A Framework for Motion Planning in Stochastic Environments: Modeling and Analysis

Steven M. LaValle   Rajeev Sharma
lavalle@cs.uiuc.edu   *rajeev@cs.uiuc.edu*
The Beckman Institute
University of Illinois, Urbana, IL 61801

Abstract

We present a framework for analyzing and determining motion plans for a robot that operates in an environment that changes over time in an uncertain manner. We first classify sources of uncertainty in motion planning into four categories, and argue that the framework addressed in this paper characterizes an important, yet little-explored category. We treat the changing environment in a flexible manner by combining traditional configuration space concepts with a Markov process that models the environment. For this context, we then propose the use of a motion strategy, which provides a motion command for the robot for each contingency that it could be confronted with. We allow the specification of a desired performance criterion, such as time or distance, and the goal is to determine a motion strategy that is optimal with respect to that criterion. A motion planning problem in this framework is formulated as the design of a stochastic optimal controller. Applications and computational issues are discussed in a companion paper [12].

1 Introduction

The ability of a robot to autonomously plan and execute motions under uncertainties greatly improves its range of operation. Substantial interest in the field of robot motion planning has led to a variety of approaches that use different models of the robot and its environment [9]. The success of a motion planning approach depends to a large extent on the manner in which various forms of uncertainty are modeled and treated.

There are two popular representations of uncertainty that have been applied to geometric motion planning problems. One representation restricts parameter uncertainties to lie within a specified set. A motion plan is then generated that is based on *worst-case analysis* (e.g., [10, 13]). We refer to this representation as *bounded uncertainty*. The other popular representation expresses uncertainty in the form of a posterior probability density. This often leads to the construction of motion plans through *average-case analysis* (e.g., [2, 14])

Uncertainty can be introduced into a motion planning problem in a number of ways. We organize this uncertainty into four basic sources for discussion:

- *Uncertainty in robot sensing* **(RS)**
- *Uncertainty in robot predictability* **(RP)**
- *Uncertainty in environment sensing* **(ES)**
- *Uncertainty in environment predictability* **(EP)**

We will now describe each of the sources of uncertainty, and the final source will be the primary focus of this paper. For the discussion, we will consider each type of uncertainty in isolation, although in general any combination of these types can be considered simultaneously in a motion planning formulation.

Type RS uncertainty. Suppose that the space of collision-free configurations, $\mathcal{C}_{free}$, is known by the robot (a space of valid configurations, $\mathcal{C}_{valid}$, which includes contact with obstacles, could alternatively be defined [9]). At a given point in time, the robot position is assumed to be characterized by a point, $\mathbf{q}$ in $\mathcal{C}_{free}$. Under Type RS uncertainty, incomplete or imperfect information is utilized by the robot to make an inference about its configuration. With a bounded uncertainty model, the robot might have sufficient information to infer that for some subset $Q \subset \mathcal{C}_{free}$, its configuration, $\mathbf{q}$ must lie in Q. In [10, 13] this representation of uncertainty is used to guarantee that the robot recognizably terminates in a goal region. With a probabilistic model, the robot might infer a posterior probability density over configurations, $p(\mathbf{q})$, that is conditioned on sensor observations, initial conditions, or additional knowledge (e.g., [3, 6]).

Type RP uncertainty. Suppose again that the space of collision-free configurations, $\mathcal{C}_{free}$, is known by the robot; however, in addition, the robot knows its current configuration $\mathbf{q} \in \mathcal{C}_{free}$. Motion commands can be given to the robot, but with Type RP uncertainty the future configurations cannot, in general, be completely predicted. With bounded uncertainty, the robot may infer that some future configuration will belong to a subset $Q \subset \mathcal{C}_{free}$. The method of *preimage backchaining* constitutes of large body of work in which bounded uncertainties are propagated and combined with Type RS uncertainty, to guarantee that the robot will achieve a goal (e.g., [10, 13]). With a probabilistic model, future configurations can be described by a posterior density over configurations, $p(\mathbf{q})$, that is conditioned on the initial configuration and the executed motion command (e.g., [2]).

We describe Types ES and EP by making a direct analogy to Types RS and RP (see Figure 1). While Type RS models sensing uncertainty in a space of *configurations*, we will consider Type ES to model sensing uncertainty in a space of *environments*. As Type RP models uncertainty in prediction in a space of *configurations*, we will consider EP to model uncertainty in a space of *environments*.

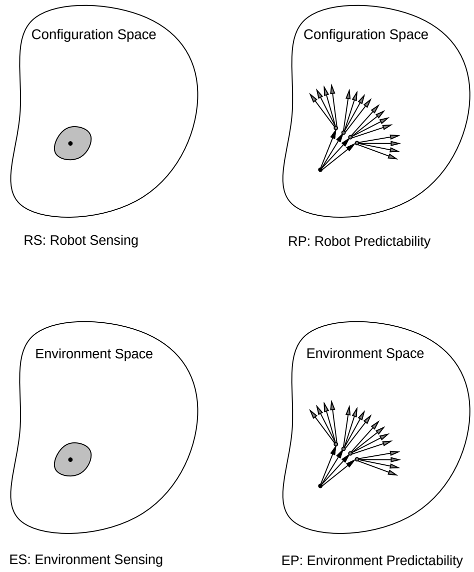

Figure 1. Four sources of uncertainty in the motion planning problem.

Although a space of configurations is a well-defined concept in robotics literature, we must define what is meant by a "space of environments." The robot's environment could be expressed in different ways. One could consider the environment as the representation of a workspace with obstacles. For the purpose of discussion, we consider $\mathcal{C}_{free}$ to be a representation of the environment. If we begin with $\mathcal{C}_{free}$ and add another obstacle to the workspace, a new environment, $\mathcal{C}'_{free}$ will be obtained.

Consider a $\mathcal{E}$ to be a set of possible environments. In such a case, we would have, for instance, $\mathcal{C}_{free} \in \mathcal{E}$ and $\mathcal{C}'_{free} \in \mathcal{E}$. For a given n-dimensional $\mathcal{C}$-space, $\mathcal{E}$ might be defined as the power set of $\mathcal{C}$. This would characterize every conceivable $\mathcal{C}_{free}$ that could be obtained as a subset of $\mathcal{C}$. Besides a configuration space, the environment could represent additional information relevant for motion planning. For instance the environment could become hazardous, as considered in [14]. Typically, there are many restrictions on the types of environments that can occur, and we will primarily consider $\mathcal{E}$ conceptually as a reference set for discussion.

Type ES uncertainty. By analogy to Type RS, suppose that a space of possible environments, $\mathcal{E}$, is known to the robot. At a given point in time, the particular environment that the robot belongs to can be considered as a point in $e \in \mathcal{E}$, which represents some $\mathcal{C}_{free}$. Under Type ES uncertainty, incomplete or imperfect information is utilized by the robot to make an inference about its environment. If more information is available, it might be the case that the possibilities for $\mathcal{E}$ are restricted to some finite set. For instance, the only uncertain aspect about the environment might be the existence of a few obstacles at some fixed locations. With a bounded uncertainty model, the robot might have sufficient information to infer that for some subset $F \subset \mathcal{E}$, the environment, e must lie in F. With a probabilistic model, the robot might infer a posterior probability density over environments, $p(e)$, that is conditioned on sensor observations, initial conditions, or additional knowledge (e.g., [5, 15]).

Type EP Uncertainty. Suppose again that the space of environments, $\mathcal{E}$, is known by the robot; however, in addition, the robot knows its current environment $e \in \mathcal{E}$. Predictable motion commands might be given to the robot, but with Type EP uncertainty future environments cannot be completely predicted. With bounded uncertainty, the robot may infer that some future environment will belong to a subset $F \subset \mathcal{E}$. With a probabilistic model, future environments can be described by a posterior density over environments, $p(e)$, that can be conditioned on the initial environment, the robot configuration, or an executed motion command.

In this paper we are concerned with systematically handling Type EP uncertainty, when the environment is partially predictable. The basic framework that we present models problems in which $\mathcal{E}$ is restricted to a finite number of environments. This covers a wide variety of motion planning problems, and as shown in the companion paper [12], leads to computationally feasible solutions. As an illustrative example, consider the problem in Figure 2, in which a robot must reach a goal region in minimal time, while the configuration space may unpredictably change during execution. In particular, the doors may open or close randomly, changing the feasible paths to the goal.

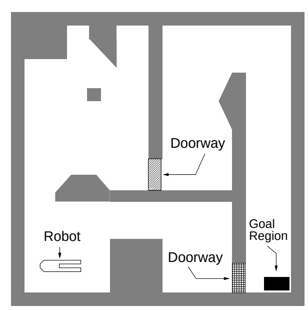

Figure 2. A changing environment in which the workspace changes over time, by (a) the opening and closing of "doors."

We provide general formulations of the robot, its motion model, a goal region, and the environment. For geometric aspects of the problem we use general configuration space concepts that characterize basic motion planning problems [9]. The environment is modeled as a Markov process, which is powerful enough to encode many important stochastic processes, such as a Wiener process (Brownian motion) or a Poisson process. By combining these concepts into a composite state space, we can model a variety of motion planning problems.

We now discuss the importance of determining optimal motion *strategies*. In classical motion planning approaches, the output of algorithm is usually a "motion plan" for a given description of the $\mathcal{C}$, the initial and the goal positions. When unpredictable changes occur in the workspace, *dynamic replanning* is often used. This has been used, for example, in the context of error-detection and recovery [4], and task-level reasoning [7]. Alternatively, a fixed command might be given to the robot, and local collision avoidance is performed to handle unexpected aspects of the environment [1, 16]. In the framework that we propose, a "motion strategy" provides a motion command for the robot for each contingency that it could be confronted with. This motion strategy can be considered as a state-feedback stochastic controller [8], on a state space that simultaneously considers the environment and the robot configuration. Replanning is not needed when the environment changes, because the robot responds appropriately for different regions of the state space during execution. In addition, a state-feedback controller is advantageous, since it will typically be robust to small modeling errors that can develop during execution. To select a motion strategy, we formulate an explicit performance criterion (or loss functional) that evaluates a trajectory executed by the robot. This allows a variety items, such as time or distance, to be optimized through the selection of a strategy.

2 Mathematical Formulation

In this section we develop the mathematical concepts that model motion planning problems with Type EP uncertainty. Section 2.1 introduces the finite-state Markov process that is used to model the changing environment, and the relationship of this model to the configuration space of the robot. In Section 2.2 we define a model of robot motion, which accepts a motion command and produces a next configuration. Section 2.3 introduces the concept of dynamic regions in the robot's configuration space. These regions are used to explicitly define interaction that occurs with the environment and the robot, which is effected through the definition of a performance criterion.

2.1 The Environment Process

For geometric motion planning problems without uncertainty, the space of possible situations that can occur is sufficiently characterized by $\mathcal{C}_{free}$ (or $\mathcal{C}_{valid}$ [9]). In our context, the environment can additionally interfere with the motion plans of $\mathcal{A}$, compelling us to define a finite set, E, of *environment modes*.

Since we are modeling problems in which the environment changes, we require an explicit representation of time in many subsequent definitions. We define a discretized representation of time by a set of *stages*, with an index $k \in \{1, 2, \ldots, K\}$. Stage k refers to time $(k-1)\Delta t$. At a given stage, k, the environment is in some mode $e_k \in E$, which is known to the robot. We generally take Δt sufficiently small to approximate continuous trajectories.

We additionally consider the environment as a finite-state Markov process, which we call the *environment process*. As another example of using a Markov model in the analysis of motion planning, see [16]. At the initial stage ($k = 1$) the environment mode, $e_1 \in E$, is given. For a given environment mode, e_k, the next environment mode, e_{k+1}, is specified with a probability distribution over E. This probability distribution is defined by a vector P_i such that $P_i[j] = P(e_{k+1} = j | e_k = i)$.

We now present an example of a four-mode environment process that can model the problem from Figure 2.a. We define the following four environment modes: $e = 0$ if Doors 1 and 2 are open, $e = 1$ if 1 is closed and 2 is open, $e = 2$ if 1 is open and 2 is closed, and $e = 3$ if both are closed. Suppose each door is modeled with Poisson processes. Let λ denote a Poisson arrival rate. The density for the time of the first arrival is $p(t_a) = \lambda e^{-\lambda t_a}$. We denote the arrival rate of a door closing, given that it is open, as λ_c, and the arrival rate of a door opening, given that it is closed as λ_o.

Assume for this example that the two doors are governed by independent, identical Poisson processes. The probability that a closed door will open in time Δt is

$$P_{01} = \int_0^{\Delta t} \lambda_o e^{-\lambda_o t_a} dt_a = 1 - e^{-\lambda_o \Delta t}. \qquad (1)$$

The probability that it will stay closed is $P_{11} = 1 - P_{01}$. For a door that is initially open, we similarly obtain $P_{10} = 1 - e^{-\lambda_c \Delta t}$, and $P_{00} = 1 - P_{10}$.

The environment transition probabilities can be generated by taking products of pairs of P_{00}, P_{01}, P_{10}, and P_{11}:

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} P_{00}^2 & P_{00}P_{10} & P_{10}P_{00} & P_{10}^2 \\ P_{00}P_{01} & P_{00}P_{11} & P_{10}P_{01} & P_{10}P_{11} \\ P_{01}P_{00} & P_{01}P_{10} & P_{11}P_{00} & P_{11}P_{10} \\ P_{01}^2 & P_{01}P_{11} & P_{11}P_{01} & P_{11}^2 \end{bmatrix}. \qquad (2)$$

The four-mode process is depicted in Figure 3, in which we take $\lambda_o = \lambda_c = 0.10101354$ (approximately one expected arrival every ten seconds), and $\Delta t = 0.2$. This results in $P_{10} = P_{01} = 0.02$ and $P_{00} = P_{11} = 0.98$.

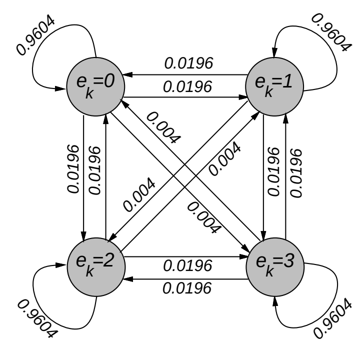

Figure 3. The environment process can be considered as a finite-state Markov process with state transition probabilities.

For this example, the environment process is independent of the robot configuration. In general, however, we allow the robot to have influence over the environment by conditioning the probabilities on the configuration of the robot. The servicing problem, discussed in [12], is an example in which this extension is needed.

In general, to uniquely identify all of the possible situations that can occur in our problem, we define a *state* *space* as the cartesian product, $X = \mathcal{C}_{free} \times E$. This is similar to the view taken in [4], in which the space for motion planning is a cartesian product of $\mathcal{C}_{free}$ with a single parameter that characterizes a hole width for a peg-in-hole task. The state at stage k is denoted by x_k, which simultaneously represents both a configuration of $\mathcal{A}$ in the geometric sense, and an environment mode, e_k. The environment modes form a partition of the state space, X. Each time the environment mode changes, the robot is forced into a different layer of X.

2.2 Defining the Robot Behavior

In this section we present several concepts that lead to the definition of a strategy, which characterizes a fixed behavior for the robot. We begin by defining an *action*, u_k, (or command), which can be issued to $\mathcal{A}$ at each stage, k. We let U denote the *action space* for $\mathcal{A}$, while requiring that $u_k \in U$. We define a *state transition distribution* as $P(x_{k+1}|x_k, u_k)$. This represents a probability distribution over a finite set of next states, given x_k as the initial state, and an action u_k.

As an example, we present a state transition distribution that applies to the case in which $\mathcal{C} \subseteq \Re^2$, and the robot is limited to translational motion. More complicated motions are considered in the companion paper. Dynamic robot constraints could also be introduced; however, velocities would have to be represented in the state space, and the robot constraints would have to be specified in state-space form. We define the action space as $U = [0, 2\pi) \cup \{\emptyset\}$. If $u_k \in [0, 2\pi)$, then $\mathcal{A}$ attempts to move a distance $\|v\|\Delta t$ toward a direction in $\mathcal{C}$, in which $\|v\|$ denotes some fixed speed for $\mathcal{A}$. If $u_k = \emptyset$, then the robot remains motionless.

Consider the case in which $x_k \in \mathcal{C}_{free}$ is at a distance of at least $\|v\|\Delta t$ from the obstacles. If $\mathcal{A}$ chooses action $u_k \neq \emptyset$ from state x_k (We use the notation $x_k[i]$ to refer to the i^{th} element of the vector x_k.), then

$$x_{k+1} = \left[\begin{array}{c} x_k[1] + \|v\|\Delta t\; cos(u_k) \\ x_k[2] + \|v\|\Delta t\; sin(u_k) \\ e_{k+1} \end{array} \right]. \qquad (3)$$

in which the environment mode e_{k+1} is known to be sampled from $P(e_{k+1}|x_k, u_k)$. We can thus consider a finite-valued random variable X_{k+1} with corresponding distribution $P(x_{k+1}|x_k, u_k)$, which can be inferred from the given model. If $u_k = \emptyset$, then $x_k[1] = x_{k+1}[1]$ and $x_k[2] = x_{k+1}[2]$; however, e_{k+1} is not necessarily equal to e_k because the environment transition equation determined e_{k+1}.

We now define the notion of a robot strategy for our context. At first it might seem appropriate to define some action u_k for each stage; however, we want a motion plan that is prepared for the various contingencies presented by the changing environment. Therefore, we define a *strategy at stage* k of $\mathcal{A}$ as a function $\gamma_k : X \rightarrow U$. For each state, x_k, the function γ_k yields an action $u_k = \gamma_k(x_k)$. The set of mappings $\{\gamma_1, \gamma_2, \ldots, \gamma_K\}$ is denoted by γ and termed a *strategy*. For most motion planning problems, the solution strategy, γ_k, will be the same for all k (i.e., each robot action depends only on the current state, and not the particular stage). Section 4 presents a discussion of time varying strategies, in which this assumption is relaxed.

We represent a desired performance criterion by a real-valued functional $L(x_1, ..., x_{K+1}, u_1, ..., u_K)$, called the *loss functional*. A strategy that produces a lower loss will be considered preferable. The ultimate goal of a planner is to determine an optimal strategy $\gamma^* = \{\gamma_1^*, \gamma_2^*, \ldots, \gamma_K^*\}$ that causes L to be minimized in an expected sense.

2.3 Defining Performance

Sections 2.1 and 2.2 have introduced the environment process and a model of the robot behavior. This section discusses the key concepts that are used to model the effect that the environment has on the robot. In particular, costs that appear in a loss functional directly depend on dynamic regions in the state space. If the robot enters a particular dynamic region, the amount of loss received might increase or decrease. For instance, a dynamic region might correspond to the robot's collision with a closed door, which would incur a very high loss.

We will define dynamic regions in the workspace, $\mathcal{W}$, and subsequently discuss how these regions are mapped into the state space, X. In addition to static obstacles, let $\mathcal{W}$ contain a set of m *dynamic regions*, denoted by $\{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$. Each dynamic region is a subset of $\mathcal{W}$, and pairs of dynamic regions are not necessarily disjoint.

A dynamic region $\mathcal{D}_i$ in $\mathcal{W}$ can map into the region $\mathcal{CD}_i^c \subset \mathcal{C}_{free}$, which is given by:

$$\mathcal{CD}_i^c = \{\mathbf{q} \in \mathcal{C}_{free} |\; \mathcal{A}(\mathbf{q}) \cap \mathcal{D}_i \neq \emptyset\}. \qquad (4)$$

We call $\mathcal{CD}_i^c$ a *contact (dynamic)* $\mathcal{C}$*-region*. This yields configurations in which the robot is in contact with $\mathcal{D}_i$. A contact $\mathcal{C}$-region is useful for problems such as that in Figure 2.a, in which contact with a door must be determined.

In some situations, we will be interested in determining whether the robot is completely contained in some $\mathcal{D}_i$. A dynamic region $\mathcal{D}_i$ in $\mathcal{W}$ maps into the region $\mathcal{CD}_i^e \subset \mathcal{C}_{free}$, which is given by:

$$\mathcal{CD}_i^e = \{\mathbf{q} \in \mathcal{C}_{free} |\; \mathcal{A}(\mathbf{q}) \subseteq \mathcal{D}_i\}. \qquad (5)$$

We call $\mathcal{CD}_i^e$ an *enclosure (dynamic)* $\mathcal{C}$*-region*. One could alternatively define $\mathcal{CD}_i^e$ as the subset of configuration space in which the robot and dynamic region interiors overlap, and also the containment in the definition of $\mathcal{CD}_i^e$ could be made strict. Note that $\mathcal{CD}^e \subseteq \mathcal{CD}^c$.

We now want to map the dynamic regions into the state space, since the loss functional depends on the state trajectory. Since the dynamic regions have been mapped into $\mathcal{C}_{free}$, the mapping into X can be considered as lifting the contact $\mathcal{C}$-region (or enclosure $\mathcal{C}$-region) into different layers of X. We want the dynamic region to only influence the robot at certain layers. For instance, with the example in Figure 2.a, we only want the dynamic region to exist in X in environment modes that correspond to the door being closed. In other modes, the door should not interfere with the robot. For each $i \in \{1, \ldots, m\}$ we select a subset, E_i, of environment states, $E_i \subseteq E$.

We can represent a state $x \in X$ by $(\mathbf{q}, e)$, in which $\mathbf{q} \in \mathcal{C}_{free}$ and $e \in E$. If $\mathcal{D}_i$ is a contact dynamic region, then we define

$$X_i = \{(\mathbf{q}, e) \in X |\ \mathbf{q} \in \mathcal{CD}_i^c \text{ and } e \in E_i\}. \qquad (6)$$

Alternatively, if $\mathcal{D}_i$ is an enclosure dynamic region, then we define

$$X_i = \{(\mathbf{q}, e) \in X |\ \mathbf{q} \in \mathcal{CD}_i^e \text{ and } e \in E_i\}. \qquad (7)$$

Each X_i may be formed from either a contact or enclosure dynamic region.

We now define a *goal region* as a special kind of dynamic region (in addition to $\mathcal{D}_i, i \in \{1, \ldots, m\}$). We first define a subset $G \subseteq \mathcal{W}$ as a goal region in the workspace. We next consider G as a *contact goal region* (or *enclosure goal region*), and apply (4) (or (5)) with $\mathcal{D}_i = G$ to obtain the goal region in configuration space. A subset $E_g \subseteq E$ is selected, and we obtain X_G by applying (6) (or (7)). The termination condition for a given motion planning problem will be to bring the system to any state in X_G.

For a given set A, let I_A denote its characteristic function: $I_A(a) = 1$ if $a \in A$, and $I_A(a) = 0$ otherwise. We assume that a loss functional is of the following additive form, which is often used in optimal control theory [8]:

$$L(x_1, \ldots x_{K+1}, u_1, \ldots, u_K) = \sum_{k=1}^{K} l_k(x_k, u_k) + l_{K+1}(x_{K+1}). \qquad (8)$$

The first K terms correspond to costs that are received at each step during the execution of the strategy. The final term, l_{K+1}, is a terminal cost that will be used to ensure that the robot achieves the goal (if the goal is reachable).

The term l_k in (8) is defined as $l_k(x_k, u_k) =$

$$\begin{cases} 0 & \text{If } x_k \in X_G \\ c_u + \sum_{i=1}^{m} [c_i I_{X_i}(x_k) + c_i' I_{X_i^c}(x_k)] & \text{Otherwise} \end{cases}. \qquad (9)$$

The constant $c_u \geq 0$ corresponds to the cost for choosing an action. This cost will often be the same for every $u_k \in U$, but in general can be dependent on the particular action. For instance, if time optimality is considered, then $c_u = \Delta t$. However, if distance optimality is considered, then one might choose $c_u = 0$ if $u_k = \emptyset$, and $c_u = \|v\| \Delta t$ otherwise.

The constant $c_i \geq 0$ is a penalty that is added if $x_k \in X_i$. The constant $c_i' \geq 0$ is a penalty that is added if $x_k \notin X_i$. In (9), X_i^c denotes $X \setminus X_i$. For the case of a changing configuration space, for instance, these constants could become $c_i = \infty$, to indicate that a collision has occurred, and $c_i' = 0$ otherwise.

The term $l_{K+1}(x_{K+1})$ in (8) is defined as $c_f I_{X_G^c}(x_{K+1})$, in which X_G^c denotes $X \setminus X_G$. The constant c_f can be considered as the cost of failure. We typically consider $c_f = \infty$, but can also associate a finite cost with failure.

In the companion paper [12], we will apply the general model presented so far to obtain the solution of specific motion planning problems. In the rest of the paper, we consider the generalizations of the basic framework presented so far, to handle more complex situations.

3 Incorporating Type ES Uncertainty

It has been assumed so far that at stage k the robot knows the environment mode, e_k. In general, it could be the case that the robot has limited sensing, and cannot perfectly determine the current environment mode. We assume that the environment transition probabilities depend only on the previous environment mode, and hence can be written $P(e_{k+1}|e_k)$. The motion planning problem is analyzed in the *information space*, which in this case is represented by a space of density functions. Detailed treatment of information spaces in optimal control theory can be found in [8], and their application to motion planning with uncertainty in control and sensing appears in [11].

In this section we briefly describe how successive distributions over E can be obtained. Suppose that the robot is equipped with a sensor that produces an observation o_k at each stage, $k \in \{1, \ldots, K\}$. We assume that a noise or error model for the sensor can be specified as $P(o_k|e_k)$. This characterizes the observations that are likely to be made for a given environment mode. The form $P(o_k|e_k)$ is typically used in a variety of robotics applications that involve statistical sensor error [6], and in general for stochastic control theory [8].

We proceed by induction, using $P(e_1)$ as a given basis, and the transition from $P(e_k|o_k, \ldots, o_1)$ to $P(e_{k+1}|o_{k+1}, \ldots, o_1)$ as the inductive step. If we have $P(e_k|o_k, \ldots, o_1)$, then before a new observation, the posterior distribution of e_{k+1} can be determined as

$$P(e_{k+1}|o_k, \ldots, o_1) = \sum_{e_k \in E} P(e_{k+1}|e_k) P(e_k|o_k, \ldots, o_1). \qquad (10)$$

The new observation, o_{k+1}, can be incorporated to obtain $P(e_{k+1}|o_{k+1}, \ldots, o_1) =$

$$\frac{P(o_{k+1}|e_{k+1}, o_k, \ldots, o_1) P(e_{k+1}|o_k, \ldots, o_1)}{P(o_{k+1}|o_k, \ldots, o_1)} \qquad (11)$$

in which $P(o_{k+1}|o_k, \ldots, o_1) =$

$$\sum_{e_{k+1} \in E} P(o_{k+1}|e_{k+1}, o_k, \ldots, o_1) P(e_{k+1}|o_k, \ldots, o_1). \qquad (12)$$

By making appropriate substitutions above, and by reducing conditionals, we obtain $P(e_{k+1}|o_{k+1}, \ldots, o_1) =$

$$\frac{P(o_{k+1}|e_{k+1}) \sum_{e_k \in E} P(e_{k+1}|e_k) P(e_k|o_k, \ldots, o_1)}{\sum_{e_k \in E} \sum_{e_{k+1} \in E} P(o_{k+1}|e_{k+1}) P(e_{k+1}|e_k) P(e_k|o_k, \ldots, o_1)}. \qquad (13)$$

Equation (13) defines $P(e_{k+1}|o_{k+1}, \ldots, o_1)$ in terms of the following probabilities: $P(e_{k+1}|e_k)$, $P(e_k|o_k, \ldots, o_1)$, and $P(o_{k+1}|e_{k+1})$, which are given. Hence, at each stage during the execution of a strategy, a new posterior distribution can be computed.

4 Time-Varying Strategies

The strategies that have been discussed up to this point are stationary in the sense that the robot actions only depend on the state. In turns out that with little effort, the model components can be allowed to vary over time. This affords the opportunity to model many interesting problems, such as the incorporation of known moving obstacles.

We briefly describe the general time-varying components that can be defined to yield nonstationary solutions. Suppose that the workspace contains obstacles, $\mathcal{B}_1(t), \ldots, \mathcal{B}_q(t)$, that may possibly be in motion. This results in a time-varying free configuration space, $\mathcal{C}_{free}(t)$ [9]. To handle discrete time, at each stage, k, we define a *stage-dependent* free configuration space

$$\mathcal{C}_{free}[k] = \bigcap_{t \in [(k-1)\Delta t, k\Delta t)} \mathcal{C}_{free}(t). \quad (14)$$

In addition, we can have moving dynamic regions $\mathcal{D}_1(t), \ldots, \mathcal{D}_m(t)$. In configuration space each of these becomes $\mathcal{CD}_i^c(t)$ or $\mathcal{CD}_i^e(t)$, and in the state space we have $X_1(t), \ldots, X_m(t)$. As done in (14), we can similarly define $X_1[k]$, ..., $X_m[k]$ to be *stage-dependent* dynamic X-regions. To obtain the appropriate loss functional, we simply replace (9) by $l_k(x_k, u_k) =$

$$\begin{cases} 0 & \text{If } x_k \in X_G \\ c_u + \sum_{i=1}^{m} [c_i I_{X_i[k]}(x_k) + c_i' I_{X_i^c[k]}(x_k)] & \text{Otherwise} \end{cases} . \quad (15)$$

In addition to the time-varying components discussed above, additional components can vary with time. By allowing the environment transition probabilities to vary, many more statistical processes can be modeled. For instance, it might be known that the workspace is more likely to become hazardous after some prescribed time, or become increasingly more likely to be hazardous over time. We can also allow the goal region to move over time, to obtain $X_G[k]$. In this case, the robot must intercept the moving goal as a terminating condition for the strategy.

5 Conclusions

We have characterized the problem of probabilistically handling Type EP uncertainty as a problem of stochastic optimal control. One could alternatively model the environment with bounded uncertainty, leading to worst-case analysis. One could also consider a continuum of environments.

In a broader setting, the combination of the additional sources of uncertainty from Section 1 should be addressed. We argue that the framework presented here can facilitate such a combination. In Section 3, we described how Type ES uncertainty can be incorporated. In addition, however, Type RP and Type RS can also be incorporated in a straightforward manner. A treatment of these forms of uncertainty in motion planning that uses concepts similar to those presented here can be found in [11]. The incremental motion model can be defined stochastically, to reflect Type RP uncertainty. The information space concepts from Section 3 can be expanded to include complete sensing history that characterizes uncertainty in the robot configuration. Computational issues involved in a combination of this form will depend directly on the dimensions of the state and information spaces for the problem.

References

[1] J. Barraquand, B. Langlois, and J. C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Trans. Syst., Man, Cybern.*, 22(2):224–241, 1992.

[2] 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.

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

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

[5] H. F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Trans. on Robotics and Automation*, 4(1):23–31, February 1988.

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

[7] J. A. Hendler and J. C. Sanborn. Planning and reaction in dynamic domains. In *Proc. DARPA Workshop on Knowledge-Based Planning Systems*, 1987.

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

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

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

[11] S. M. LaValle and S. A. Hutchinson. An objective-based stochastic framework for manipulation planning. In *IEEE/RSJ/GI International Conference on Robots and Systems*, September 1994.

[12] S. M. LaValle and R. Sharma. A framework for motion planning in stochastic environments: Applications and computational issues. *Proc. 1995 IEEE International Conference on Robotics and Automation.*

[13] 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.

[14] R. Sharma, D. M. Mount, and Y. Aloimonos. Probabilistic analysis of some navigation strategies in a dynamic environment. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(5):1465–1474, September 1993.

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

[16] Q. Zhu. Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Trans. on Robotics and Automation*, 7(3):390–397, June 1991.