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

Steven M. LaValle
lavalle@cs.uiuc.edu

Seth A. Hutchinson
seth@uiuc.edu

Dept. of Electrical and Computer Engineering
and
The Beckman Institute
University of Illinois, Urbana, IL 61801

Abstract

We consider the problem of determining robot motion plans under sensing and control uncertainties. Traditional approaches are often based on a methodology known as *preimage planning*, which involves worst-case analysis. We 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 to *motion strategies with information feedback*. For a given problem, one can define a performance criterion that evaluates any executed trajectory of the robot. We 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). We 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.

DRAFT COPY

Submitted to: *The International Journal of Robotics Research, 1995*

Contents

1 Introduction

As we continue to demand that robots perform more complicated tasks, the need for general and flexible approaches to motion planning increases. It has been widely acknowledged in the literature that modeling and overcoming uncertainty is crucial to successful motion planning in practical environments. Sources of such uncertainty include geometric model inaccuracies, limited or noisy sensing, and only partially predictable execution of motion commands.

We present an objective-based framework that addresses sensing and control uncertainties. An earlier version of this approach appeared in [49]. By objective-based we mean that motion strategies are chosen by minimizing a criterion that evaluates a trajectory, by taking into account quantities such as the path length or the probability of success. Many of the concepts introduced here are based on ideas used in stochastic optimal control theory and dynamic game theory (see [4, 44] for background). Our approach generalizes classical preimages to *performance preimages* and preimage planning to *motion strategies with information feedback*. Constructing a traditional *termination criterion* becomes equivalent to formulating an *optimal stopping problem* [8] within our framework.

Several variations of the motion planning problem have been considered in past research. 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 [39]. 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 in, e.g., [17, 30, 53], and the *manipulation planning* problem in, e.g., [13]. In other research, *manipulation planning* has been used to refer to problems that involve the transfer of objects in the workspace (e.g., [1, 47]).

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 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 basic collision-free planning issues under sensing and control uncertainties are included in the manipulation planning model. Section 2 provides an overview of manipulation

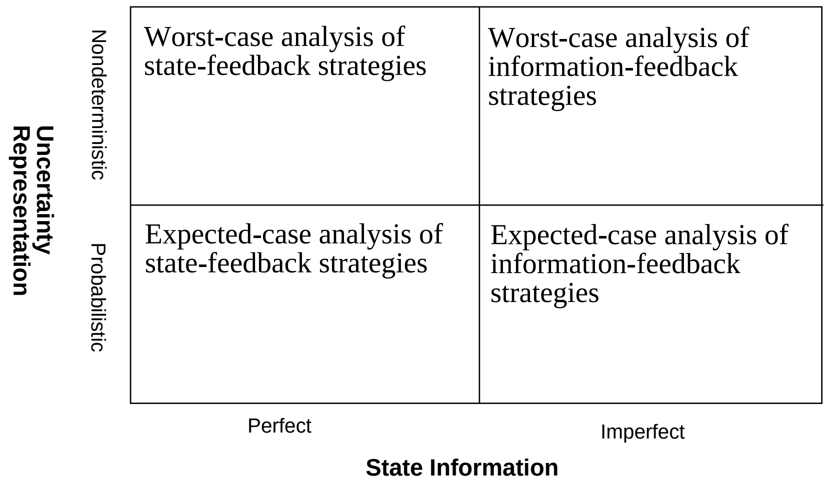


Figure 1: Four types of uncertainty that are considered in this paper.

planning research and discusses the significance of our approach.

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 control theory (as used in [4, 44]). After the general concepts and definitions have been presented, they are utilized 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 4-6 can be logically divided into four components that represent different types of uncertainty (see Figure 1). The precise meaning of these types of uncertainty should become clearer after reading Sections 2 and 3. These types are based on two choices: 1) using probabilistic representations vs. nondeterministic (or bounded-set) representations; 2) modeling control errors and assuming perfect sensing vs. modeling both control and sensing errors. Both probabilistic and nondeterministic representations offer 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. By assuming perfect sensing, many effects of control uncertainty on manipulation planning can be observed. These 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. The motion planning analysis for this case is performed on this information space, as opposed to the original configuration space.

Section 7 discusses several issues regarding our current approach, and how it can be extended. Section 8 summarizes our contributions.

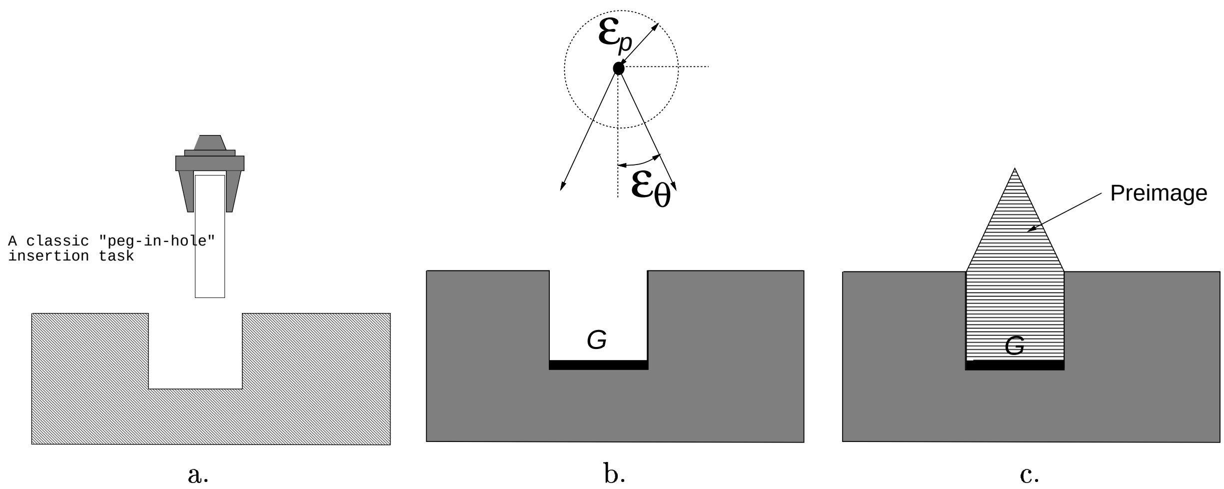


Figure 2: a) A classic 2D "peg-in-hole" insertion task without rotation; b) Such a task can be represented in configuration space with bounded uncertainty in commanded velocity and sensed configuration; c) The classical preimage.

2 Background and Motivation

Section 2.1 provides an overview of previous manipulation planning research. Section 2.2 discusses the key contributions of our research in terms of previous research.

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 by Lozano-Perez, et al. [53]. Using geometric reasoning techniques, a robot 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 2 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.

We will next discuss several important aspects of manipulation planning that have been elucidated in preimage planning research, and conclude Section 2.1 by summarizing some additional related motion planning research.

Motion models One of the most important manipulation planning concerns is the motion of the robot when it is in contact with obstacles. *Force compliant control* is usually permitted, which enables the robot to move along obstacle boundaries [56]. A variety of motion models have been considered for compliant control (e.g., [64, 65, 67, 79]), and the *generalized damping model* has been most often considered in this context [47, 78]. More complex interactions have also been analyzed, such as the motion of objects as they are pushed by a manipulator [12, 29, 54, 58, 77]. Figure 3 depicts the type of motion that occurs when the generalized damping model is combined with friction. If the commanded velocity is outside of the friction cone, but is pointing into the obstacle boundary, then the robot moves along the boundary. If the commanded velocity is inside the friction cone, then the robot remains motionless. When the robot is not in contact the set of available motions could be constrained, as for nonholonomic robots as discussed in [7], or the robot might be free to move along any continuous path. For manipulation planning problems it is thus important to allow different motions to be defined on different subsets of the configuration space.

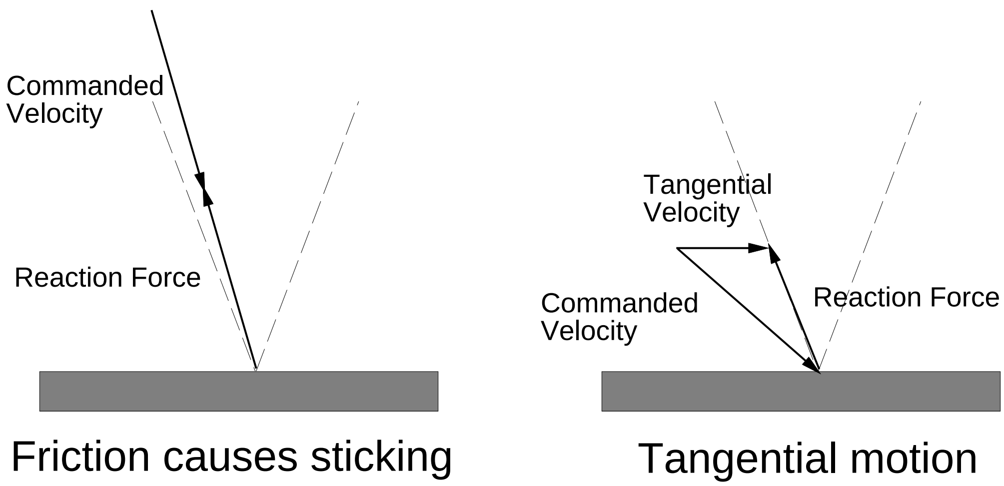


Figure 3: Motions under the generalized damping model with friction.

Modeling uncertainty Two basic representations of sensing and control uncertainty have been proposed in the manipulation planning literature. We refer to these as *nondeterministic uncertainty* and *probabilistic uncertainty*, as done in [27]. 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., [30, 47, 48, 53]). 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 [12, 13, 14]. 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 [13, 27], 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.

History Operation under sensing uncertainty can be improved by incorporating sensing and/or control history into the robot's decision making. This is similar to the use of information feedback in control theory and dynamic game theory. There is naturally a close correspondence between a feedback controller and a motion plan that incorporates sensing information (in our framework these concepts are equivalent).

Several approaches to manipulation planning have incorporated sensor information. Erdmann presented an approach that yields motion strategies that are conditioned on knowledge states [28]. These knowledge states are inferred from the sensing history, and at a high level correspond closely to using information feedback. In [48] Latombe et al. used sensing and control history 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. Goldberg [35] assumed that no sensor information was available, and conditioned squeezing operations on the history of previous operations (which is equivalent to the control history). Donald and Jennings have defined *perceptual equivalence classes* which determine distinguishable scenarios for a mobile robot based on its sensing history and the projection from the configuration space onto the sensor space [24].

Termination condition 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 does not realize that it is in the goal region (*recognizability*) [30]. Under nondeterministic uncertainty it is said that a plan *achieves* a goal if the robot is guaranteed to halt in the goal region.

A logical predicate called the *termination condition* is typically defined; this is a function of the sensing history that is available to the robot. A general form for the termination condition is [48]:

$$TC(t, \mathbf{q}^*[0,t], \mathbf{f}^*[0,t]) \rightarrow \{true, false\}, \tag{1}$$

in which $\mathbf{q}^*[0,t]$ and $\mathbf{f}^*[0,t]$ represent the full history of position and force sensing up to time t. A specialized version of (1) is usually implemented for practical considerations.

Computing manipulation plans Mason has shown that preimage backchaining is bounded complete (i.e., if a solution with a bounded number of motions exists, the preimage backchaining method will find it), and that it suffices to consider directional preimages (i.e, preimages computed with respect to a single, fixed motion direction) as subgoals in the recursive backchaining process [57]. These results, however, do not imply that preimages are computable. In fact, Erdmann has proven by a reduction from the halting problem, that, in arbitrary environments, preimages and backprojections are uncomputable [30]. That the recursively defined constraints in the proof do not generally occur in practice led Erdmann to conjecture without proof that in an environment with a known finite number of constraints, preimages should be computable. Canny has shown that this is indeed the case when the set of possible robot trajectories has a finite parameterization, and the set of feasible trajectories is a semi-algebraic subset of the parameter space [17]. Canny's approach is to cast the manipulation planning problem as a decision problem in the theory of the real numbers, and to then use quantifier elimination algorithms (see, e.g., [18]) to derive parametric semi-algebraic sets that are preimages.

Erdmann realized that conditions for recognizability and reachability could be decoupled [30]. This led to the development of the *backprojection*, which can be can be considered as an approximation to the preimage that does not directly incorporate the termination condition. Backprojections from recognizable goal regions constitute valid preimages (although not necessarily maximal), and are considered in [30, 48]. A directional backprojection is a backprojection computed for a single, fixed motion direction. The *nondirectional backprojection* [10, 22] is a subset of $\mathcal{C} \times S^1$ in which there is one directional backprojection for each orientation in S^1 (we use C to denote the configuration space). Donald has shown that the topology of the directional backprojection changes only at a finite set of critical velocity orientations in S^1 [21]. Therefore, the nondirectional backprojection can be represented by a finite set of directional backprojections; one for each critical orientation, and one for each non-critical interval.

In general, the manipulation planning problem under sensing and control uncertainty has been shown to have high complexity. Consider the problem of finding a sequence of motions that is guaranteed to move every point in a polyhedral initial region to a polyhedral goal region, amidst polyhedral obstacles, while permitting compliance. This problem, considered in traditional preimage planning, has been shown by Natarajan to be PSPACE-hard in $\Re^2$ [60]. Canny showed that this problem is nondeterministic exponential-time hard in $\Re^3$ [16].

Other related approaches Several other planning problems and approaches are related to the context developed in this paper. In a series of papers by Donald [21, 22, 23], it 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 [35, 36]. Fox and Hutchinson developed methods for computing backprojections that result from using visual constraint rays that result from the correspondence between edges in the workspace and the image plane [33]. Algorithms for computing strategies in the presence of probabilistic uncertainty for mobile robots have been developed in [19, 38, 42]. In [75] a *Sensory Uncertainty Field* was introduced that indicates positional sensing accuracy for a mobile robot as a function of configuration. The field is then used to determine a continuous path from the initial configuration to the goal configuration that minimizes the amount of expected sensing error or some combination of sensing error and path length. In general, other forms of uncertainty have been introduced into the motion planning paradigm, such as uncertainty in the environment model [22, 26, 37, 61, 72, 73], or uncertainty in predicting a changing environment [52, 68]. Object grasping has also been carefully studied in the context of manipulation planning under uncertainty (e.g., [11]).

2.2 Motivation

Consider two basic questions that we can ask about the performance of a particular robot strategy: 1) How likely is the goal to be achieved; 2) If the goal is achieved, how efficient is the solution? In manipulation planning work, usually only the first question is precisely addressed, although there is often some weak preference for more efficient plans (e.g., fewer backchaining steps). In traditional preimage backchaining work, people have been interested in generating strategies that answer the first question by guaranteeing that the goal will be achieved. One notable exception is the consideration of sensorless backchaining in terms of geometric optimality on a Markov chain that describes grasped part orientations [35]. Recently, Brost and Christiansen have considered the probability that a goal will be achieved using a particular robot command, leading to the development of a probabilistic backprojection [13, 14].

When a probabilistic or stochastic formulation is considered, both of the previous questions should be carefully considered. Many stochastic models will lead to guaranteed goal achievement for any possible set of bounded-velocity motion commands, even though the probability that the goal will be achieved in some reasonable finite time interval is very small. For example, continuous Brownian motion will eventually lead the robot to any nonzero-measure goal region, which under perfect sensing indicates that Brownian motion achieves *probabilistic completeness* (a term used in [6]). This fact forms the basis for incorporating specific diffusion processes into robotic plans in [27]. For our context, however, the expected time to actually achieve the goal can be arbitrarily high. This indicates that for at least some problems, the probability that the goal will eventually be achieved is of little value for selecting a planning strategy.

In general, the same type of difficulty can be imagined under nondeterministic uncertainty. There might exist a large set robot strategies that are guaranteed to achieve the goal, but many solutions could be impractical if the execution time is too high. Therefore the efficiency of the solution (e.g., the amount of time the robot is expected to take to achieve the goal) is of great importance in evaluating a robot strategy under general models of uncertainty. This form of efficiency has also been useful in related robot control contexts (e.g., [9, 41, 70]). In this paper we show how objectives can be precisely defined that answer both of the questions above and guide the selection of a robot strategy.

The classical preimage has been a useful conceptual tool for developing manipulation planning algorithms. In its most common use, however, the preimage is defined in a limited manner. Usually a preimage requires that: 1) the robot executes a fixed motion command, and 2) elements in the preimage indicate from where the robot is *guaranteed* to achieve the goal. In our work, we replace fixed motion commands with a state-feedback or information-feedback controller for which trajectories can be evaluated with a precise objective. We also consider models that do not limit preimages to guaranteeing that the goal is achieved. A notable departure from this model was in the definition of probabilistic backprojections in [13]. One motivation behind probabilistic backprojections, as well as our framework, is that worst-case analysis tends to eliminate the consideration of many reasonable robot strategies. The absolute requirement that the goal is recognizably achieved can be too strong, particularly as the amount of uncertainty in control and sensing is increased. Furthermore, if bounded uncertainty models are replaced by smooth probability density functions, then it becomes impossible to *guarantee*[1] that the goal will be achieved in a fixed amount of time, except in restricted cases. For this reason, we consider the formulations under probabilistic uncertainty to represent important tools for analyzing manipulation planning problems.

The relationship between sensor and action history and decision making has long been considered important in planning under uncertainty (e.g. [30, 47, 53]). For our context we want to optimize the performance of the robot, while directly taking into account the complications due to limited sensing. By using the concept of information state, as considered in stochastic control and dynamic game theory, we provide a useful characterization of this relationship. When there is perfect sensing information, the robot conditions its actions on its current state (or configuration); however, with imperfect sensing information the robot actions are defined in terms of information states. We will define a robot strategy to be a mapping (or sequence of mappings) on an information space, which *a priori* takes into account the various contingencies presented during execution. For this reason, no form of dynamic replanning is required. The information state concept is similar to the definition of knowledge states, considered in [28].

The general structure of our framework allows error models to be changed without altering the general computational approach. This feature can facilitate the improvement of error models that are appropriate for a particular robotic system. For instance, if a better sensing model (in

[1] "Guarantee" in a stochastic setting should technically be replaced by "achieve with probability one."

which the error is described in terms of a probability density function) is determined for a given application, the appropriate density can be replaced, and much of the general approach remains the same. The formulation of objectives can also easily be changed. By easily changing these components, better models can be iteratively developed. This framework can also be adapted to a more general multi-player, dynamic game theory, in which the interaction of potentially conflicting objectives can be analyzed [4, 49], or to include model and environment uncertainties [52].

We next discuss the nature of the solutions that we obtain. Recall from Section 2.1 that the computational complexity of a basic manipulation planning problem is PSPACE-complete for $\Re^2$ [60] and nondeterministic exponential-time hard for $\Re^3$ [16, 17]. Solutions for various stochastic, Markov systems are also known to have difficult complexity [62, 63]. In optimal control theory, which shares many commonalities with our approach, closed form solutions can sometimes be found, which allows efficient computation of the solution for a particular system. For instance, one of the most celebrated solutions in stochastic optimal control theory is for the *Linear Quadratic Gaussian* (LQG) regulator problem. For this case the system (or motion model in our terms) is linear, the performance criterion is quadratic, and the uncertainties characterized by Gaussian random variables.

For manipulation planning problems that we define there are several complications: 1) The motion models are complex in comparison to a linear model, particularly when contact with obstacles occurs; 2) The performance depends directly on halting the robot (or system state) in some subset of the configuration space (or state space); 3) We would like to use different densities (for probabilistic uncertainty) or bounded sets (for nondeterministic uncertainty) to model uncertainty without the burden of performing a distinct, detailed analysis for each case.

These complications offer limited hope for computation of closed form, exact solutions, which motivates us to consider numerical solutions as a practical alternative. Brost and Christiansen have argued that numerical solutions may be the only practical approach to probabilistic backprojection computation [13]. Given the variety of problems that we wish to consider, a numerical computation method provides the greatest flexibility. We present algorithms that are based on the dynamic programming principle, and apply to a general set of manipulation planning problems. The numerical solutions that we obtain can be improved at the expense of greater computation. The complexity of our method, however, is exponential in the dimension of the state space or information space, which we consider to be practical for a few dimensions.

3 General Definitions and Concepts

In this section we define the general concepts and terminology that form the basis for our framework. Section 3.1 presents the basic concepts that characterize the motions and control of the robot.

Section 3.2 introduces information space concepts, which are used for planning under imperfect sensing. Section 3.3 describes alternative representations of the information states. Section 3.4 defines a robot strategy, the incorporation of a termination condition into a strategy, and the evaluation of a strategy with a loss functional. Section 3.5 presents a specific model that is based on previous manipulation planning research, and will be used for the examples in this paper.

We conceptualize the manipulation planning problem as a dynamic game that is played between two players: the robot, $\mathcal{A}$, 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, while nature makes some decisions that potentially interfere with $\mathcal{A}$. At an abstract level, this general view of robotic manipulation tasks has been advocated in [76].

3.1 States, Stages, and Actions

State space The position of $\mathcal{A}$ in a workspace is represented by a point in an n-dimensional *configuration space*, $\mathcal{C}$, for which n is the number of degrees of freedom of $\mathcal{A}$. For manipulation planning a subset of $\mathcal{C}$, denoted as $\mathcal{C}_{valid}$, is usually defined (see [47] for configuration-space details). This corresponds to points in $\mathcal{C}$ at which either: 1) $\mathcal{A}$ does not touch an obstacle, or 2) the boundary of $\mathcal{A}$ 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 [78], 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 $X = \mathcal{C}_{valid}$ (for collision-free planning, we would use $X = \mathcal{C}_{free}$, which is the interior of $\mathcal{C}_{valid}$). In general, however, we could include dynamics in the state-space representation. In this case robot commands could, for instance, be based on both configuration and velocity.

Stages We consider a discretized representation of time as *stages*, with an index $k \in \{1, 2, \ldots, K\}$. Stage k refers to time $(k-1)\Delta t$. The state of $\mathcal{A}$ at stage k is denoted by x_k. We generally take Δt sufficiently small to approximate continuous paths. The final stage, K is only defined to ease technical considerations as the system evolves toward infinite time. Since we expect the robot 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, which will be discussed shortly. The framework could also be defined in sufficient generality without discretizing time, and consequently defining controlled diffusions [43]; the definitions that would follow require the use of continuous stochastic processes and measure-theoretic concepts. For our context a discretized representation of time facilitates the development of the numerical computation approach. Furthermore, actual robot systems will be limited to discrete-time sampling for acquiring sensor information and executing motion commands. Hence, the discrete-time representation is most appropriate in our context.

Actions An *action* (or command), which is denoted by u_k, can be issued to $\mathcal{A}$ at each stage k. We let U denote the *action space* for $\mathcal{A}$, requiring that $u_k \in U$. We also allow nature to choose actions. Let θ_k denote an action for nature, which is chosen from a set Θ. We consider θ_k to 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 $\mathcal{A}$'s actions, and θ_k^s affects the sensor observations of $\mathcal{A}$.

State transition equation To describe the effect of a robot action with respect to state, we define a *state transition equation* as

$$x_{k+1} = f(x_k, u_k, \theta_k^a). \tag{2}$$

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

Suppose that we are considering nondeterministic uncertainty. We can use the state transition equation to obtain the following subset of X:

$$F_{k+1}(x_k, u_k) = \{f(x_k, u_k, \theta_k^a) \in X | \theta_k^a \in \Theta^a\}. \tag{3}$$

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

Under probabilistic uncertainty, we assume that the probability density function (pdf), $p(\theta_k^a)$, is 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 (see [80] for a treatment of these technical concerns). By using the state transition equation, we can obtain a pdf for x_{k+1}, which is represented by $p(x_{k+1}|x_k, u_k)$.

In general, we will use the notation, F, to refer to minimal subsets of X that can be inferred from the arguments. The role of F in our expressions 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, while p is a generic representation for a density on X.

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 [44] and dynamic game theory [4] to fit our particular context.

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,

$$y_k = h_k(x_k, \theta_k^s), \tag{4}$$

which we term the *observation equation.* A specific example of an observation equation is given in Section 3.5.

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 (see, e.g., [55]). 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 [32], in which the vision system's sampling rate varies according to the amount of processing required to track moving objects in the scene). Such variations are straightforward to consider.

Suppose that we are considering nondeterministic uncertainty. The set of possible values for x_k after only observing y_k can be determined from the observation equation as:

$$F_k(y_k) = \{x_k \in X | y_k = h(x_k, \theta_k^s), \theta_k^s \in \Theta^s\}. \tag{5}$$

Under probabilistic uncertainty, we assume that the pdf, $p(\theta_k^s)$, is known. By using the observation equation, we can obtain a pdf for x_k, which is represented by $p(x_k|y_k)$. 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 $p(x_k|y_k)$ 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 (4) represents the output equation used in control theory, and is also similar to the projection of world states onto sensor values, used in previous robotics contexts (e.g., [24]). Also, such transformations have been studied extensively in statistical image modeling [71, 34, 50] and in sensor error modeling [40].

History and information As discussed in Section 2.2 the relationship between sensing and action has long been considered important in motion planning under uncertainty. The following definitions precisely describe the sensing and action history that $\mathcal{A}$ has available. For a given stage

k, let η_k denote some subset:

$$\eta_k \subseteq \{u_1, u_2, \ldots, u_{k-1}, y_1, y_2, \ldots, y_k\}. \tag{6}$$

The value η_k is a set of past actions and observations that are known to $\mathcal{A}$ at stage k, and is termed the *information state* of $\mathcal{A}$. For instance, we could consider a memoryless robot, in which $\eta_k = y_k$. As another example, we could have a sensorless robot as considered in [35], in which $\eta_k = \{u_1, \ldots, u_{k-1}\}$. We could also consider the stage index k as part of the information space for the purpose of developing robot strategies that involve timing; however, we will not explicitly consider k as part of η_k in this paper.

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 $1 \leq k \leq K$. As it is presently defined, the dimension of the information space grows linearly with the number of stages, which appears impractical. It turns out that alternative representations of the information space can be determined; this is the subject of Section 3.3.

3.3 Representations of the Information State

In this section we present alternative ways to interpret the information space. Under nondeterministic uncertainty, this will be a subset of the state space. Under probabilistic uncertainty, this will be a pdf on the state space.

Consider the case in which the robot has perfect memory. Each η_k then corresponds to 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 $[k(n_1 + n_2) + n_2]$-dimensional. A space that grows significantly with each stage (and becomes infinite-dimensional when $K = \infty$) is very unappealing for designing strategies. The information space representations that are presented in this section do not grow with the number of stages, and provide more intuitively satisfying characterizations of the information state.

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, $F_k(\eta_k)$ represents the set of all states, x_k, that could possibly be the true system state, given the history η_k. By using this approach, we will show that 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.

For practical purposes $F_k(\eta_k)$ can be considered as an alternative representation of the information state. This representation is intuitively satisfying since we can think of $\mathcal{A}$'s uncertainty model as a set representing possible locations in the state space X. In traditional preimage planning research, uncertainty due to imperfect sensing has often been viewed in this way.

We briefly indicate how an information state subset is obtained at a given stage. Initially, we have $F_1(\eta_1)$. We can derive an expression for $F_{k+1}(\eta_{k+1})$ in terms of $F_k(\eta_k)$, u_k, and y_{k+1}. Suppose inductively that we have $F_k(\eta_k)$. First consider the effect on the state space of using the action, u_k. Recall from (3) 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

$$F_{k+1}(\eta_k, u_k) = \bigcup_{x_k \in F_k(\eta_k)} F_{k+1}(x_k, u_k). \tag{7}$$

Note that η_{k+1} can be specified with η_k, u_k, and y_{k+1}. Recall from (5) 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:

$$F_{k+1}(\eta_{k+1}) = F_{k+1}(y_{k+1}) \bigcap F_{k+1}(\eta_k, u_k), \tag{8}$$

which depends on (3) and (5).

If $\mathcal{A}$ does not have perfect memory, then the condition $\{\eta_k, u_k\}$ is replaced in (8) by the appropriate subset of history.

The case of probabilistic uncertainty Under probabilistic uncertainty, the information state can be considered as a conditional density on the state space, denoted as $p(x_k|\eta_k)$. By using this approach, the information state density $p(x_{k+1}|\eta_{k+1})$ can be determined from $p(x_k|\eta_k)$, 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 [44]. This representation is intuitively satisfying since we can think of $\mathcal{A}$'s uncertainty model as a density representing possible locations in the state space X.

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 $p(x_1|\eta_1)$. We can derive an expression for $p(x_{k+1}|\eta_{k+1})$ in terms of $p(x_k|\eta_k)$, u_k, and y_{k+1}. Suppose inductively that we have $p(x_k|\eta_k)$. 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,

$$p(x_{k+1}|\eta_k, u_k) = \int p(x_{k+1}|x_k, u_k) p(x_k|\eta_k) dx_k. \tag{9}$$

Recall from Section 3.1 that $p(x_{k+1}|x_k, u_k)$ 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' rule on X_{k+1} and Y_{k+1}, the following can be obtained:

$$p(x_{k+1}|\eta_{k+1}) = \frac{p(y_{k+1}|x_{k+1})p(x_{k+1}|\eta_k, u_k)}{\int p(y_{k+1}|x_{k+1})p(x_{k+1}|\eta_k, u_k)dx_{k+1}}, \tag{10}$$

which is a function of $p(y_{k+1}|x_{k+1})$ as defined in Section 3.2. A more detailed discussion of (10) can be found in [44]. If $\mathcal{A}$ does not have perfect memory, then the condition $\{\eta_k, u_k\}$ is replaced in (10) by the appropriate subset of history.

3.4 Strategy Concepts

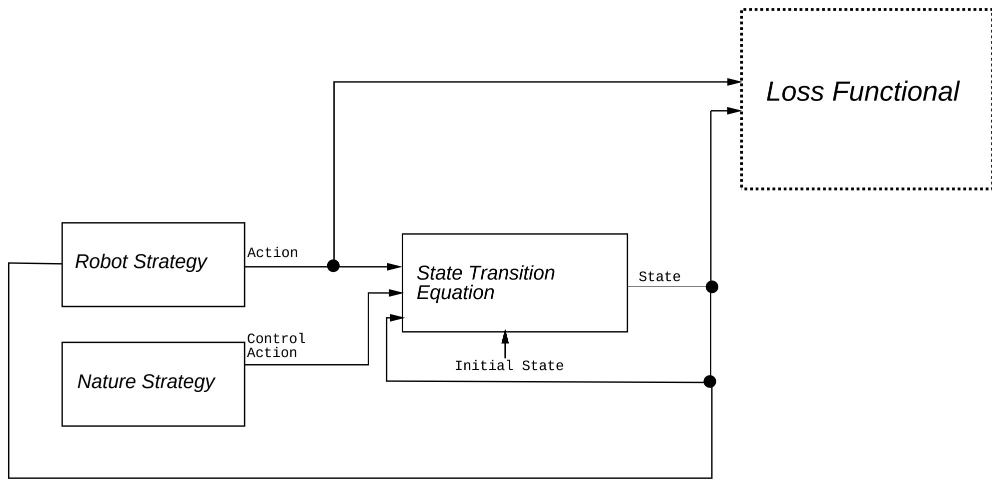
Motion strategies At first it might seem appropriate to define some action u_k for each stage. In general, due to the control uncertainty, it is not possible to predict the trajectory of the robot for given motion commands. It is therefore advantageous to allow the robot to respond to information that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

We consider robot strategies for two cases: perfect information and imperfect information. Figure 3.4 shows block-diagram representations of the concepts that will be described in this section. 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* of $\mathcal{A}$.

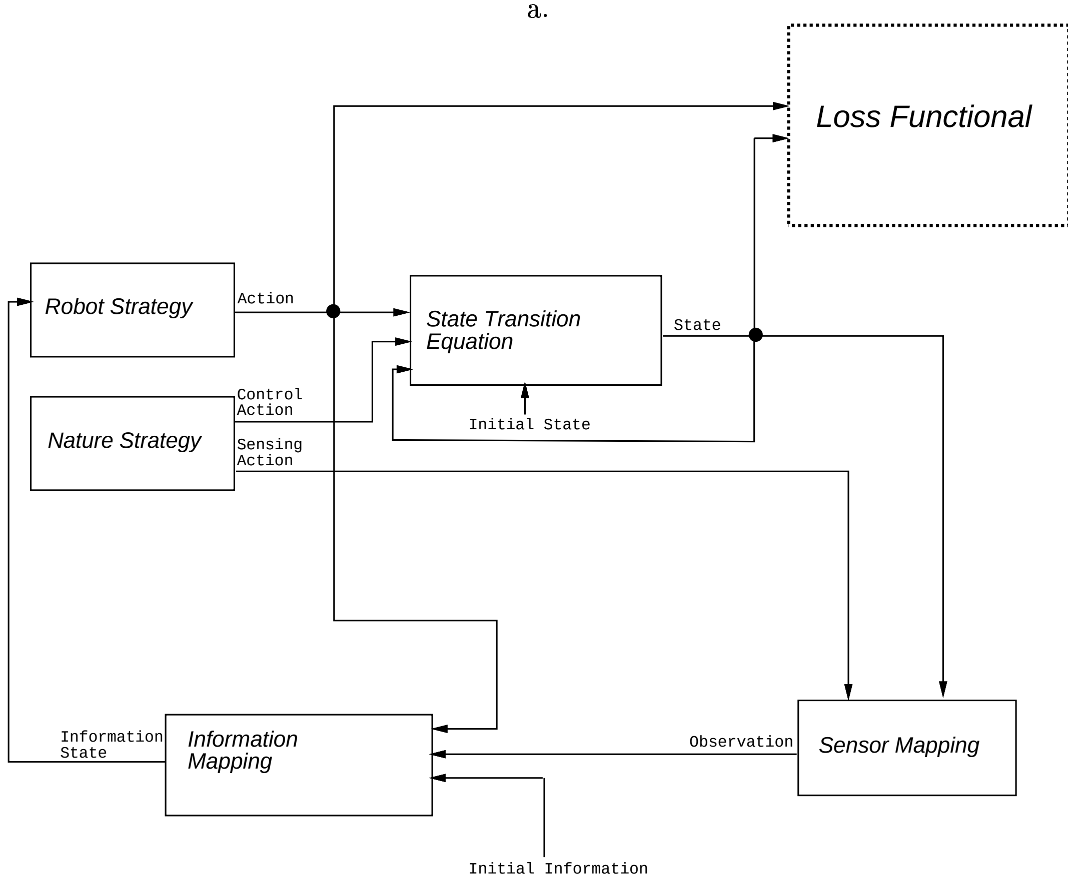
If the robot does not have direct access to state information, its actions are instead conditioned on the information state. In this case we define a *strategy at stage* k of $\mathcal{A}$ as a function $g_k : N_k \rightarrow U$. For each information state, η_k, a strategy yields an action $u_k = g_k(\eta_k)$. In a sense, the "planning" actually occurs in this information space. These strategy concepts are equivalent to a feedback control law [8, 44], and are similar to a conditional multi-step plan in manipulation planning [48].

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 of the robot, u_k. We can define a *pure* or *deterministic* strategy for nature as a deterministic mapping at each stage as $\gamma^\theta_k : X \times U \rightarrow \Theta^a$. 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 $p(\theta_k|x_k, u_k)$). 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.



a.



b.

Figure 4: A dynamic game against nature with: a) perfect information and b) imperfect information.

Termination conditions The notion of a termination condition has been quite useful for formulating robot plans that tell the robot when to halt, based on its current, partial information [30, 48, 53]. The same concept is needed in our context; hence we define a *termination condition* TC_k at each stage by a binary-valued mapping,

$$TC_k : N_k \rightarrow \{true, false\}. \tag{11}$$

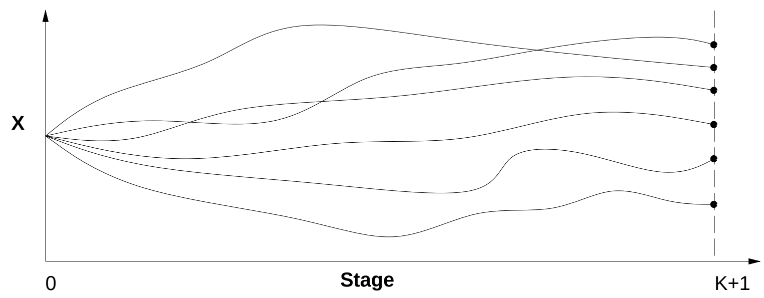
The termination condition can be considered as a special type of action that is available to the robot, which can cause it to halt. With perfect state information N_k is simply replaced by X in (11). We require that if $TC_k = true$, then $TC_{k+1} = true$. This implies that once the termination condition has been applied, it cannot be retracted.

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 is 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 [48]. 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 defining an *optimal stopping rule*, as done in optimal control theory [44].

When perfect information is available, the most appropriate choice for TC_k is $TC_k = true$ if and only if x_k lies in the goal region. The termination condition becomes more interesting under imperfect information. Since the state x_k cannot necessarily be predicted for a given initial state, x_1 and strategy, γ, the particular stage at which the termination condition will be applied is uncertain. Since the robot remains motionless after the termination condition is true, we can, however, consider the resulting states at stage $K+1$. Figure 5 indicates the effect of the termination condition.

Loss functionals We encode the objectives that are to be achieved by a nonnegative real-valued functional $L(x_1, ..., x_{K+1}, u_1, ..., u_K, TC)$, called the *loss functional.* Note that the loss functional is not written as a function of γ, but in terms of the *actual* executed trajectory and action histories. The ultimate goal of the planner is to determine a strategy g and termination condition TC that causes L to be optimized in an appropriate sense. As will be discussed in Section 6, an equilibrium solution will be determined for nondeterministic uncertainty, and the minimal solution in the expected sense will be determined for probabilistic uncertainty.

We assume that a loss functional is of the following additive form, which, except for the termi-



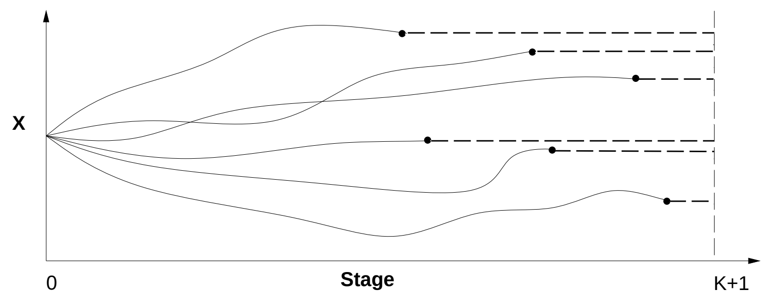


Figure 5: The termination condition forces the robot to halt at a given state. The particular stage is *a priori* uncertain for a given initial state and strategy.

nation condition, is often used in optimal control theory [44]:

$$L(x_1, ..., x_{K+1}, u_1, ..., u_K, TC) = \sum_{k=1}^{K} l_k(x_k, u_k, TC_k) + l_{K+1}(x_{K+1}). \tag{12}$$

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.

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

$$L(x_1, ...x_{K+1}, u_1, ..., u_K, TC) = \begin{cases} 0 & \text{if } x_{K+1} \in G \\ 1 & \text{otherwise} \end{cases} . \tag{13}$$

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 [20]).

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:

$$L(x_1, ...x_{K+1}, u_1, ..., u_K, TC) = \begin{cases} \sum_{k=1}^{K+1} l(u_k, TC_k) & \text{if } x_{K+1} \in G \\ C_f & \text{otherwise} \end{cases} . \tag{14}$$

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. As 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.

Stationarity It turns out that the optimal action u_k, and termination condition, TC_k, do not depend on the stage index, k, for the problems that we consider. This is because the optimal strategies for the problems in this context are *stationary*, which is a concept that is defined and discussed in [4, 44]. If the state transition equation, free configuration space, or loss functional were stage dependent, then stationarity would be lost. For instance, it might be the case that the workspace contains a known, moving obstacle. Many quantities would depend on the stage index, resulting in a robot strategy that was time-dependent. This is analogous to the configuration-time space, $\mathcal{CT}$, as considered in [47] for planning amidst moving obstacles. In general, our framework supports the analysis of time-dependent problems; however, we preclude them from consideration in this paper.

3.5 Specific Model Details

In this section we present specific definitions of a state transition equation and a observation equation. These models are inspired by those used in previous manipulation planning research, and are used for our examples throughout this paper. In general, a variety of other types of models could be defined.

The control model Suppose the robot $\mathcal{A}$ is a polygon translating in the plane amidst polygonal obstacles. The action set of $\mathcal{A}$ 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. First consider the state transition equation when $x_k \in \mathcal{C}_{free}$, at a distance of at least $\|v\|\Delta t$ away from the obstacles. If $\mathcal{A}$ chooses action u_k from state x_k, and nature chooses θ_k^a, then x_{k+1} is given by

$$f(x_k, u_k, \theta_k^a) = x_k + \|v\|\Delta t \left[\begin{array}{c} cos(u_k + \theta_k^a) \\ sin(u_k + \theta_k^a) \end{array} \right]. \tag{15}$$

Let $\mathcal{C}_{contact}$ represent the boundary of $\mathcal{C}_{free}$ (hence $\mathcal{C}_{contact} = \mathcal{C}_{valid} - \mathcal{C}_{free}$). If $x_k \in \mathcal{C}_{contact}$, 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 (see e.g., [78]) 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 $\mathcal{C}_{free}$. The remaining cases describe when the robot moves from $\mathcal{C}_{free}$ to $\mathcal{C}_{contact}$, from $\mathcal{C}_{contact}$ to $\mathcal{C}_{free}$, or from one edge in $\mathcal{C}_{valid}$ to another. These cases are straightforward to define with the generalized damper model, as discussed in Section 2.1.

This model of uncertainty does not correspond completely to the model used in [13, 48, 53]. In our model nature repeatedly acts at each time Δt. To correspond more closely with the traditional model, no additional uncertainty would be introduced if $u_k = u_{k+1}$. Under the control model that we have defined this would mean $\theta_{k+1} = 0$ if $u_k = u_{k+1}$. This model can be implemented by including the previous u_k as a component of the state space.

The sensing model We now present a sensing model that is similar to that used in [13, 30, 48]. This sensing model will be used in Section 6.6. The robot $\mathcal{A}$ 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 3-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

$$p(\theta_k^{s,p}) = \left\{ \begin{array}{cl} \frac{2}{\pi\epsilon_p^2} & \text{for } \|\theta_k^{s,p}\| < \epsilon_p \\ 0 & \text{otherwise} \end{array} \right. , \tag{16}$$

for some prespecified radius ϵ_p, and $\theta_k^{s,p}$ is 2-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 $x_k \in \mathcal{C}_{contact}$, and the true normal is given by $\alpha(x_k)$, or 2) An empty value, $\emptyset$, when the robot is in $\mathcal{C}_{free}$. When the robot configuration lies in $\mathcal{C}_{contact}$ and probabilistic uncertainty is in use, then we might choose

$$p(\theta_k^{s,f}) = \begin{cases} \frac{1}{2\epsilon_f} & \text{for } |\theta_k^{s,f}| < \epsilon_f \\ 0 & \text{otherwise} \end{cases}, \tag{17}$$

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.

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 (e.g., [13, 30]). 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

Recall from Section 3.4 that under nondeterministic uncertainty, the strategy of nature γ^θ is deterministic, but unknown to the robot. Under perfect sensing, γ^θ defines a specific action $\theta_k \in \Theta$ that will be taken by nature at every stage, k. The resulting nondeterministic forward projection will include all of the system states that could result from the various actions of nature. In this way, it yields a set of possible futures under the implementation of a strategy.

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

$$F_{k+1}(x_k, g) = F_{k+1}(x_k, g_k(x_k)) = F_{k+1}(x_k, u_k), \tag{18}$$

in which $F_{k+1}(x_k, u_k)$ is given by (3). 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 (3), 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 next be taken into account to yield

$$F_{k+2}(x_k, g) = \{f(x_{k+1}, u_{k+1}, \theta^a_{k+1}) \in X | x_{k+1} \in F_{k+1}(x_k, g), \;\; \theta^a_{k+1} \in \Theta^a\}. \tag{19}$$

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

$$F_{k+2}(x_k, g) = \bigcup_{x_{k+1} \in F_{k+1}(x_k, g)} F_{k+2}(x_{k+1}, g). \tag{20}$$

One interpretation for this representation is that from each possible state in the single-stage forward projection from stage k to stage $k+1$, the single-stage forward projection from stage $k+1$ to stage $k+2$ is possible. The resulting $x_{k+2} \in X$ represents the union of all of these single-stage forward projections (see Figure 6).

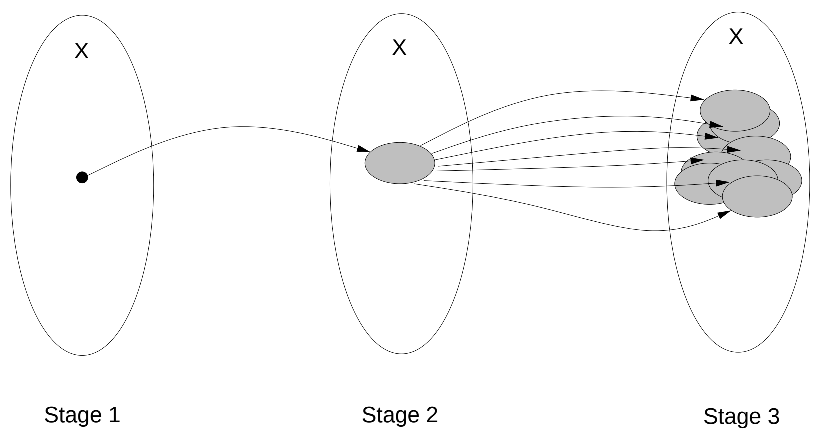


Figure 6: A depiction of a two-stage forward projection under nondeterministic control uncertainty.

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

$$F_k(x_1, g) = \bigcup_{x_2 \in F_2(x_1, g)} \; \bigcup_{x_3 \in F_3(x_2, g)} \cdots \bigcup_{x_{k-1} \in F_{k-1}(x_{k-2}, g)} F_k(x_{k-1}, g), \tag{21}$$

which is an extension of (20). 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. Recall 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 Equations (18) to (21).

The classical reachability and recognizability concepts [28] 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 k. We can say that the goal is *reachable* if for every possible state trajectory, $\{x_1, .., x_{K+1}\}$ (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 guaranteeed to terminate in the goal region.

4.1.2 The imperfect information case

We consider, as in the perfect information case, a deterministic strategy for nature, γ^θ, which is unknown to the robot. We will define the forward projection in a manner similar to the perfect information case.

The previous forward projection (21) 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 appling an action u_k and receiving sensor observation y_{k+1}, we obtain $\tilde{F}_{k+1}(\eta_k, g(\eta_k)) = \tilde{F}_{k+1}(\eta_k, u_k) =$

$$\{\eta_{k+1} \in N_{k+1} | \eta_k \cup \{u_k, y_{k+1}\} \subset \eta_{k+1},\ x_{k+1} \in F_{k+1}(x_k, u_k) \cap F_{k+1}(y_{k+1}),\ x_k \in F_k(\eta_k)\}, \tag{22}$$

which depends on (3) and (5), and $F_k(\eta_k) \subseteq X$ is the subset representation of the information state from Section 3.2.

To obtain the information forward projection from stage 1 to some stage k, we can iteratively apply (22).

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

$$\bigcup_{\eta_k \in \tilde{F}_k(\eta_1, g)} F_k(\eta_k). \tag{23}$$

The goal is *reachable at stage k* if set defined in (23) 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

Under nondeterministic uncertainty, the forward projections yielded subsets of the state space; however, for probabilistic uncertainty, the forward projections will be specified by pdf's on the state space. We use the notation $p(x_j|x_i, g)$ 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

$$p(x_{k+1}|x_k, g) = p(x_{k+1}|x_k, g_k(x_k)) = p(x_{k+1}|x_k, u_k). \tag{24}$$

Recall from Section 3.1 that $p(x_{k+1}|x_k, u_k)$ 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:

$$p(x_{k+2}|x_k, g) = \int p(x_{k+2}|x_{k+1}, g_{k+1}(x_{k+1}))p(x_{k+1}|x_k, g_k(x_k))dx_{k+1}. \tag{25}$$

The result after applying two actions is a posterior density on X. Figure 7 depicts the forward projection; this can be contrasted to Figure 6, which showed the forward projection under nondeterministic uncertainty.

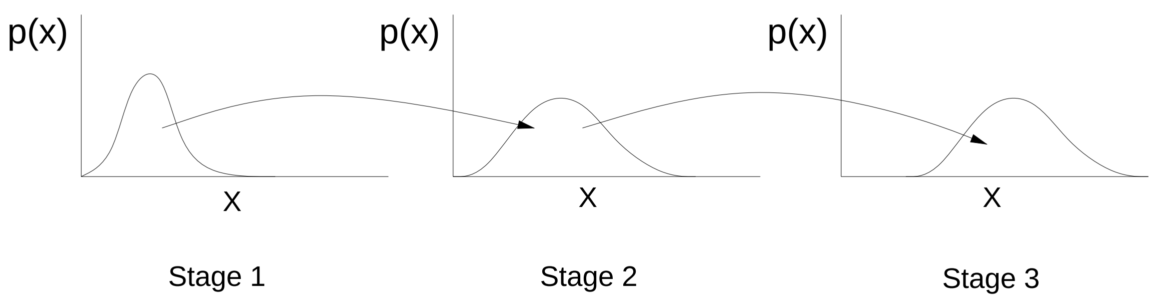


Figure 7: A depiction of a two-stage forward projection under probabilistic control uncertainty.

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

$$p(x_k|x_1, g) = \int p(x_k|x_{k-1}, g_{k-1}(x_{k-1}))p(x_{k-1}|x_{k-2}, g_{k-2}(x_{k-2}))\cdots p(x_2|x_1, g_1(x_1))dx_2dx_3\cdots dx_{k-1}. \tag{26}$$

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 $p(x_{K+1}|x_1,\gamma)$ 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

$$\int_G p(x_k|x_1,g)dx_k, \tag{27}$$

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

The probability that the goal is recognizably achieved is

$$\int_G p(x_{K+1}|x_1,\gamma)dx_{K+1}. \tag{28}$$

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 $p(\eta_k|\eta_1,\gamma)$, resulting in a pdf on the information space.

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

$$p(x_k|\eta_1,g) = \int p(x_k|\eta_{k-1},g_{k-1}(\eta_{k-1}))p(\eta_{k-1}|\eta_{k-2},g_{k-2}(\eta_{k-2}))\cdots p(\eta_2|\eta_1,g_1(\eta_1))d\eta_{k-1}\cdots d\eta_2. \tag{29}$$

The first term in the integrand can be determined using (9). Each of the remaining terms can be reduced to

$$p(\eta_{k+1}|\eta_k,g_k(\eta_k)) = p(y_1,\ldots,y_{k+1},u_1,\ldots,u_k|y_1,\ldots,y_k,u_1,\ldots,u_k) = p(y_{k+1}|\eta_k,u_k). \tag{30}$$

This reduction occurs because most of the sensing and action history appears on both sides of the density expression. The right side of (30) can be further reduced to

$$p(y_{k+1}|\eta_k,u_k) =$$

$$\int p(y_{k+1}|x_{k+1})p(x_{k+1}|\eta_k,u_k)dx_{k+1} = \int\int p(y_{k+1}|x_{k+1})p(x_{k+1}|x_k,u_k)p(x_k|\eta_k)dx_k dx_{k+1}, \tag{31}$$

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

To include the termination condition we replace g by γ above to obtain $p(x_{K+1}|\eta_1,\gamma)$. 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. These will also be compared to the forward projections shown in this section.

We have computed forward projection examples in a straightforward way, by using a discretized, array representation for the state space. Under nondeterministic uncertainty, this be considered as a bit-map representation of the forward projection. Under probabilistic uncertainty, the representation approximates a pdf on X by using a fine grid. 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 (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., $\mathcal{C}_{valid} \subseteq \Re^2$), in which each coordinate is constrained to lie in the interval $[0,100]$. This could, for example, represent the configuration space of a planar robot that is capable of translating in the plane. The obstacles in the workspace will be indicated in figures by gray regions, and a black region will represent the goal.

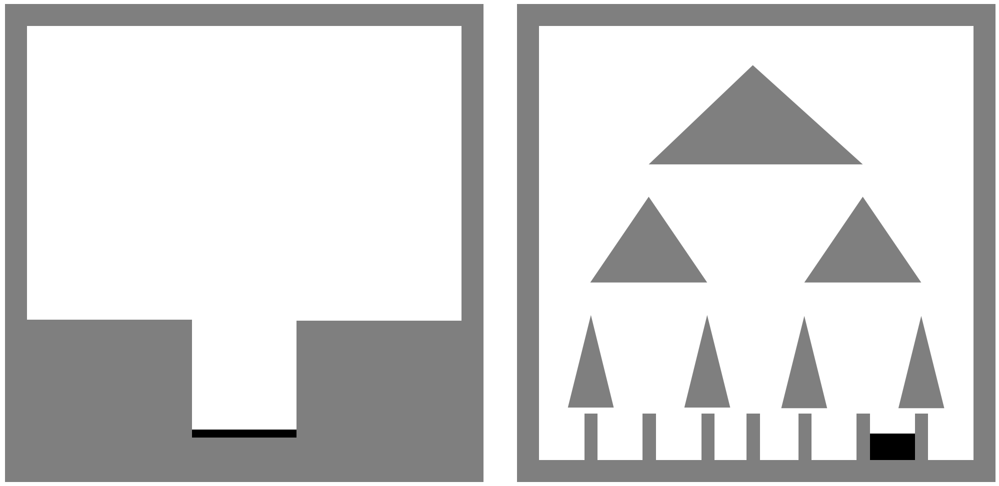
The first example is depicted in Figure 8.a, and can be considered as a configuration-space representation of the classical *peg-in-hole* problem (see for example [30, 13, 48, 53]). The second example is depicted in Figure 8.b, and is designed to spread the possible locations of the robot over a large 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 3 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 9 and 10 show the forward projections at several different stages, under nondeterministic uncertainty. Figures 11 and 12 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 one. This effect can be seen in Figure 12 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.

A significant distinction between probabilistic and nondeterministic forward projections becomes clear after examining these results. Consider the problem from Figure 8.a. The nondeterministic forward projections indicate that very little prediction is possible since the set of possible states grows very quickly. By observing the probabilistic forward projection, however, most of the probability mass appears to terminate in the goal region. This corresponds closely to the arguments about worst-case analysis eliminating many reasonable motion plans; these arguments were given in Section 2.2, and also in [13, 14].

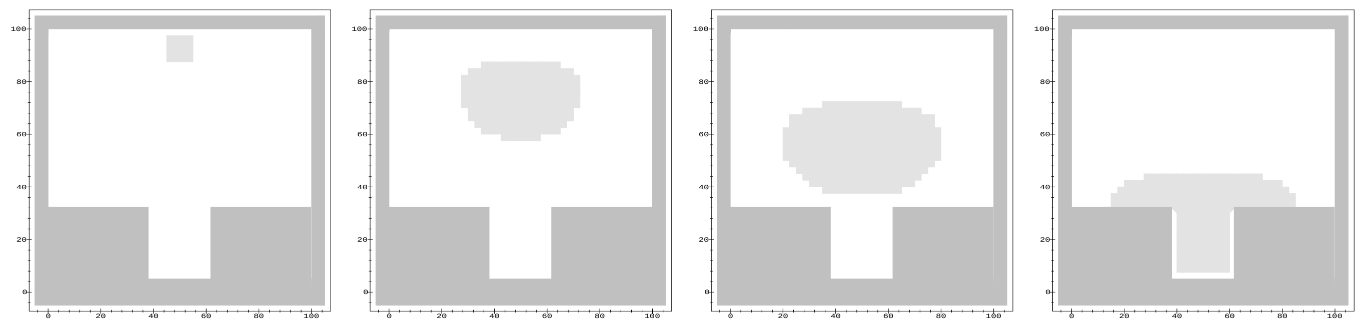
It is interesting to note that the densities in Figure 11 appear to be Gaussian, even though the uncertainty model is specified as a uniform density. Since the effects of the control uncertainty combine additively, the Central Limit Theorem [80] implies that the densities will tend toward Gaussian. Even in Figure 12 at $k = 11$, the probability mass appears to be two disjoint Gaussians. We have observed that as the probability mass divides because of obstacles, the individual components also tend toward being Gaussian. These results indicate that there may be little sensitivity to the particular choice of error model, as long as the mean and variance remain constant.

Figure 13 shows some additional results that are closely related to the forward projections, and apply to the case of probabilistic uncertainty. By specifying a strategy, a random process is automatically defined. The transition probabilities are known, and are derived in part from the densities that represent nature's actions. The random process can be considered as a probability space in which the elements are sample paths. Each sample path corresponds to one possible trajectory that could occur under the implementation of the strategy. We can iteratively sample actions for nature and generate a path that represents the state trajectory under the implementation of a strategy. The figures show several sample paths that were superimposed.



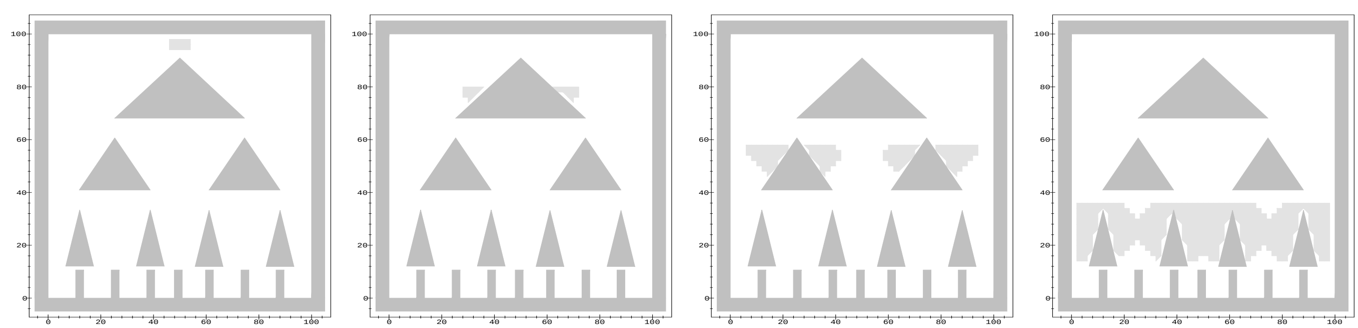
a. b.

Figure 8: 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.



$k = 2$ $k = 9$ $k = 16$ $k = 27$

Figure 9: The nondeterministic forward projection is represented by the lightly-shaded regions.



$k = 1$ $k = 11$ $k = 22$ $k = 33$

Figure 10: The nondeterministic forward projection is represented by the lightly-shaded regions.

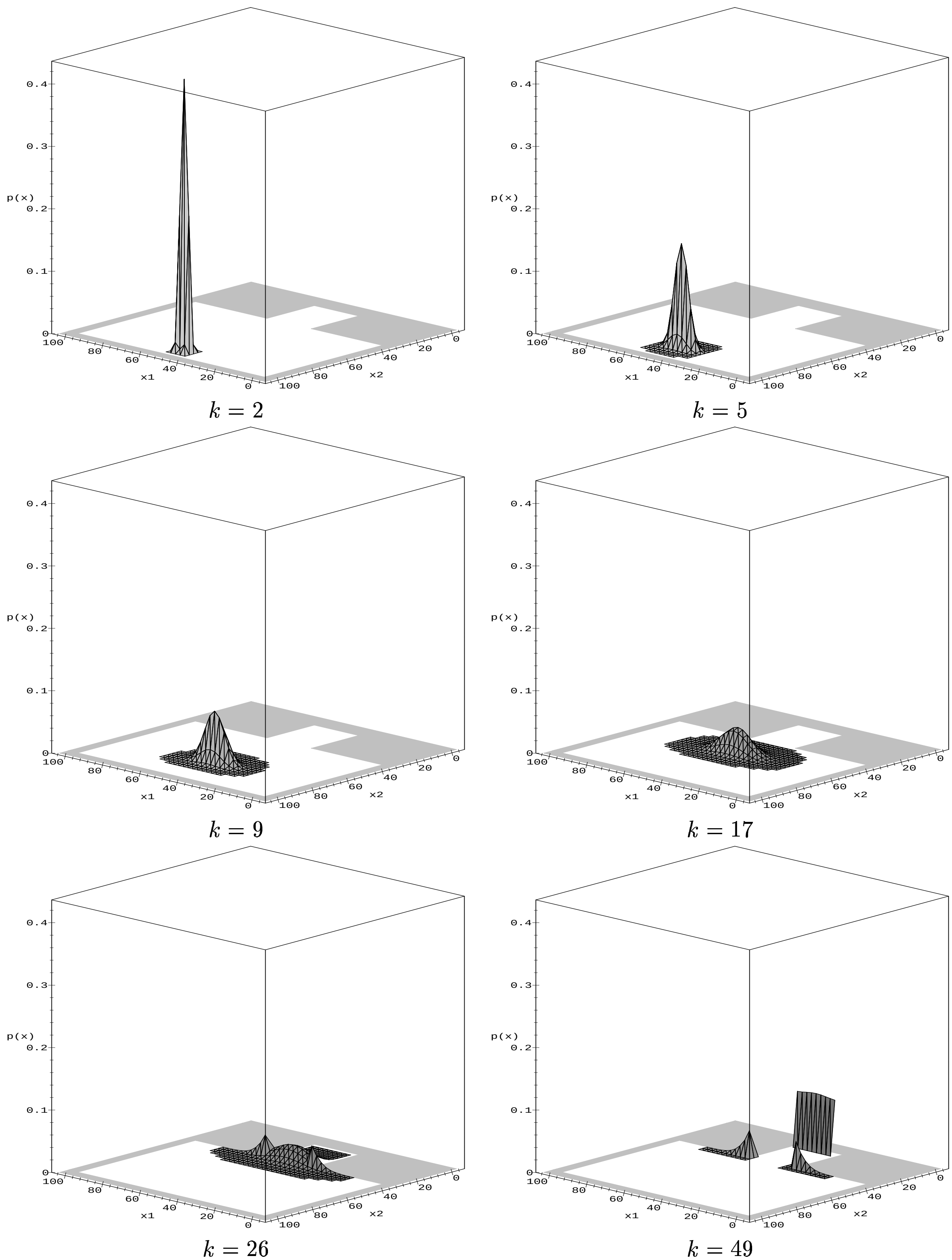


Figure 11: The forward projection at several stages, under probabilistic uncertainty.

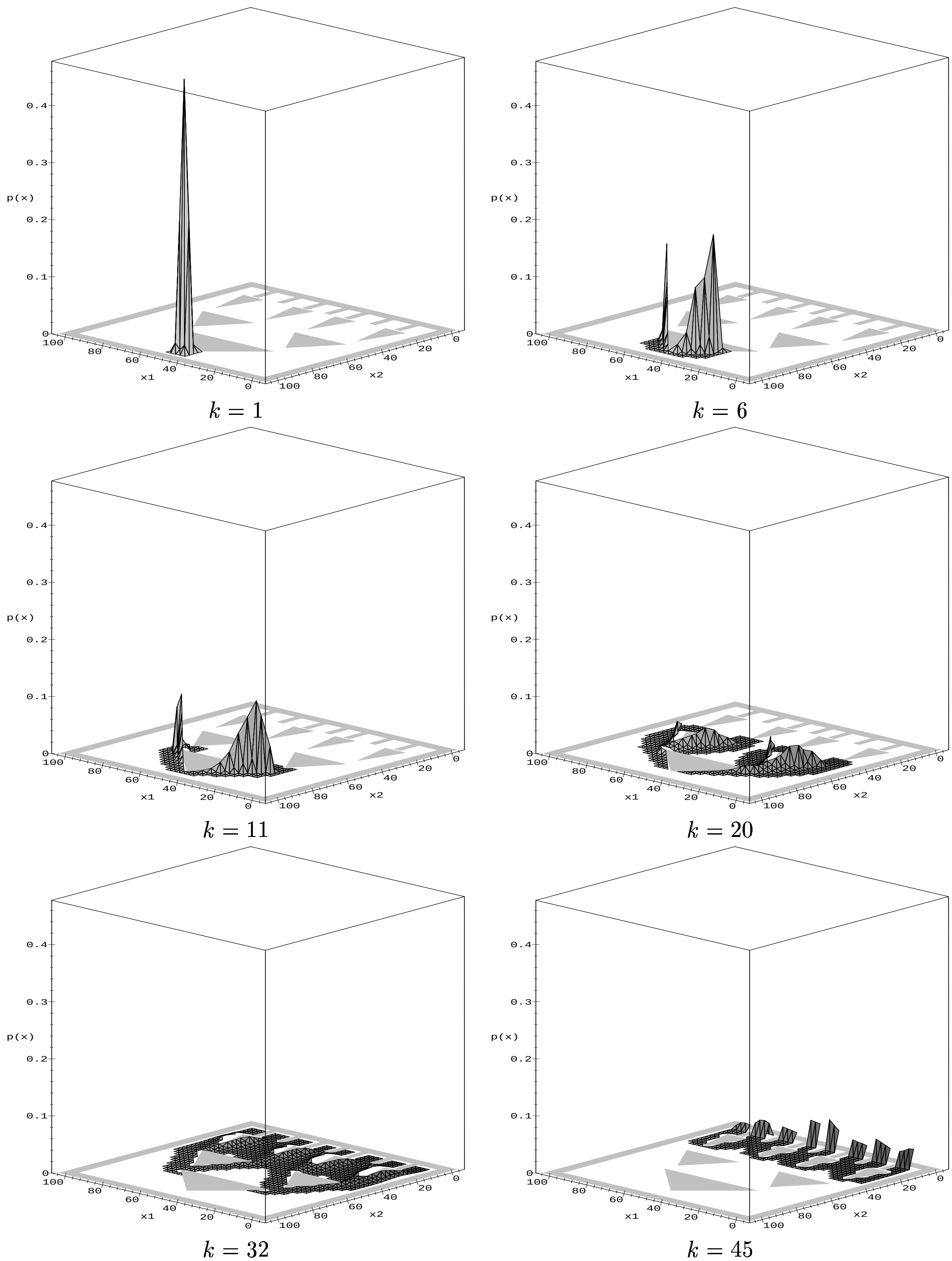
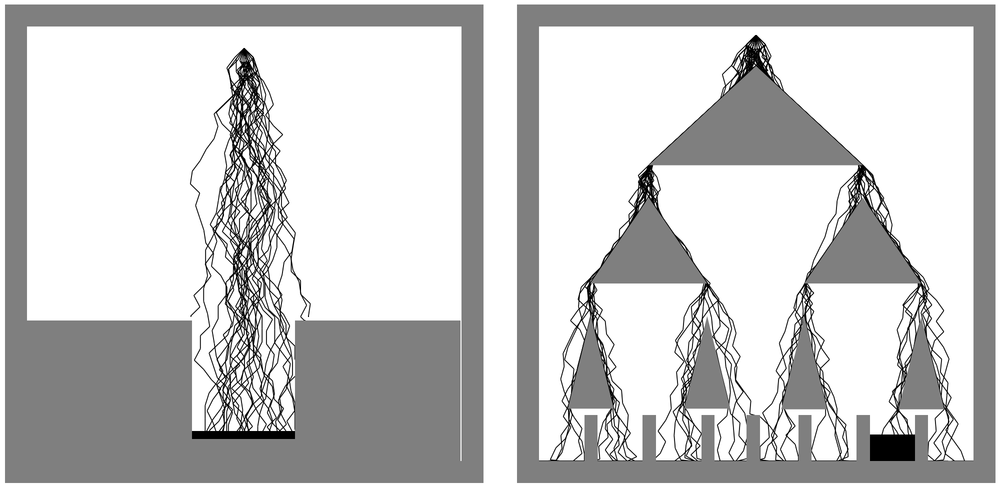


Figure 12: The forward projection at several stages, under probabilistic uncertainty.



a. b.

Figure 13: Each figure shows sample paths of the random process that results from the given strategy (assuming probabilistic uncertainty). This can be considered as an alternative way to view the forward projection.

5 Performance Preimages

In this section we present performance preimages for each of the four uncertainty cases that are 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 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

We combine the classical preimage concept with the loss functional to evaluate a given strategy. Suppose for a moment that the strategy for nature γ^θ was given to the robot; then the loss for 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

$$\check{L}(x_1, \gamma) = \sup_{\gamma^\theta \in \Gamma^\theta} L(x_1, \gamma, \gamma^\theta), \tag{32}$$

which represents the maximum loss 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 [3].

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 (13) for a strategy that consists of a fixed action, repeated at every stage (which 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.

We will next 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

$$\check{\pi}_x(\gamma, R) = \{x_1 \in X | \check{L}(x_1, \gamma) \in R\}. \tag{33}$$

The set $\check{\pi}_x(\gamma, R) \subseteq X$ indicates places in the state space from which if $\mathcal{A}$ 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 (13), $\check{\pi}_x(\gamma, 0)$ yields the classical preimage. Under (14) and $R = [0, m)$ we obtain a performance preimage that indicates all $x_1 \in X$ from which the goal will be achieved with a loss that is guaranteed to be less than m.

If we replace γ with g, and replace the condition "if $x_{K+1} \in G$" in (13) by "if $x_k \in G$ for some k" then $\check{\pi}(g, 0)$ yields a *backprojection* that is similar to that appearing in [30].

5.1.2 The imperfect information case

We next consider the case in which the robot has imperfect state information. Let $L(\eta_1, \gamma, \gamma^\theta)$ represent the loss that is obtained if the robot implements γ and nature implements γ^θ. By replacing x_1 by η_1 in (32) we define

$$\check{L}(\eta_1, \gamma) = \sup_{\gamma^\theta \in \Gamma^\theta} L(\eta_1, \gamma, \gamma^\theta), \tag{34}$$

which represents maximum amount of loss that the robot could receive under the implementation of γ, while starting from η_1. Note that here γ^θ represents both control and sensing actions.

Note that $\check{L}(\eta_1, \gamma)$ can be considered as a real-valued function of η_1 for a fixed γ. Consider some subset of the reals, $R \subseteq \Re$. We define the *performance preimage on* N_1 as a subset of N_1, denoted by $\check{\pi}(\gamma, R)$, that is given by

$$\check{\pi}(\gamma, R) = \{\eta_1 \in N_1 | \check{L}(\eta_1, \gamma) \in R\}. \tag{35}$$

The set $\check{\pi}(\gamma, R) \subseteq N_1$ indicates places in the information space from which if $\mathcal{A}$ begins, the loss will lie in R. Concepts such as isoperformance classes can also be defined on the information space. We could also consider performance preimages on any N_k.

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.

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 (2), (4), 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

$$\bar{L}(x_1, \gamma) = \int L(x_1, \gamma, \theta) p(\theta) d\theta, \tag{36}$$

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,

$$\bar{\pi}_x(\gamma, R) = \{x_1 \in X | \bar{L}(x_1, \gamma) \in R\}. \tag{37}$$

The set $\bar{\pi}_x(\gamma, R) \subseteq X$ indicates places in the state space from which if $\mathcal{A}$ 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 $r \geq 0$, then 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 (13) implies that we are only interested in achieving the 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 (13) by "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 [13]. 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. Although in our case, it is more appropriate to claim that the goal will be achieved with probability one.

5.2.2 The imperfect information case

In general, the robot $\mathcal{A}$ will begin in some uncertain initial state. Therefore, we also consider performance preimages on the robot's initial information space, N_1. The preimages represent places in the initial information space where if $\mathcal{A}$ begins, the expected performance will lie within some $R \subseteq \Re$. This result specializes to (37) in the case of a given initial state, $\eta_1 = y_1 = x_1$. As a minor extension, one could also consider performance preimages on any information space N_k.

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

$$\bar{L}(\eta_1, \gamma) = \int L(\eta_1, \gamma, \theta) p(\theta) d\theta. \tag{38}$$

For a subset $R \subseteq \Re$, we define the *performance preimage on* N_1 as a subset of N_1, denoted by $\bar{\pi}(\gamma, R)$, that is given by

$$\bar{\pi}(\gamma, R) = \{\eta_1 \in N_1 | \bar{L}(\eta_1, \gamma) \in R\}. \tag{39}$$

5.3 Computed Examples

In this section we present several computed preimages. As done 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. It will be seen that representation of the performance preimages is a by-product of determining the optimal strategy. The evaluation of a given strategy, which is done in this section, can be considered as the trivial case of computing the optimal strategy for which there is only choice in the space of possible strategies.

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 (13) for all of the examples in this section since we have observed that (14) produces very similar curves under fixed motion commands; when considering optimal strategies, however, the differences between the two loss functionals becomes much more important.

Figure 14.a 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 14.a corresponds closely to the classical preimage that has been determined for this problem in previous manipulation planning research (e.g., [30, 47]).

In Figure 14.b we show probabilistic backprojections that are quite similar that those that appear in [12, 13, 14]. Figure 15 shows a three-dimensional plot of $\bar{L}(x_1, \gamma)$. 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 figures shows isoprobability curves from $\bar{\pi}_x(0.2)$ to $\bar{\pi}_x(0.9)$, at evenly spaced probability values. The innermost curve represents the $\bar{\pi}(0.2)$.

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 14.c. 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 14.d shows the resulting contours (with the same preimage values as used previously).

We conclude this section with two additional examples, which are depicted in Figures 16.a and 16.b. Probabilistic performance preimages are shown for these problems in Figures 16.c and 16.d, respectively. These examples differ from the previous example since 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 16.c, 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.

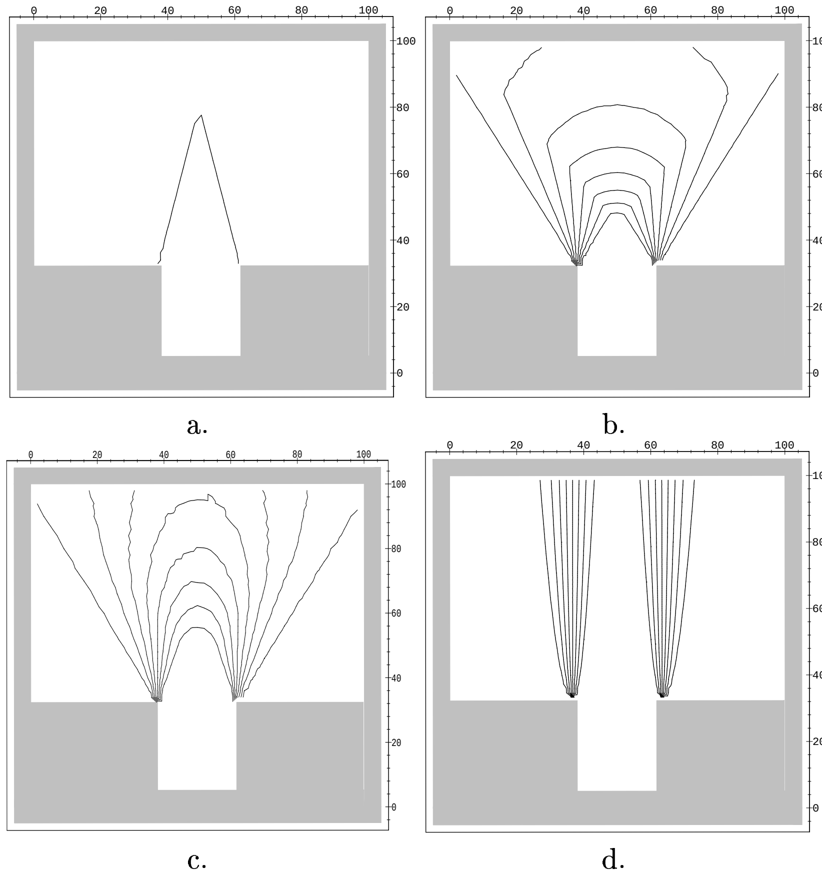


Figure 14: Several computed performance preimages

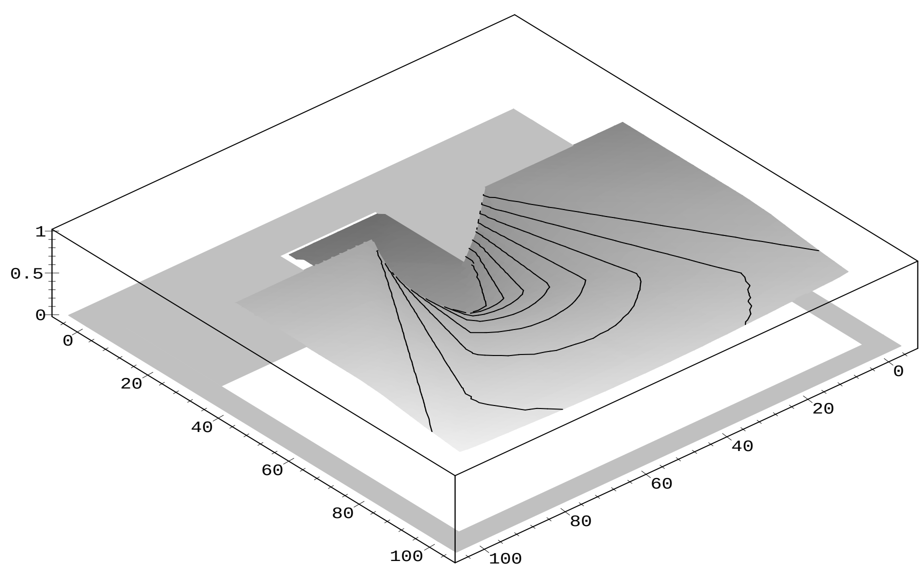


Figure 15: A plot of $\bar{L}$ under probabilistic uncertainty.

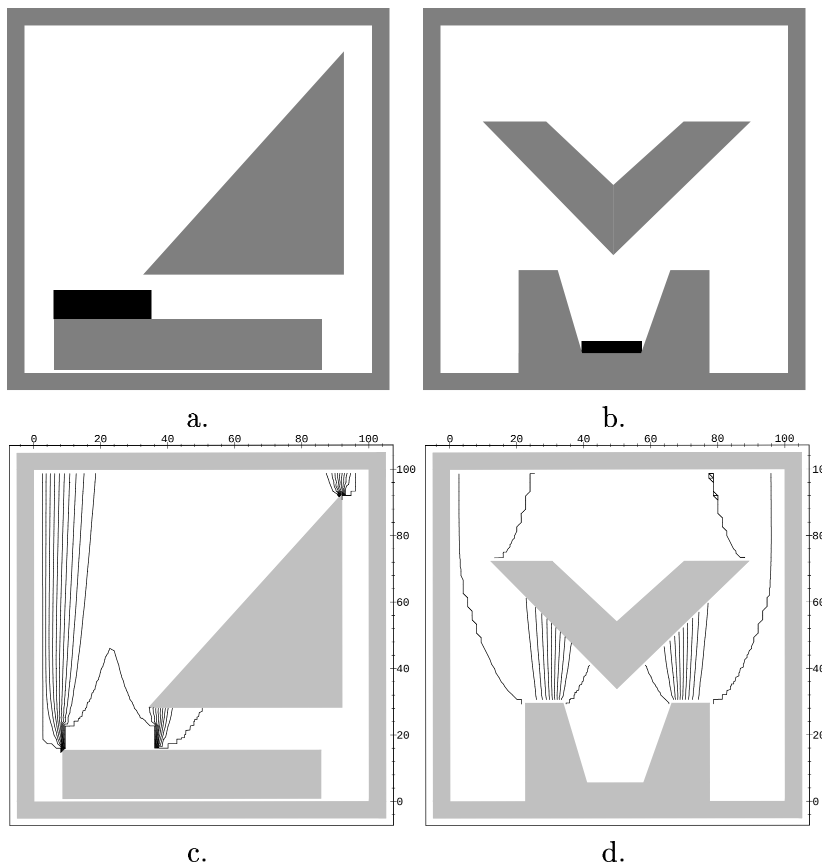


Figure 16: Two computed performance preimages. In these examples, the obstacles displace the curves.

6 Designing Optimal Strategies

Sections 4 and 5 have presented methods that evaluate a *given* strategy. In this section define concepts of optimality, and present a computational approach that *selects* a strategy. Section 6.1 defines optimality conditions for each of the four uncertainty types that have been considered throughout the paper. Section 6.2 presents the principle of optimality, which represents a recursive constraint that significantly reduces the amount of computation necessary to determine a solution. Section 6.3 discusses computational issues that are involved in computing solutions due to uncertainty in control. Section 6.4 discusses additional issues that are involved when planning in the information space as a result of imperfect information. Section 6.5 discusses the computational performance. Finally, Section 6.6 presents computed optimal strategies, which includes forward projections and preimages.

6.1 Defining Optimality

A strategy, γ, has been used in this work to precisely describe the behavior of the robot with respect to state or information. The design problem is to select the most desirable strategy, γ^*, from the space of allowable strategies Γ. We have already defined a loss functional that evaluates the state trajectory. Without the effects of nature's actions, the design problem would simply be formulated as selecting the strategy that produces a trajectory that minimizes the loss. 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.

6.1.1 Optimality under nondeterministic uncertainty

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

$$\check{L}(x_1, \gamma^*) = \inf_{\gamma \in \Gamma} \check{L}(x_1, \gamma) = \inf_{\gamma \in \Gamma} \sup_{\gamma^\theta \in \Gamma^\theta} L(x_1, \gamma, \gamma^\theta) \tag{40}$$

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 [3].

If we use the loss functional, (13), the space of strategies Γ can be partitioned into two equivalence classes: those that achieve the goal (resulting in a worst-case loss of one), and those that may fail to achieve the goal (resulting in a worst-case loss of zero). Any strategy in the first equivalence class satisfies (40), 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 (14), (40) can be considered as partitioning Γ into many more classes; this induces preferences on the set of strategies that achieve the goal.

Imperfect information To obtain the concept of optimality for the imperfect information case, the objective is to select $\gamma^* \in \Gamma$ such that

$$\check{L}(\eta_1, \gamma^*) = \inf_{\gamma \in \Gamma} \check{L}(\eta_1, \gamma) = \inf_{\gamma \in \Gamma} \sup_{\gamma^\theta \in \Gamma^\theta} L(\eta_1, \gamma, \gamma^\theta) \tag{41}$$

for all $\eta_1 \in N_1$. Equation (41) is almost identical to (40), except that the state space is replaced by the information space.

6.1.2 Optimality under probabilistic uncertainty

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

$$\bar{L}(x_1, \gamma^*) = \inf_{\gamma \in \Gamma} \bar{L}(x_1, \gamma) = \inf_{\gamma \in \Gamma} \int L(x_1, \gamma, \theta) p(\theta) d\theta \tag{42}$$

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 [44].

Imperfect information Under imperfect uncertainty, the state space in (42) is replaced by the information space to obtain

$$\bar{L}(\eta_1, \gamma^*) = \inf_{\gamma \in \Gamma} \bar{L}(\eta_1, \gamma) = \inf_{\gamma \in \Gamma} \int L(\eta_1, \gamma, \theta) p(\theta) d\theta \tag{43}$$

for all $\eta_1 \in N_1$.

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. Bertsekas considers solvable problems to typically be an exception in applications

[8]. 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 [2, 4, 15]. As an example, the principle of optimality forms the basis of the analytic solution to the linear-quadratic Gaussian (LQG) optimal control problem [44].

Since few problems can be solved analytically, there has been a large focus on numerical dynamic programming procedures [8, 45, 46]. In several motion planning approaches, numerical dynamic programming techniques have found successful application [5, 38, 59, 74]. In this section we formulate the principle of optimality for each of the four types of uncertainty that are considered in this paper. For each type, this principle can be viewed as a useful constraint that significantly simplifies the amount of computation that required to determine an optimal solution.

6.2.1 The nondeterministic case

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

$$\check{L}_k^*(x_k) = \sup_{\gamma^\theta \in \Gamma^\theta} \left\{ \sum_{i=k}^{K} l_i(x_i, \gamma_i^*(x_i)) + l_{K+1}(x_{K+1}) \right\}, \tag{44}$$

in which Γ_k refers to the set of possible choices for γ_k. 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 [8].

The principle of optimality [4] states that $\check{L}_k^*(x_k)$ can be obtained from $\check{L}_{k+1}^*(x_{k+1})$ by the following recurrence:

$$\check{L}_k^*(x_k) = \inf_{\gamma_k \in \Gamma_k} \sup_{\theta_k^a \in \Theta_k^a} \left\{ l_k(x_k, \gamma_k(x_k)) + \check{L}_{k+1}^*(f(x_k, u_k, \theta_k^a)) \right\}. \tag{45}$$

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 (45) to obtain the optimal actions. At stage $K+1$, we can use the last term of (13) 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 (45). Using the $u_K \in U$ and TC_k that minimize (45) at x_K, we define $\gamma_K^*(x_K) = \{u_K, TC_K\}$. We then apply (45) 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 loss function, $\check{L}_1^*$, shares similarities with the concept of a global navigation function in motion planning [47, 66], 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 [5, 38, 59, 74]; for instance, the *wavefront expansion method* that is described in [47] 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.

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

$$\check{L}_k^*(\eta_k) = \sup_{x_k \in F_k(\eta_k)} \check{L}_k^*(x_k). \tag{46}$$

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 (18). 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:

$$\check{L}_k^*(\eta_k) = \inf_{\gamma_k \in \Gamma_k} \sup_{\eta_{k+1} \in \tilde{F}_{k+1}(\eta_k, u_k)} \left\{ l(\gamma_k(\eta_k)) + \check{L}_{k+1}^*(\eta_{k+1}) \right\}, \tag{47}$$

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 (12) to obtain

$$\bar{L}_{K+1}^*(\eta_{K+1}) = \sup_{x_{K+1} \in F_{K+1}(\eta_{K+1})} l_{K+1}(x_{K+1}). \tag{48}$$

6.2.2 The probabilistic case

Perfect information We next 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

$$\bar{L}_k^*(x_k) = E\left\{\sum_{i=k}^{K} l_i(x_i, \gamma_i^*(x_i)) + l_{K+1}(x_{K+1})\right\}, \tag{49}$$

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

The principle of optimality [44] states that $\bar{L}_k^*(x_k)$ can be obtained from $\bar{L}_{k+1}^*(x_{k+1})$ by the following recurrence:

$$\bar{L}_k^*(x_k) = \min_{\gamma_k \in \Gamma_k} \left\{ l_k(x_k, u_k) + \int \bar{L}_{k+1}^*(x_{k+1}) p(x_{k+1}|x_k, u_k) dx_{k+1} \right\}. \tag{50}$$

Note that the integral is taken over states that can be reached using (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 $p(x_k|\eta_k)$ on X we have

$$\bar{L}_k^*(\eta_k) = \int \bar{L}_k^*(x_k) p(x_k|\eta_k) dx_k. \tag{51}$$

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

$$\bar{l}(\eta_k, \gamma_k(\eta_k)) = \int l(x_k, \gamma_k(\eta_k)) p(x_k|\eta_k) dx_k. \tag{52}$$

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 (52) is determined from (12) and (10). 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:

$$\bar{L}_k^*(\eta_k) = \inf_{\gamma_k(\eta_k)} \left\{ \bar{l}(\eta_k, \gamma_k(\eta_k)) + \int \bar{L}_{k+1}^*(\eta_{k+1}) p(\eta_{k+1}|\eta_k, \gamma_k(\eta_k)) d\eta_{k+1} \right\}. \tag{53}$$

Above, $p(\eta_{k+1}|\eta_k, \gamma_k(\eta_k))$ is determined by replacing g with γ in (30).

At stage $K+1$, we can use the last term of (12) to obtain

$$\bar{L}_{K+1}^*(\eta_{K+1}) = \int l_{K+1}(x_{K+1}) p(x_{K+1}|\eta_{K+1}) d\eta_{K+1}. \tag{54}$$

6.3 Approximating the State Space

We determine optimal strategies numerically by successively building approximate representations of $\bar{L}_k^*$. This offers flexibility, since analytical solutions are very difficult to obtain. Each dynamic programming iteration can be considered as the construction of an approximate representation of $\bar{L}_k^*$. We decompose the state space into cells of uniform size; however, it is important to note the differences between the use of this decomposition in our context and hierarchical decompositions that are often used in geometric motion planning (see, for example, [47]). Our primary interest in using the decomposition is to construct a good approximation of the continuous function $\bar{L}_k^*$ over the entire state space (or information space under sensing uncertainty). This makes the application of a hierarchical decomposition more difficult.

We obtain the value for $\bar{L}_k^*(x_k)$ by computing the right side of (45) (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 [46].

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 only concerned with strategies that require a finite number of stages before terminating in the goal region, and assume that stationarity holds, as discussed in Secion 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 only retain 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 $x \in X$ though the use of (45) (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. This space, however, is considerably more difficult to utilize than the state space. In one representation, the dimension of the information space is proportional to the number of stages. In another representation, the state space is considered as a function space of pdf's or a space of subsets. In this section, we discuss methods that can be used to simplify the information space. These methods involve tradeoffs between the computational expense and the quality of the information space representation. We are presently experimenting with these alternatives, and 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 [28].

These 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 in the information space, as opposed to the state space.

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:

$$\eta_k = \{u_{k-i+1}, u_{k-i+2}, \ldots, u_{k-1}, y_{k-i}, y_{k-i+1}, \ldots, y_k\}. \tag{55}$$

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.

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* [25, 44]. A sufficient statistic implies that the transformation does not cause any loss of "information." In other words, any decision that is based on the complete history can equivalently be made by only considering the statistic. The identity function is trivially a sufficient statistic. A statistic can be selected that is not sufficient; however, the effects that the projection has on the information space and decisions must be carefully considered.

In general a transformation of the form

$$\eta_k = z_k(u_1, \ldots, u_{k-1}, y_1 \ldots, y_k), \tag{56}$$

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.

Functional approximation with moments One special type of statistic that can be used to approximate the information spaces is the set of *moments*. Recall that the information space can be considered as a function space. For example, with probabilistic uncertainty in sensing, the information space can be represented by the space of possible density functions. We will discuss the moment-based approach with respect to probabilistic uncertainty; however, a similar principle can be applied when using nondeterministic uncertainty. Recall that moments summarize information in random variables. In general, we can use moment computations to approximate this function space. Recall that continuous functions can be approximated by coefficients of a Taylor series expansion. As the dimension of the expansion is increased, the approximations are more accurate. We can obtain a similar situation when considering the information space. The implication of using fixed-order moments is that all decisions must be made without being able to distinguish information states that differ in higher-order moments.

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, $p(x_k|\eta_k)$. Let μ_k and Σ_k represent the mean vector and covariance matrix, respectively, of $p(x_k|\eta_k)$. 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, $p(\eta_{k+1}|\eta_k, \gamma_k(\eta_k))$, will be obtained. This was used in (53) as part of the principle of optimality, that can be used to compute an optimal strategy. When using moments, we can replace $p(\eta_{k+1}|\eta_k, \gamma_k(\eta_k))$ by $p(\mu_{k+1}, \Sigma_{k+1}|\mu_k, \Sigma_k, \gamma_k(\mu_k, \Sigma_k))$ in (53). 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 $\eta_k \in N_k$. Another way the evaluate the moment-based approach is to determine whether

$$\bar{L}(\eta_1, \gamma^*) \approx \bar{L}(\mu_1, \Sigma_1, \gamma^*). \tag{57}$$

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 $\mathcal{C}_{free}$. Let n denote the dimension of the information space (which becomes the dimension of the state space in the case of perfect information). Let $|U|$ denote the number of actions that are considered. Let $|\Theta|$ 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 $O(Q^n|U|\,|\Theta|)$, 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 (45)), has time complexity $O(|U|\ |\Theta|)$, with n fixed.

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 (see [39, 47] for surveys and comparisons). 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 this section. For these strategies, we show forward projections and preimages, which can be compared to the results in Sections 4.3 and 5.3.

For the results in this section, we used the loss functional (14) with $l(u_k, TC_k) = \|v\|\Delta t$ and $C_f = 10000$. We have found the loss functional (13) 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, (13) 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 17.a-b show computed results that were obtained under probabilistic uncertainty with perfect state information. Figure 17.a 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 17.b shows isoperformance classes for every 6 units (i.e., there is a contour for every two expected stages of motion). Figure 18 shows a three-dimensional rendering of the cost-to-go function, along with the preimages. This can be compared to the preimage results from Section 5.3; under the implementation of the optimal strategy, the curves emanate radially from the goal region. In Figure 19 we show the probabilistic forward projection. Under the optimal strategy, all of the probability mass ends up in the goal region. This can be compared to the forward projection for the fixed motion command that was shown in Figure 11, for which some of the probability mass did not reach the goal.

Figures 17.c-d 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 20 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 21 shows a three-dimensional rendering of the cost-to-go function, along with the preimages for the example in Figures 20.d-f. Figure 22 shows the forward projection for the example in Figures 20.g-i.

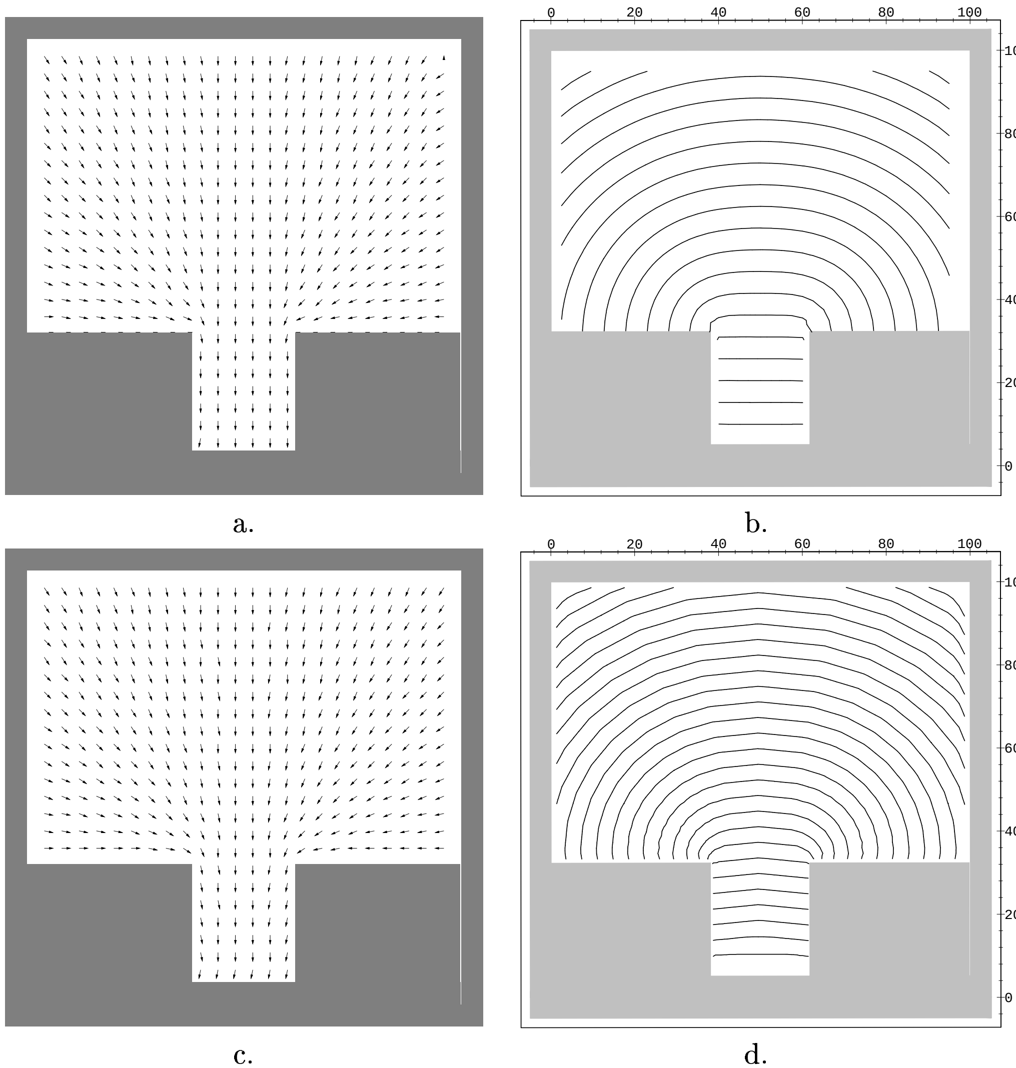


Figure 17: Optimal strategies and performance preimages for the *peg-in-hole* problem under probabilistic control uncertainty and nondeterministic control uncertainty.

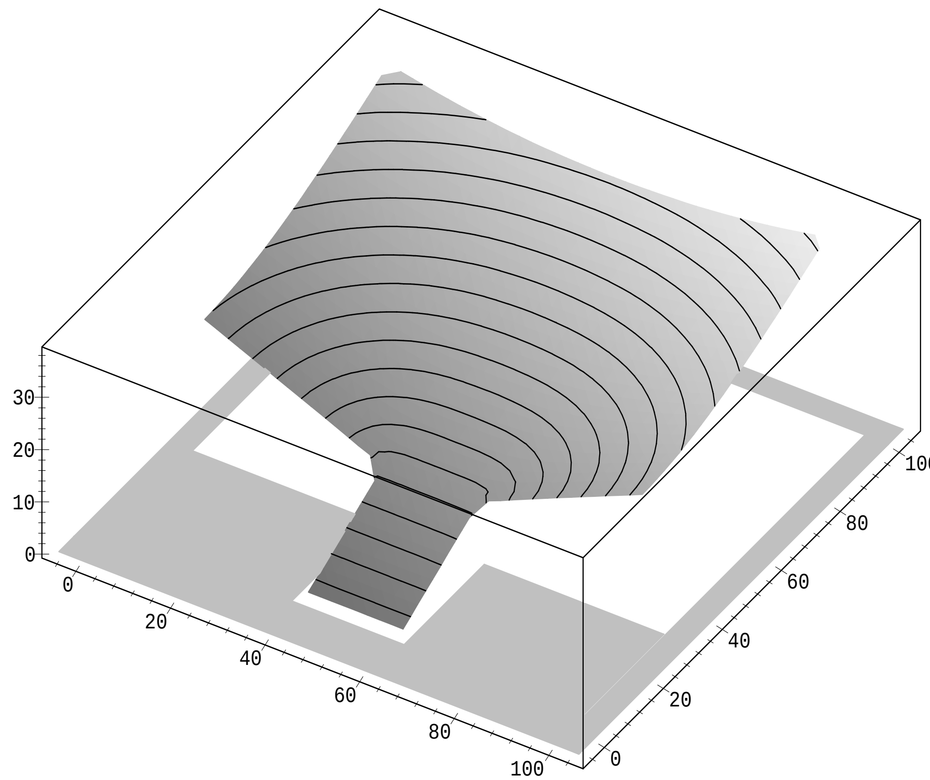


Figure 18: A plot of $\bar{L}^*$ under probabilistic uncertainty and perfect information.

Figure 23 shows 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 23 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 24 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 since 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. Figure 25 shows L^* for the example in Figures 24.a-b.

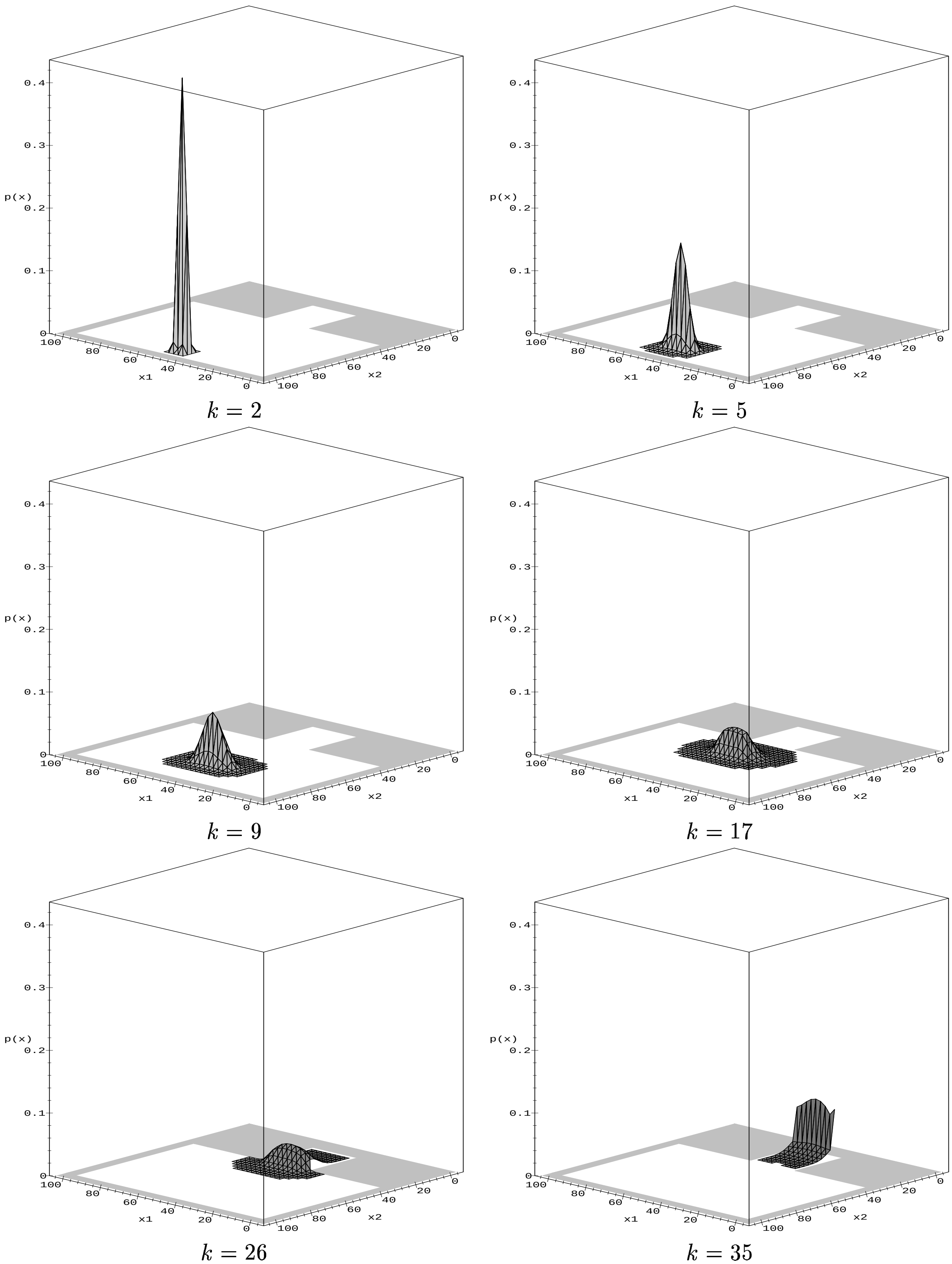


Figure 19: The forward projection for the optimal strategy under perfect information.

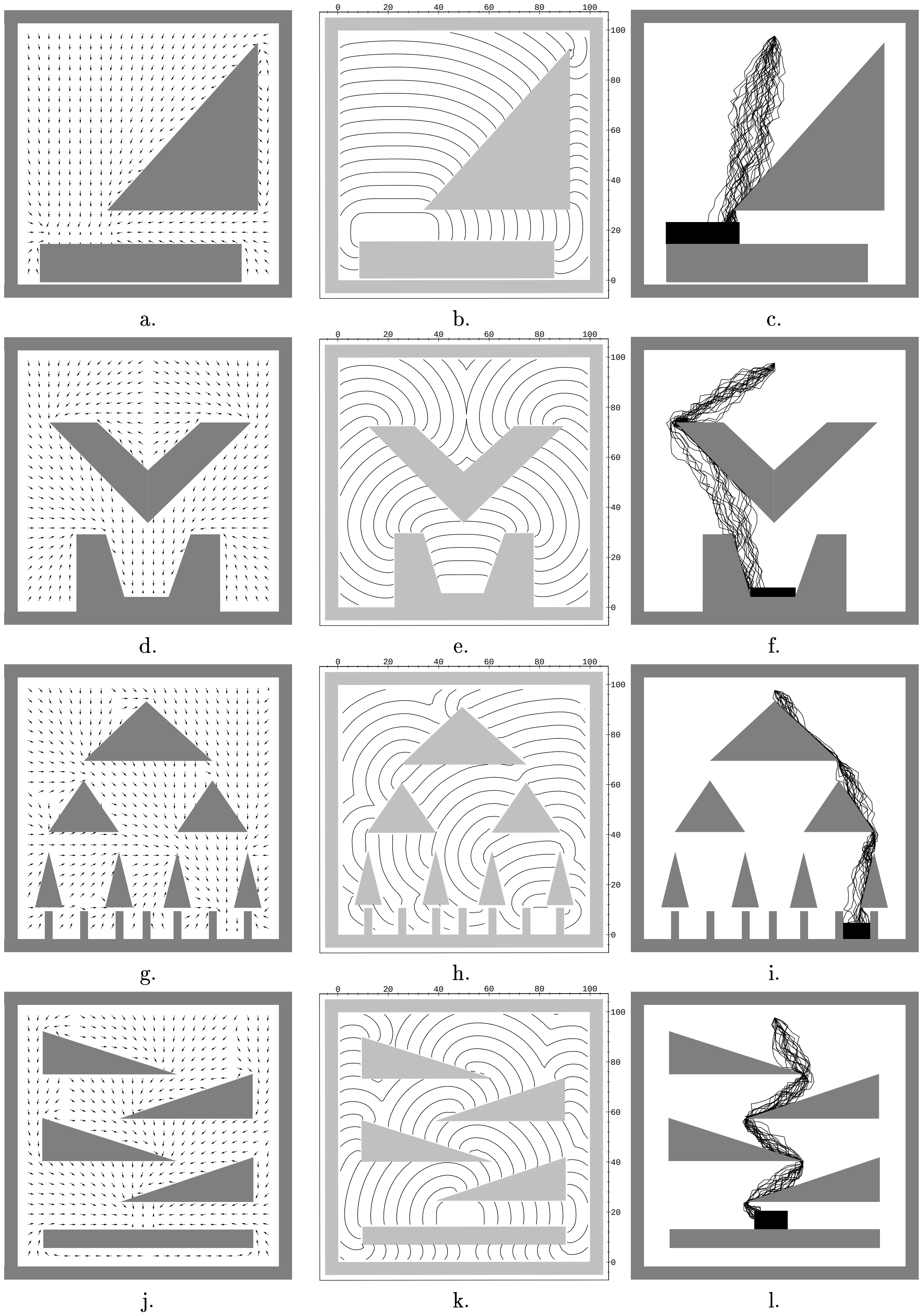


Figure 20: Several examples that were computed under probablistic uncertainty and perfect state information.

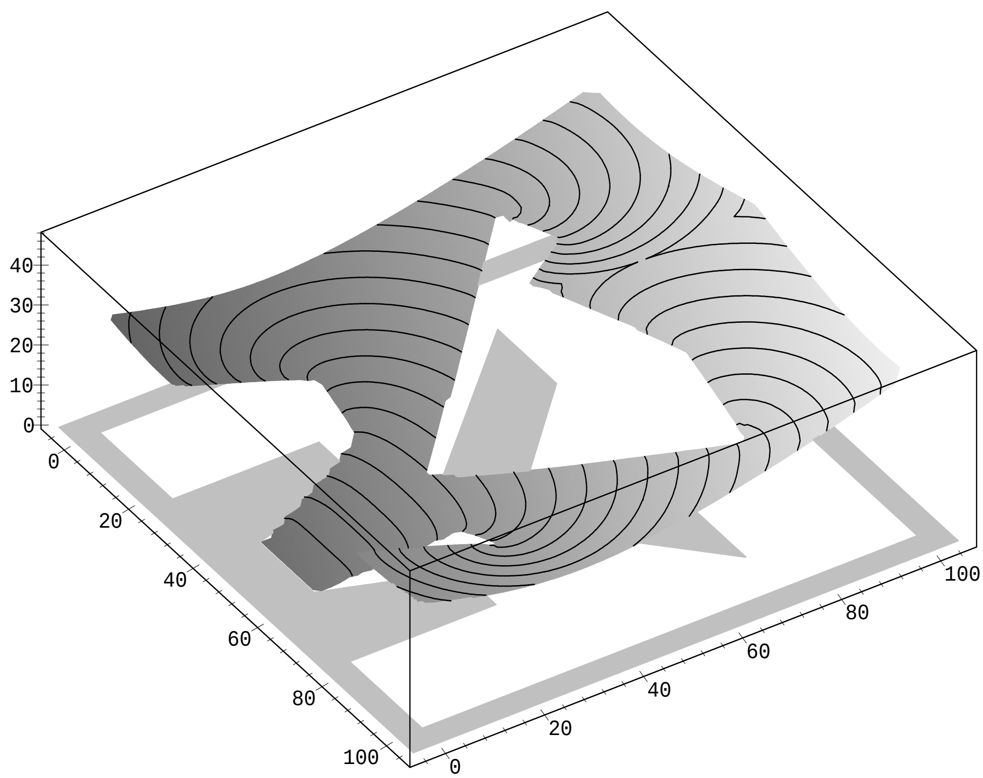


Figure 21: A plot of $\bar{L}^*$ under probabilistic uncertainty.

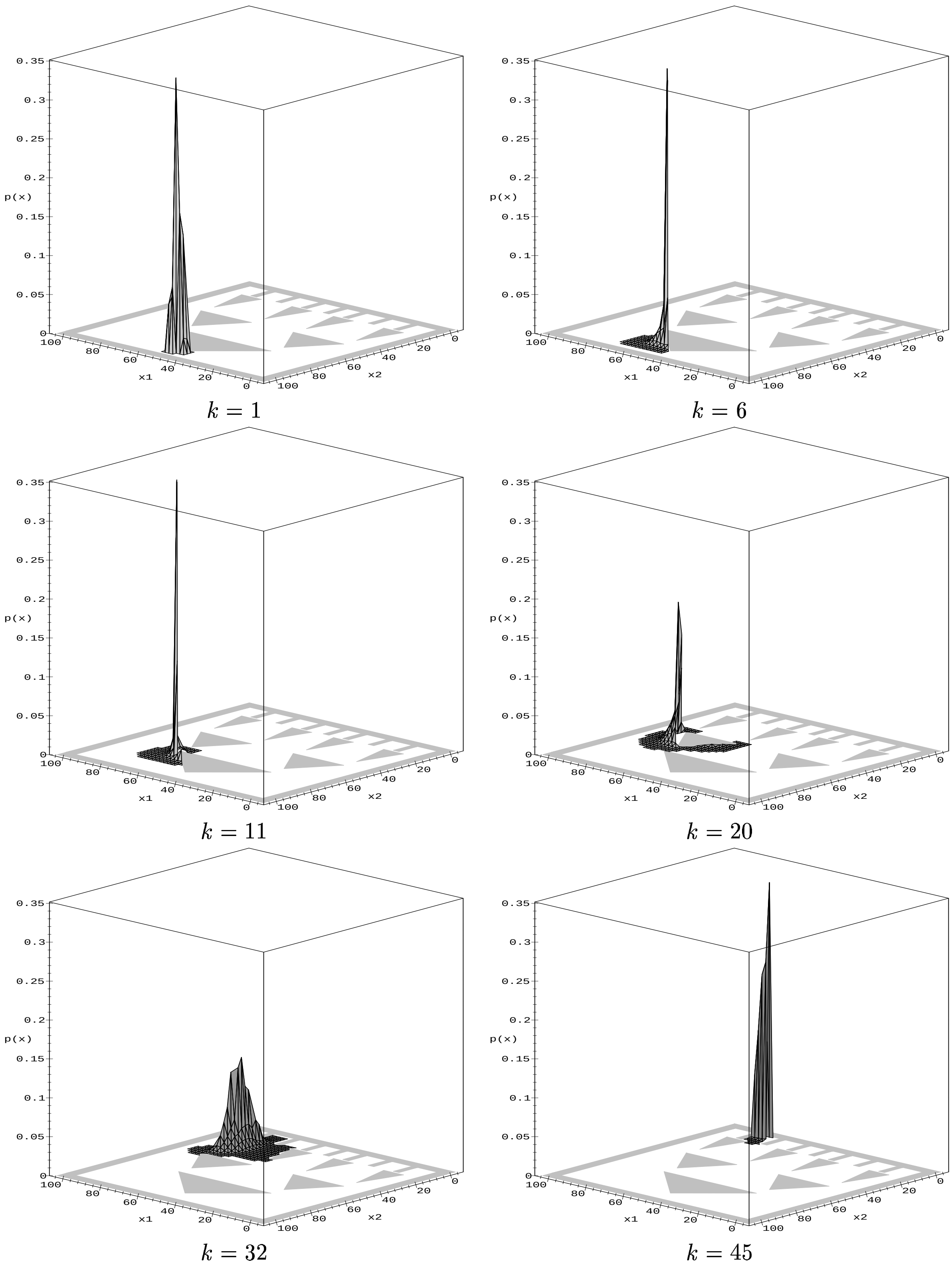


Figure 22: The forward projection for the optimal strategy under imperfect information.

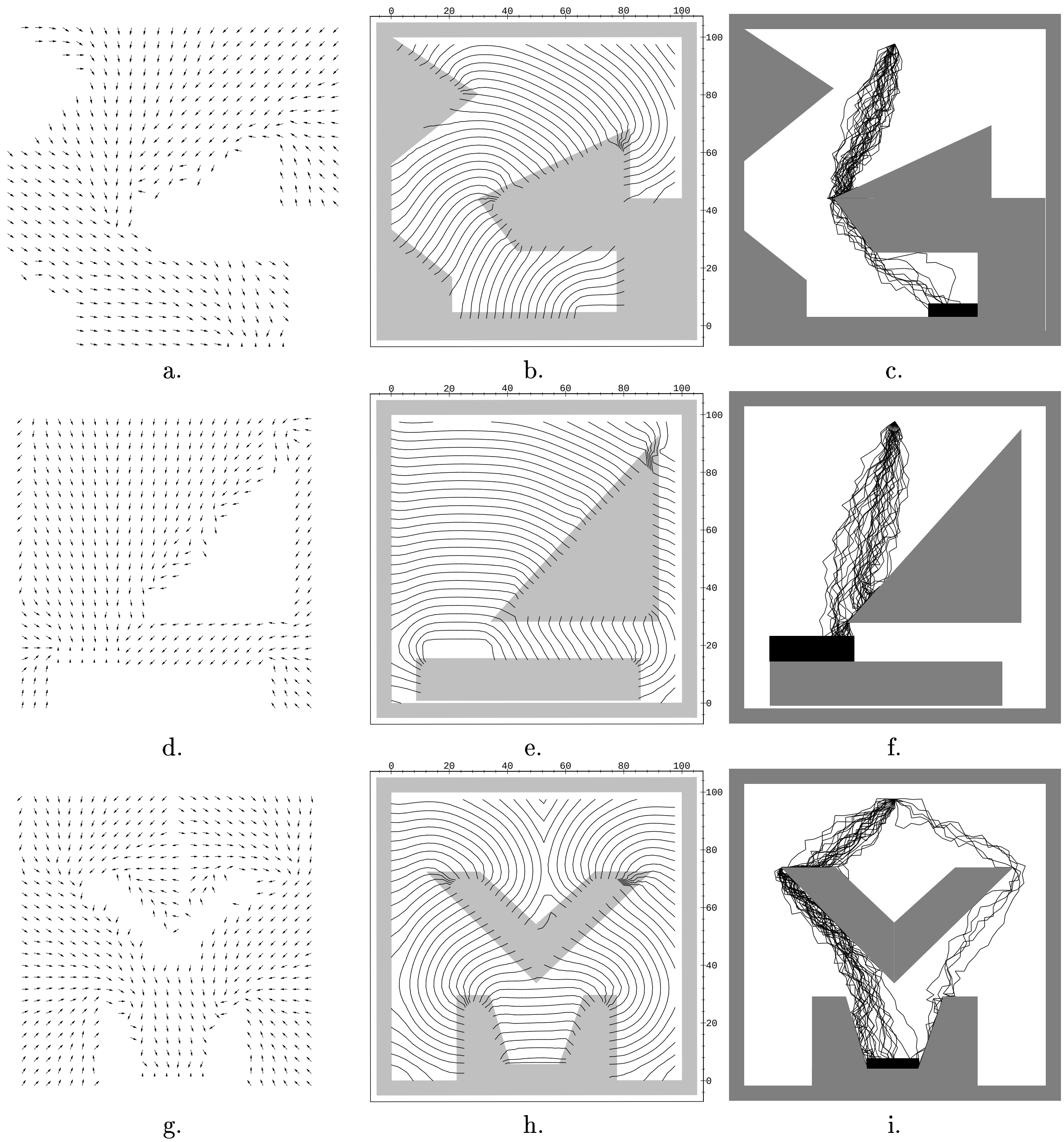
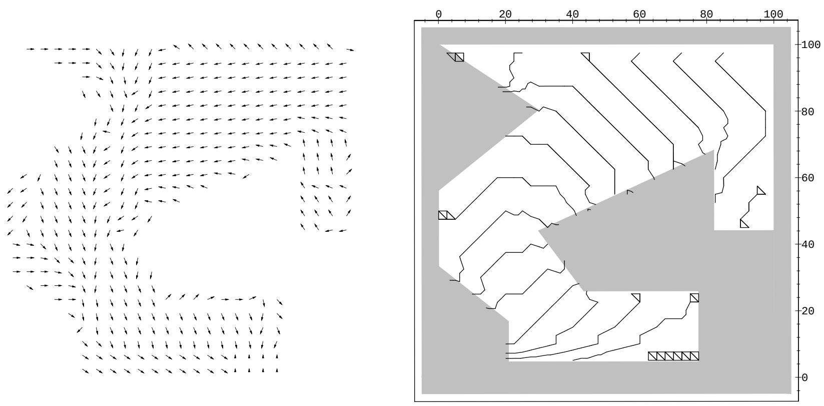
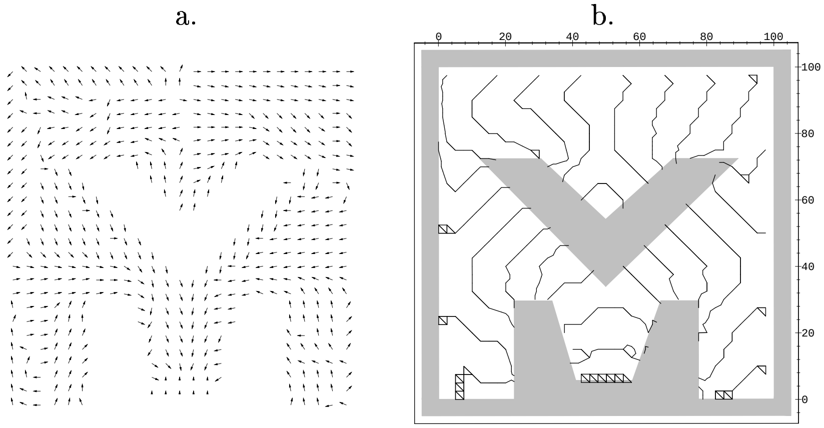


Figure 23: Examples that were computed under probablistic uncertainty and imperfect state information.



a. b.



c. d.

Figure 24: Examples that were computed under nondeterministic uncertainty and imperfect state information.

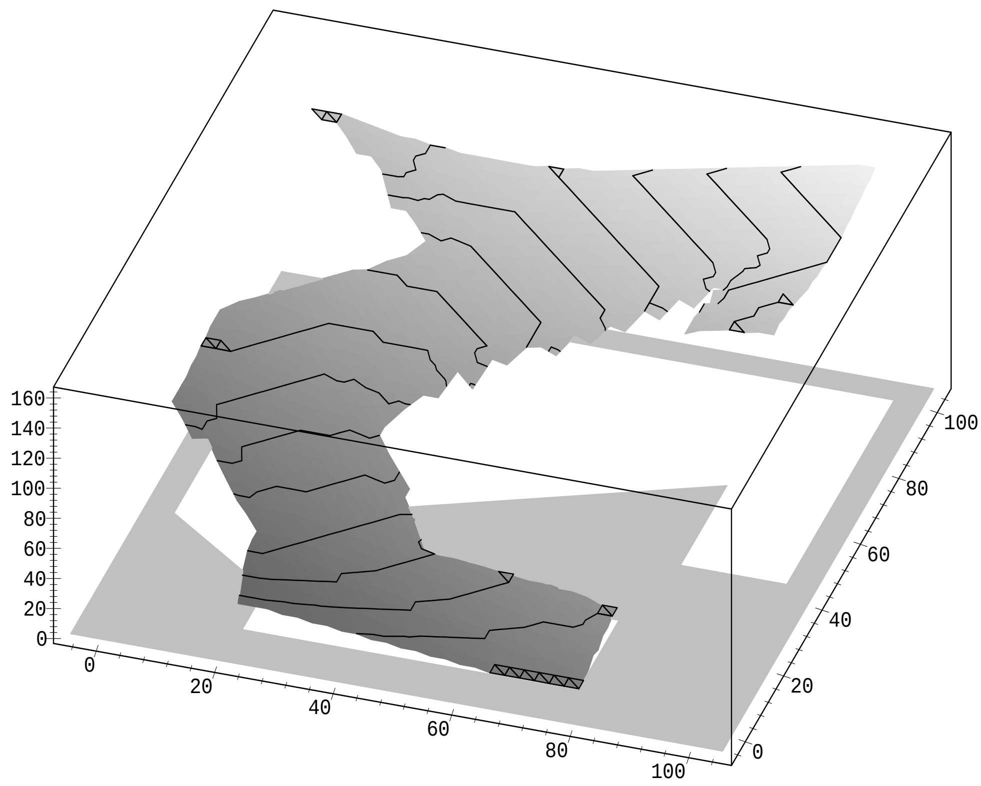


Figure 25: A plot of $\bar{L}^*$ under nondeterministic uncertainty and imperfect information.

7 Discussion

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

Incorporation of model or environment uncertainty In this work we have only considered sensing and control uncertainty; however, in general other forms of uncertainty may exist. For instance, Donald has considered the development of strategies that are robust to errors in the model of an obstacle [22]. By parameterizing the uncertainty in the model, and performing worst-case analysis, a motion plan can be determined that overcomes this form of uncertainty. In [68, 69], uncertainty in the form of a changing environment was analyzed for several specific motion planning problems. These forms of uncertainty can be modeled by augmenting the state space to include information about the environment. Suppose that a set E characterizes different possible environments. For instance, if the true width of a slot is not known for an insertion task, then elements in E could correspond to different possible widths. The state space can be defined subset of $E \times \mathcal{C}$, in which different free configuration spaces are obtained for different values of E. In the same manner that the configuration of a robot is sensed, we can imagine that the environment itself is sensed by the robot. Hence the information space additionally encodes the uncertainty in the current environment. In addition, it is possible that the environment changes over time in a manner that is not predictable by the robot; this is analogous to control uncertainty, but in this case the *environment* must be controlled. Our methodology can be expanded to include these forms of uncertainty by designing motion strategies on the augmented state space or information space. There would, however, be increased computational cost due to a larger state space representation. Issues that pertain to determining optimal motion strategies in an uncertain environment are presented in [52, 51].

Randomized actions In work by Erdmann [27, 28, 31], 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 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 [21, 22, 23]. 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 require to complete the goal, even though robustness is strengthened. In the limiting case, pure Brownian motion can be executed. This essentially makes no modeling assumptions, 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 tradeoff between the distance of a strategy from optimality and the potential incorrectness of the models.

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 [28]. 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 [30]).

Determining accurate uncertainty models The flexibility of our approach permits the use of a variety models for sensing and control uncertainty. In many previous 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 [14], to determine valid error distributions for the computations of appropriate probabilistic backprojections.

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 only changed, however, during the few occurrences of meeting the termination condition. In this paper we have assumed a sampling rates 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. 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 incorporating 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.

Acknowledgements

We gratefully acknowledge Narendra Ahuja, Tamer Başar, 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 #IRI-9110270, a Beckman Institute research assistantship, and a Mavis Fellowship.

References

[1] R. Alami, T. Siméon, and J. P. Laumond. A geometrical approach to planning manipulation tasks. In *Fifth International Symposium of Robotics Research*, pages 113–119, 1989.

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

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

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

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

[6] J. Barraquand and J.-C. Latombe. A Monte-Carlo algorithm for path planning with many degrees of freedom. In *IEEE Int. Conf. on Robotics and Automation*, pages 1712–1717, 1990.

[7] J. Barraquand and J.-C. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. In *IEEE Int. Conf. on Robotics and Automation*, pages 2328–2335, 1991.

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

[9] J. E. Bobrow, S. Dubowsky, and J. S. Gibson. Time-optimal control of robotic manipulators along specified paths. *Int. J. of Robot. Res.*, 4(3):3–17, 1985.

[10] A. Briggs. An efficient algorithm for one-step compliant motion planning with uncertainty. In *5th ACM Symp. Comp. Geom.*, 1989.

[11] R. C. Brost. Automatic grasp planning in the presence of uncertainty. *Int. J. of Robot. Res.*, 7(1):3–17, 1988.

[12] R. C. Brost. *Analysis and Planning of Planar Manipulation Tasks.* PhD thesis, Carnegie Mellon, Pittsburgh, PA, 1991.

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

[14] R. C. Brost and A. D. Christiansen. Probabilistic analysis of manipulation tasks: A computational framework. Technical Report SAND92-2033, Sandia National Laboratories, Albuquerque, NM, January 1994.

[15] A. E. Bryson and Y.-C. Ho. *Applied Optimal Control.* Hemisphere Publishing Corp., New York, NY, 1975.

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

[17] J. F. Canny. On computability of fine motion plans. In *IEEE Int. Conf. on Robotics and Automation*, pages 177–182, 1989.

[18] G. E. Collins. *Lecture Notes in Computer Science.* Springer-Verlag, New York, 1975.

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

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

[21] B. R. Donald. *Error Detection and Recovery for Robot Motion Planning with Uncertainty.* PhD thesis, MIT, Cambridge, MA, 1987.

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

[23] B. R. Donald. Planning multi-step error detection and recovery strategies. *Int. J. of Robot. Res.*, 9(1):3–60, 1990.

[24] B. R. Donald and J. Jennings. Sensor interpretation and task-directed planning using perceptual equivalence classes. In *IEEE Int. Conf. on Robotics and Automation*, pages 190–197, Sacramento, CA, April 1991.

[25] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, NY, 1973.

[26] H. F. Durrant-Whyte. *Integration, Coordination and Control of Multi-Sensor Planning Systems.* Kluwer Academic Publishers, Boston, MA, 1988.

[27] M. Erdmann. Randomization in robot tasks. *Int. J. of Robot. Res.*, 11(5):399–436, October 1992.

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

[29] M. Erdmann. On a representation of friction in configuration space. *Int. J. of Robot. Res.*, 13(3):240–271, 1994.

[30] M. A. Erdmann. On motion planning with uncertainty. Master's thesis, MIT, Cambridge, MA, August 1984.

[31] M. A. Erdmann. *On Probabilistic Strategies for Robot Tasks.* PhD thesis, MIT, Cambridge, MA, 1992.

[32] J. T. Feddema and O. R. Mitchell. Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5(5):691–700, October 1989.

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

[34] S. Geman. Experiments in Bayesian image analysis. *Bayesian Statistics*, 3:159–172, 1988.

[35] K. Y. Goldberg. *Stochastic Plans for Robotic Manipulation.* PhD thesis, Carnegie-Mellon, Pittsburgh, PA, August 1990.

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

[37] G. D. Hager. *Task-Directed Sensor Fusion and Planning.* Kluwer Academic Publishers, Boston, MA, 1990.

[38] H. Hu, M. Brady, and P. Probert. Coping with uncertainty in control and planning for a mobile robot. In *IEEE/RSJ Intern. Workshop on Intelligent Robots and Systems*, pages 1025–1030, Osaka, Japan, November 1991.

[39] Y. K. Hwang and N. Ahuja. Gross motion planning–a survey. *ACM Computing Surveys*, 24(3):219–291, September 1992.

[40] K. Ikeuchi and T. Kanade. Modeling sensors: Toward automatic generation of object recognition program. *Comp. Vision, Graphics, and Image Process.*, 48:50–79, 1989.

[41] M. E. Kahn and B. E. Roth. The near minimum-time control of open-loop articulated kinematic chains. *Trans. ASME J. Dyn. Sys., Meas., Contr.*, 93(3):164–172, 1971.

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

[43] N. V. Krylov. *Controlled diffusion processes.* Springer-Verlag, New York, NY, 1980.

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

[45] R. E. Larson. *State Increment Dynamic Prgramming.* Elsevier, New York, NY, 1968.

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

[47] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991.

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

[49] S. M. LaValle and S. A. Hutchinson. An objective-based stochastic framework for manipulation planning. In *Proc. IEEE/RSJ/GI Int'l Conf. on Intelligent Robots and Systems*, pages 1772–1779, September 1994.

[50] S. M. LaValle and S. A. Hutchinson. A framework for constructing probability distributions on the space of segmentations. *Computer Vision and Image Understanding*, 61(2), March 1995.

[51] S. M. LaValle and R. Sharma. Motion planning in stochastic environments: Applications and computational issues. In *IEEE Int'l Conf. on Robotics and Automation*, 1995. To appear.

[52] S. M. LaValle and R. Sharma. Motion planning in stochastic environments: Theory and modeling issues. In *IEEE Int'l Conf. on Robotics and Automation*, 1995. To appear.

[53] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor. Automatic systhesis of fine-motion strategies for robots. *Int. J. of Robot. Res.*, 3(1):3–24, 1984.

[54] K. M. Lynch and M. T. Mason. Pulling by pushing, slip with infinite friction, and perfectly rough surfaces. *Int. J. of Robot. Res.*, 14(2):174–183, 1995.

[55] N. Mahadevamurty, T-C. Tsao, and S. Hutchinson. Multi-rate analysis and design of visual feedback digital servo control systems. *ASME Journal of Dynamic Systems, Measurement and Control*, pages 45–55, March 1994.

[56] M. T. Mason. Compliance and force control for computer controlled manipulators. In B. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Perez, and M. T. Mason, editors, *Robot Motion: Planning and Control*, pages 373–404. MIT Press, Cambridge, Mass., 1982.

[57] M. T. Mason. Automatic planning of fine motions: Correctness and completeness. In *Proc. IEEE Int'l Conference on Robotics*, pages 492–503, 1984.

[58] M. T. Mason. Mechanics and planning of manipulator pushing operations. *Int. J. of Robot. Res.*, 5(3):53–71, 1986.

[59] J. Miura and Y. Shirai. Planning of vision and motion for a mobile robot using a probabilistic model of uncertainty. In *IEEE/RSJ Intern. Workshop on Intelligent Robots and Systems*, pages 403–408, Osaka, Japan, May 1991.

[60] B. K. Natarajan. The complexity of fine motion planning. *Int. J. of Robot. Res.*, 7(2):36–42, 1988.

[61] B. J. Oommen, N. Andrade, and S. S. Iyengar. Trajectory planning of robot manipulators in noisy work spaces using stochastic automata. *Int. J. of Robot. Res.*, 10(2):135–148, April 1991.

[62] C. H. Papadimitriou. Games against nature. *Journal of Computer and System Sciences*, 31:288–301, 1985.

[63] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, August 1987.

[64] R. P. Paul and B. Shimano. Compliance and control. In *Proc. of the Joint American Automatic Control Conference*, pages 1694–1699, 1976.

[65] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems*, 102, 1981.

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

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

[68] R. Sharma. Locally efficient path planning in an uncertain, dynamic environment using a probabilistic model. *IEEE Trans. on Robotics and Automation*, 8(1):105–110, February 1992.

[69] R. Sharma. A probabilistic framework for dynamic motion planning in known environments. In *IEEE Int. Conf. on Robotics and Automation*, pages 2459–2464, Nice, France, May 1992.

[70] K. G. Shin and N. D. McKay. Minimum-time control of robot manipulators with geometric path constraints. *IEEE Trans. Automatic Control*, 30(6):531–541, 1985.

[71] J. F. Silverman and D. B. Cooper. Bayesian clustering for unsupervised estimation of surface and texture models. *IEEE Trans. Pattern Anal. Machine Intell.*, 10(4):482–496, July 1988.

[72] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *Int. J. of Robot. Res.*, 5(4):56–68, 1986.

[73] R. C. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Fourth International Symposium on Robotics Research*, pages 421–429, 1987.

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

[75] H. Takeda, C. Facchinetti, and J.-C. Latombe. Planning the motions of a mobile robot in a sensory uncertainty field. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(10):1002–1017, October 1994.

[76] R. H. Taylor, M. T. Mason, and K. Y. Goldberg. Sensor-based manipulation planning as a game with nature. In *Fourth International Symposium on Robotics Research*, pages 421–429, 1987.

[77] J. C. Trinkle and D. C. Zeng. Prediction of the quasistatic planar motion of a contacted rigid body. *IEEE Trans. on Robotics and Automation*, 11(2):229–246, April 1995.

[78] D. Whitney. Force feedback control of manipulator fine motions. *Trans. ASME J. of Dyn. Syst., Meas., Contr.*, 99:91–97, 1977.

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

[80] P. Williams. *Probability with Martingales.* Cambridge Press, New York, NY, 1991.