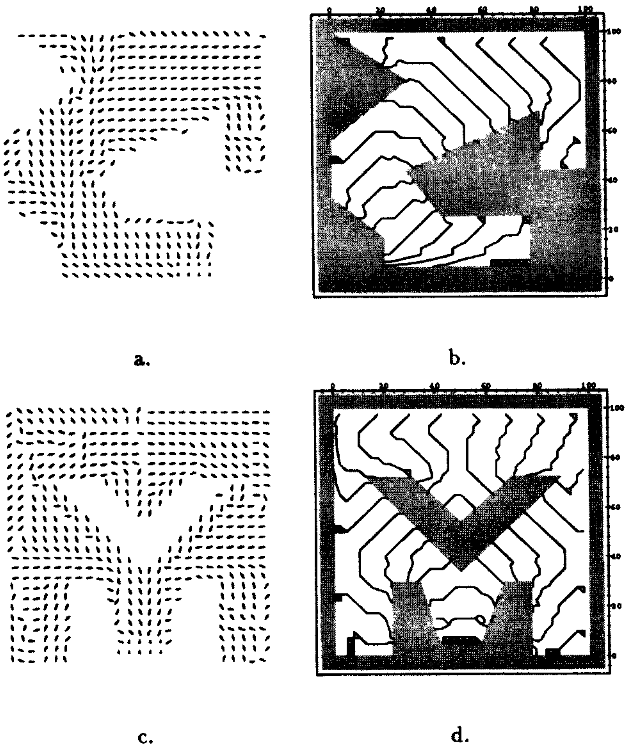


Fig. 10. Examples that were computed under nondeterministic uncertainty and imperfect state information.

We have developed a performance preimage as a useful concept for evaluating motion strategies, which generalizes the classical preimage. This work identifies termination criteria with optimal stopping problems from optimal control theory and allows the incorporation of a termination condition into the optimal strategy. We apply information space concepts from stochastic control and dynamic game theory to incorporate history into a motion strategy with uncertainty in sensing. We additionally provide a computational approach that numerically determines optimal motion strategies under a wide class of performance functionals by applying the dynamic programming principle to approximate stationary cost-to-go functions and illustrate the concepts through computed examples. One of the most important directions for future research will be to investigate different methods of approximately representing the information space.

Acknowledgments

We gratefully acknowledge Narendra Ahuja, Tamer Basar, Randy Brost, Becky Castaño, Mike Erdmann, Peter Leven, Kevin Nickels, Jean Ponce, Rajeev Sharma, Steve Sullivan, and Mark Spong for their helpful comments and suggestions. This work was sponsored by NSF under Grant No. IRI-9110270, a Beckman Institute research assistantship, and a Mavis Fellowship.

References

Alami, R., Siméon, T., and Laumond, J. P. 1989. A geometrical approach to planning manipulation tasks. *5th Int. Symp. on Robotics Research*, pp. 113–119.

Anderson, B. D., and Moore, J. B. 1990. *Optimal Control: Linear-Quadratic Methods*. Englewood Cliffs, NJ: Prentice Hall.

Başar, T., and Kumar, P. R. 1987. On worst case design strategies. *Comput. Math. Appl.* 13(1-3):239–245.

Başar, T., and Olsder, G. J. 1982. *Dynamic Noncooperative Game Theory*. London: Academic Press.

Barraquand, J., and Ferbach, P. 1994. A penalty function method for constrained motion planning. *IEEE Int. Conf. on Robotics and Automation*, pp. 1235–1242.

Bertsekas, D. P. 1987. *Dynamic Programming: Deterministic and Stochastic Models*. Englewood Cliffs, NJ: Prentice Hall.

Brost, R. C. 1991. Analysis and planning of planar manipulation tasks. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.

Brost, R. C., and Christiansen, A. D. 1993. Probabilistic analysis of manipulation tasks: A research agenda. *IEEE Int. Conf. on Robotics and Automation*, Vol. 3, pp. 549–556.

Brost, R. C., and Christiansen, A. D. 1996. Probabilistic analysis of manipulation tasks: A computational framework. *Int. J. Robotics Research* 15(1):1–23.

Bryson, A. E., and Ho, Y.-C. 1975. *Applied Optimal Control*. New York: Hemisphere.

Canny, J. F. 1988. *The Complexity of Robot Motion Planning*. Cambridge, MA: MIT Press.

Canny, J. F. 1989. On computability of fine motion plans. *IEEE Int. Conf. on Robotics and Automation*, pp. 177–182.

Dean, T. L., and Wellman, M. P. 1991. *Planning and Control*. San Mateo, CA: Morgan Kaufman.

Devijver, P. A., and Kittler, J. 1982. *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, NJ: Prentice Hall.

Donald, B. R. 1987. Error detection and recovery for robot motion planning with uncertainty. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Donald, B. R. 1988. A geometric approach to error detection and recovery for robot motion planning with uncertainty. *Artificial Intell.* 37:223–271.

Donald, B. R. 1990. Planning multi-step error detection and recovery strategies. *Int. J. Robotics Research* 9(1):3–60.

Donald, B. R., and Jennings, J. 1991 (Sacramento, CA). Sensor interpretation and task-directed planning using perceptual equivalence classes. *IEEE Int. Conf. on Robotics and Automation*, pp. 190–197.

Duda, R. O., and Hart, P. E. 1973. *Pattern Classification and Scene Analysis*. New York: Wiley.

Erdmann, M. A. 1984. On motion planning with uncertainty. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

Erdmann, M. A. 1989. On probabilistic strategies for robot tasks. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Erdmann, M. 1992. Randomization in robot tasks. *Int. J. Robotics Research* 11(5):399–436.

Erdmann, M. 1993. Randomization for robot tasks: Using dynamic programming in the space of knowledge states. *Algorithmica* 10:248–291.

Feddema, J. T., and Mitchell, O. R. 1989. Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation* 5(5):5691–5700.

Fox, A., and Hutchinson, S. 1995. Exploiting visual constraints in the synthesis of uncertainty-tolerant motion plans. *IEEE Trans. on Robotics and Automation* 1(11):56–71.

Goldberg, K. Y. 1990. Stochastic plans for robotic manipulation. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.

Goldberg, K. Y. 1993. Orienting polygonal parts without sensors. *Algorithmica* 10:201–225.

Hu, H., Brady, M., and Probert, P. 1991 (Osaka, Japan). Coping with uncertainty in control and planning for a mobile robot. *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pp. 1025–1030.

Hwang, Y. K., and Ahuja, N. 1992. Gross motion planning—A survey. *ACM Computing Surveys* 24(3):219–291.

Kirman, J., Basye, K., and Dean, T. 1991. Sensor abstraction for control of navigation. *IEEE Int. Conf. on Robotics and Automation*, pp. 2812–2817.

Kumar, P. R., and Varaiya, P. 1986. *Stochastic Systems*. Englewood Cliffs, NJ: Prentice Hall.

Larson, R. E. 1968. *State Increment Dynamic Programming*. New York: Elsevier.

Larson, R. E., and Casti, J. L. 1982. *Principles of Dynamic Programming, Part II*. New York: Dekker.

Latombe, J.-C. 1991. *Robot Motion Planning*. Boston: Kluwer Academic.

Latombe, J.-C., Lazanas, A., and Shekhar, S. 1991. Robot motion planning with uncertainty in control and sensing. *Artificial Intell.* 52:1–47.

LaValle, S. M. 1995. A game-theoretic framework for robot motion planning. Ph.D. thesis, University of Illinois, Urbana.

LaValle, S. M. 1997. Robot motion planning: A game-theoretic foundation. *Algorithms for Robotic Motion and Manipulation*, eds. J. -P. Laumond and M. Overmars. Boston: A. K. Peters.

LaValle, S. M., and Hutchinson, S. A. 1994. An objective-based stochastic framework for manipulation planning. *Proc. IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems*, pp. 1772–1779.

Lozano-Pérez, T., Mason, M. T., and Taylor, R. H. 1984. Automatic synthesis of fine-motion strategies for robots. *Int. J. Robotics Research* 3(1):3–24.

Mahadevamurty, N., Tsao, T.-C., and Hutchinson, S. 1994. Multi-rate analysis and design of visual feedback digital servo control systems. *Trans. ASME J. Dyn. Sys. Meas. Control*, pp. 45–55.

Mason, M. T. 1982. Compliance and force control for computer controlled manipulators. *Robot Motion: Planning and Control*. MIT Press, Cambridge, MA, pp. 373–404.

Miura, J., and Shirai, Y. 1991 (Osaka, Japan). Planning of vision and motion for a mobile robot using a probabilistic model of uncertainty. *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pp. 403–408.

Natarajan, B. K. 1988. The complexity of fine motion planning. *Int. J. Robotics Research* 7(2):36–42.

Papadimitriou, C. H. 1985. Games against nature. *J. Computer and System Sciences* 31:288–301.

Papadimitriou, C. H., and Tsitsiklis, J. N. 1987. The complexity of markov decision processes. *Math. Oper. Research* 12(3):441–450.

Paul, R. P., and Shimano, B. 1976. Compliance and control. *Proc. of the Joint American Automatic Control Conf.*, pp. 1694–1699.

Raibert, M. H., and Craig, J. J. 1981. Hybrid position/force control of manipulators. *Trans. ASME J. Dyn. Sys. Meas. Control*, p. 102.

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

Shan, Y., and Koren, Y. 1995. Obstacle accommodation motion planning. *IEEE Trans. on Robotics and Automation* 1(11):36–49.

Suh, S.-H., and Shin, K. G. 1988. A variational dynamic programming approach to robot-path planning with a distance-safety criterion. *IEEE Trans. on Robotics and Automation* 4(3):334–349.

Takeda, H., Facchinetti, C., and Latombe, J.-C. 1994. Planning the motions of a mobile robot in a sensory uncertainty field. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 16(10):1002–1017.

Taylor, R. H., Mason, M. T., and Goldberg, K. Y. 1987. Sensor-based manipulation planning as a game with nature. *Fourth Int. Symp. on Robotics Research*, pp. 421–429.

Whitney, D. 1977. Force feedback control of manipulator fine motions. *Trans. ASME J. Dyn. Sys. Meas. Control* 99:91–97.

Whitney, D. E. 1985. Historical perspectives and the state of the art in robot force control. *IEEE Int. Conf. on Robotics and Automation*, pp. 262–268.