Comparing the Power of Robots

Jason M. O'Kane and Steven M. LaValle

The International Journal of Robotics Research 2008 27: 5
DOI: 10.1177/0278364907082096

Comparing the Power of Robots

Jason M. O'Kane
Steven M. LaValle
Department of Computer Science, University of Illinois
at Urbana-Champaign, 201 North Goodwin Avenue
Urbana, IL 61801, USA
{jokane, lavalle}@cs.uiuc.edu

Abstract

Robots must complete their tasks in spite of unreliable actuators and limited, noisy sensing. In this paper, we consider the information requirements of such tasks. What sensing and actuation abilities are needed to complete a given task? Are some robot systems provably "more powerful", in terms of the tasks that they can complete, than others? Can we find meaningful equivalence classes of robot systems? This line of research is inspired by the theory of computation, which has produced similar results for abstract computing machines. Our basic contribution is a dominance relation over robot systems that formalizes the idea that some robots are stronger than others. This comparison, which is based on how the robots progress through their information spaces, induces a partial order over the set of robot systems. We prove some basic properties of this partial order and show that it is directly related to the robots' ability to complete tasks. We give examples to demonstrate the theory, including a detailed analysis of a limited-sensing global localization problem.

KEY WORDS—information spaces, robot dominance, sensors, minimalism

1. Introduction

Suppose that we want a robot to complete some task, such as navigating to a goal, manipulating an object or localizing itself within its environment. Many different combinations of sensing and motion modalities have been used to complete each of these tasks. Indeed, much of the robotics literature is concerned with finding *sufficient conditions* on the sensing and actuation capabilities needed to complete such tasks. In this paper we take a different approach. For a given task, we are interested in determining the *necessary conditions*. What sensors and actuators are needed? What are the *information requirements* of robotic tasks? The long-term goal of this research is to develop a theory of robots and sensing that helps to answer such questions. Answers to these questions are important because we expect that a deep understanding of the difficulty of tasks in terms of their information requirements will lead to simpler and less-expensive robot designs.

This work is inspired in part by the theory of computation, which begins with precisely defined models of abstract machines, such as finite automata, Turing machines, and so on (Sipser 1997; Hopcroft et al. 2000). In this context, a *problem* is usually a language of strings; to solve the problem is to accept strings in this language and reject all others. The theory of computation gives answers to several kinds of basic questions about these machines and problems:

1. *Solvability*: Can a given machine solve a given problem?
2. *Complexity*: If the machine can solve the problem, how efficiently (in terms of time or space, for example) can it do so?
3. *Comparison*: Are some machines strictly more powerful, in terms of the problems they can solve, than others? It is known, for example, that pushdown automata can accept a strictly larger set of languages than finite automata are able to. Likewise, Turing machines are more powerful than pushdown automata.
4. *Equivalence*: Are there apparently dissimilar machines that can solve the same set of problems? For example, it is a standard result that a Turing machine with multiple tapes is functionally equivalent to an ordinary single-tape Turing machine. Less obviously, Turing machines and recursive functions have been shown to have equivalent computation power.

These ideas are well understood. In the sense that they form the formal foundation of the discipline, they are part of the core of computer science. Current robotic science lacks a comparable foundation; the field needs a unified theory in which

The International Journal of Robotics Research
Vol. 27, No. 1, January 2008, pp. 5–23
DOI: 10.1177/0278364907082096

Figures 1–5, 7–13, 15–16 appear in color online: http://ijr.sagepub.com

meaningful statements can be made about the complexity of robotic tasks and the robot systems we build to complete these tasks.

Can we adapt standard models of computation to the robotics context? Unfortunately, these models are fundamentally ill-suited for studying robotics problems, because they assume that all of the relevant information is supplied ahead of time on the machine's tape. Sensing and uncertainty are central, defining issues in robotics; this structure is destroyed by an *a priori* encoding of the problem on a machine's tape. Traditional models of online computation (Sleator and Tarjan 1985; Karp 1992; Borodin and El-Yaniv 1998) are also inadequate, because they assume that some fixed encoding of the problem is revealed incrementally. In contrast, robotics problems are generally interactive, in the sense that the robot's decisions influence the information that becomes available in the future. Others study robotics problems use similar tools (Papadimitriou and Yannakakis 1991; Gabriely and Rimon 2004), but do not explicitly consider the effects of varying sensing and motion capabilities.

The aim of this paper is to develop a "sensor-centered" theory for analyzing and comparing robot systems. The central idea we present is the notion of *dominance* of one robot model over another. In informal terms:

> A robot R_2 *dominates* another robot R_1 if R_2 can "simulate" R_1, collecting at least as much information as R_1.

We make three primary contributions in developing this idea. First, we present the idea of *robotic primitives* for modeling robot systems as collections of independent components. A single robotic primitive represents a self-contained "instruction set" for the robot that may involve sensing, motion or both. A robot model is defined by a set of primitives that the robot can use to complete its task. By selecting a "catalog" of primitives from which complete robot systems are constructed, we effectively determine a set of robot systems to consider. For clarity, we define these models in an idealized setting in which time is modeled as a series of discrete stages and the robot has perfect knowledge of its environment, perfect control and perfect sensing. Second, we give a definition for the dominance of one robot system over another that formalizes the imprecise definition above. This definition is based on comparing reachability in a *derived information space* (LaValle 2006). By mapping sensor–action histories from a variety of robots into the same derived information space, we can compare the abilities of these robots in a concrete, formal way. We prove some basic properties of this dominance relation and give some examples, including a detailed investigation of the global localization problem. Third, we demonstrate the generality of our ideas by showing how to remove several of the simplifying assumptions we make in the initial presentation.

Our approach is based on two main ideas:

1. *Information spaces*. Traditional planning methods focus on the robot's progression through a space of states. What happens when the state is hidden and sensing thereby becomes relevant? One approach is to use *state estimation*, in which the robot uses the information available to it to make an "educated guess" about its state. The robot can treat this estimated state as its true state and ignore the uncertainty. In some extremely limited contexts this is provably optimal (see, for example, Bertsekas (2001, Section 6.1)). We, however, are interested in a broader class of tasks for which accurate state estimation is impossible.

   The relevant space for such problems is the robot's *information space*. This space fully describes the information available to the robot, including its initial condition, the history of actions it has applied and the history of sensor observations it has received. The robot's "state" in this space is always fully known. Information spaces originated in game theory (Kuhn 1953), but have been used in robotics for some time (Goldberg and Mason 1990; Erdmann 1993; Barraquand and Ferbach 1995; LaValle and Hutchinson 1998; LaValle 2006).

2. *Tradeoffs expressed as partial orders*. We present a partial order defining the *dominance* of one robot system over another. The definition, in turn, is based on another partial order, an *information preference relation* over information space, that indicates which information states are preferred to others. Although these relations admit the possibility that no meaningful comparisons can be made, we find this desirable: physical tasks and robot systems exhibit complex relationships and tradeoffs that can potentially defy meaningful linear ordering.

The challenge of robotics lies in the interactions between sensing, actuation and computation. In this paper, we focus the effects of varying choices for the robot's sensing and actuation capabilities. The robot's computational abilities (as measured, for example, by processing power or memory limitations) are also relevant, but we do not consider them here.

The remainder of this paper is organized as follows. Section 2 reviews related research. Section 3 lays a foundation of basic definitions for robotic planning problems. Section 4 introduces the concept of robotic primitives and defines the set of robots induced by a catalog of primitives. In Section 5, we describe the information preference relation. The definition of dominance and some basic properties thereof appear in Section 6. In Section 7, we apply the results from Sections 4–6 to the global localization task. In Section 8, we present several generalizations of our basic results to account for environment uncertainty, imperfect control and sensing, and continuous time. Section 9 discusses the limitations of this work and describes some open problems.

Preliminary versions of this work appeared in O'Kane and LaValle (2006, 2007a).

2. Related Work

Our approach can be viewed as *minimalist* in the sense that we are interested in solutions that use sensing sparingly. The minimalist approach in robotics has a long history, dating back perhaps to Whitney (1986). Minimalist approaches have been used in manufacturing contexts for part orientation (Erdmann and Mason 1988; Goldberg and Mason 1990; Goldberg 1993; Wiegley et al. 1997; Akella et al. 2000; van der Stappen et al. 2000; Agarwal et al. 2001; Moll and Erdmann 2002) and in mobile robotics for navigation and exploration (Dúnlaing and Yap 1982; Kutulakos et al. 1994; Lumelsky and Tiwari 1994; Choset and Burdick 1995; Kamon et al. 1999; Acar and Choset 2001b; Tovar et al. 2004).

Our goals are similar to those of Donald (1995). The reductions in that work are similar to our dominance relation; Donald's notion of calibration is related to our idea of initial conditions. The most fundamental difference is that our analysis is rooted in the information space. We claim that for robotic problems in which sensing is a crucial issue, the information space is the space in which the problem can most naturally be posed. The work of Erdmann (1995) is grounded in the preimage planning ideas due to Lozano-Pérez et al. (1984). In Erdmann's work, sensors are modeled by giving a partition of state space. The problem of sensor design is to choose a partition so that from each region in the partition, the robot knows what action to select in order to make progress toward its goal. Others in artificial intelligence Brafman et al. (1998) and control theory (Egerstedt 2002; Girard and Pappas 2007; Abate et al. 2006) have addressed related issues.

Although the examples in this paper use non-deterministic uncertainty, which is based on set membership, the basic structure of our analysis is compatible with probabilistic uncertainty models such as those of Thrun et al. (2005). Many probabilistic methods (for example, Austin and Jensfelt (2000) or Lenser and Veloso (2000)) can be characterized as operating in an information space whose members are probability distributions over state space. Our methods can be viewed as axiomatic because they can be applied in any situation that satisfies the definitions of Sections 3–5. In this sense, the model of uncertainty used is orthogonal to the questions addressed in this work.

3. Basic Definitions

This section presents basic definitions for robotic planning problems. To keep the presentation as clear as possible, we make several simplifying assumptions here and in Section 8 we show how to relax them.

Fig. 1. A robot in a planar environment E. Its state space is $X = E \times S^1$.

3.1. States, Actions and Observations

We allow a robot to move in a state space X. Many of the examples in this paper are for a point robot with orientation in the plane. In these examples, we use $X = E \times S^1$, in which $E \subset \mathbb{R}^2$ is the robot's environment and $S^1 = [0, 2\pi]/\sim$, where $\sim$ is an equivalence relation identifying 0 and 2π, represents the robot's orientation. Note that this formulation encodes the geometry of the robot's environment into its state space. Situations in which the environment is unknown can be modeled using a richer state space, as described in Section 8.1. In general, however, we allow arbitrary state spaces, including configuration spaces and phase spaces of physical systems.

Time proceeds in variable-length *stages*, indexed by consecutive integers starting with 1. In each stage, the robot selects an action u from its action space U and moves to a new state according a state transition function $f : X \times U \to X$. At the conclusion of each stage, the robot's sensors provide an observation y from an observation space Y, according to $h : X \times U \to Y$. Call h the robot's *observation function*. Let x_k, u_k and y_k denote the state, action and observation at stage k, respectively. These sequences are related to each other by f and h:

$$x_{k+1} = f(x_k, u_k) \quad (1)$$

$$y_k = h(x_k, u_k). \quad (2)$$

Although we are assuming in this section that both state transitions and observations are deterministic, we acknowledge that in realistic contexts, managing unpredictability in motion and sensing is a crucial issue. We omit such uncertainty here because of the additional complications it would introduce. The extensions needed to relax this assumption are introduced in Section 8.

For convenience, we also define an iterated version of f that applies k actions in succession:

$$f(x, u_1, \ldots, u_k) = f(\cdots f(f(x, u_1), u_2) \cdots, u_{k-1}), u_k). \quad (3)$$

The robot's capabilities are modeled in the action and observation sets U and Y and in the maps f and h that interpret these

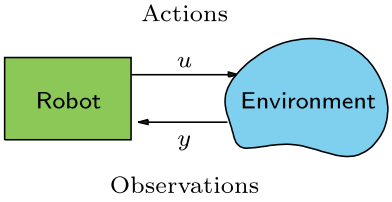

Fig. 2. The robot interacts with its environment by executing actions and receiving observations.

sets. See Figure 2. A *robot model* is a 5-tuple (X, U, Y, f, h) giving values to each of these elements.

3.2. Information Spaces

Although the robot does not know its state, it does have access to the history of actions it has selected and observations it has made. The space of such histories is the robot's *history information space* (history I-space), denoted by $\mathcal{I}_{\text{hist}}$:

$$\mathcal{I}_{\text{hist}} = \bigcup_{i=0}^{\infty} (U \times Y)^i. \tag{4}$$

After k stages, the robot's *history information state* (history I-state) is a sequence of length $2k$:

$$\eta_k = (u_1, y_1, \ldots, u_k, y_k). \tag{5}$$

We occasionally abuse notation by writing $(\eta_k, u_{k+1}, y_{k+1})$ for the history I-state formed by appending u_{k+1} and y_{k+1} to η_k. How is the state space related to the robot's history I-space? One connection is by way of the notion of states consistent with an I-state:

Definition 1. A state $x \in X$ is *consistent with* a history I-state $\eta_k = (u_1, y_1, \ldots, u_k, y_k)$ if there exists some $x_1 \in X$ such that $x = f(x_1, u_1, \ldots, u_k)$ and $y_j = h(f(x_1, u_1, \ldots, u_{j-1}), u_j)$ for each $j = 1, \ldots, k$.

The intuition is that a state x is consistent with an I-state η_k if a robot having I-state η_k might possibly be at state x.

We may define a *policy* $\pi : \mathcal{I}_{\text{hist}} \to U$ over history I-space. Note that, given a state x_k and a history I-state η_k, the history I-states reached by repeatedly executing π are fully determined. As a shorthand, we define a function F that applies a policy several times in succession, so that m applications of a policy π, starting at state x_k and information state η_k, lead to a new history I-state given by

$$\eta_{m+k} = F^m(\eta_k, \pi, x_k). \tag{6}$$

Note that $F^m(\eta_k, \pi, x_k)$ depends on the true state x_k (which is unknown to the robot) because x_k influences the observation sequence the robot receives.

The history I-space is not particularly useful by itself. For pairs of robots whose action or observation spaces differ, the history I-spaces also differ, making the history I-space unhelpful for comparing robots. Therefore, we select a *derived information space* (derived I-space) $\mathcal{I}$ and an *information mapping* (I-map) $\kappa : \mathcal{I}_{\text{hist}} \to \mathcal{I}$. Informally, an I-map computes a "compression" or "interpretation" of the history I-state. If the history I-spaces of several robot models are mapped to the same derived I-space $\mathcal{I}$, then the robots can be compared by examining their progression through $\mathcal{I}$. In principle, we may select $\mathcal{I}$ and κ arbitrarily. The usefulness of a derived I-space lies in its ability to capture the information relevant to the task of interest.

Example 1. We define the *non-deterministic I-space* $\mathcal{I}_{\text{ndet}}$, in which derived I-states are non-empty subsets of X. The interpretation is that the robot's derived I-state is the minimal set guaranteed to contain the true state. For any history I-state η, the non-deterministic derived I-state $\kappa_{\text{ndet}}(\eta)$ is the set of states consistent with η. Equivalently, the I-map $\kappa_{\text{ndet}} : \mathcal{I}_{\text{hist}} \to \mathcal{I}_{\text{ndet}}$ can be defined recursively:

$$\kappa_{\text{ndet}}(\,) = X \tag{7}$$

$$\kappa_{\text{ndet}}(\eta, u, y) = \{f(x, u) \mid x \in \kappa_{\text{ndet}}(\eta), y = h(x, u)\} \tag{8}$$

Note that in Equation 7 we assume that the robot initially has no information about its state.

An important special case is the value of κ for an empty history, that is, $\kappa(\,)$. This value gives an *initial condition* for the robot, reflecting any knowledge the robot may have before its execution begins.

A *task* for the robot is a goal region $\mathcal{I}_G \subseteq \mathcal{I}$ in derived I-space that the robot must reach. This notion is a generalization of the traditional idea of a goal state or goal region in state space. A *solution* is a policy π under which, for any $x \in X$, there exists some l such that $F^l(\eta_1, \pi, x) \in \mathcal{I}_G$.

4. Defining a Set of Robot Systems

In this section we discuss how a set of robots can be defined in terms of a set of independent components.

4.1. Robotic Primitives

At the most concrete level, a robot is a collection of motors and sensors connected to some sort of computer. Between these components there may be interactions by means of open- or closed-loop controls. We abstract this complexity by defining the notion of a *robotic primitive*. Each robotic primitive defines

a "mode of operation" for the robot. When primitives are implemented, they may draw on one or more of the robot's physical sensors or actuators. Every kind of motion or sensing available to the robot must be modeled as a robotic primitive. Robotic primitives correspond roughly to the *oracles* that appear in the theory of computation (Soare 1987; Sipser 1997), in the sense that they provide the ability to make certain transitions and collect certain observations, without specifying how these abilities are implemented.

Formally, we define robotic primitives in terms of the action and observation abilities they provide.

Definition 2. A *robotic primitive* (or simply a *primitive*) P_i is a 4-tuple

$$P_i = (U_i, Y_i, f_i, h_i) \tag{9}$$

giving an action set U_i, an observation set Y_i, a state transition function $f_i : X \times U_i \to X$ and an observation function $h_i : X \times U_i \to Y_i$.

Let $\mathcal{RP} = \{P_1, \ldots, P_N\}$ denote a catalog of primitives. We may form a robot model by selecting non-empty subset of $\mathcal{RP}$. A robot defined by the primitive set $R = \{P_{i_1}, \ldots, P_{i_m}\} \subseteq \mathcal{RP}$ has the action set $U_R = U_{i_1} \sqcup \cdots \sqcup U_{i_m}$ and observation set $Y_R = Y_{i_1} \sqcup \cdots \sqcup Y_{i_m}$. The $\sqcup$ notation indicates a disjoint union operation, under which identical elements from different source sets remain distinct. The state transition function $f_R : X \times U_R \to X$ and observation function $h_R : X \times U_R \to Y_R$ are formed by unioning the f and h maps from the relevant primitives. When it can be done without ambiguity, we use the phrase *robot model* to refer directly to the set of primitives, rather than to the 5-tuple (X, U, Y, f, h) formed by these primitives. With this usage, it is meaningful to apply set operations such as union or intersection directly to robots.

Note that, given a catalog of primitives $\mathcal{RP}$, we can form a "master" robot model $\widehat{R}$ that includes every primitive in $\mathcal{RP}$. The history I-space of $\widehat{R}$ contains as a subset the history I-space of every other robot model that can be formed from $\mathcal{RP}$. As a result, any I-map for $\widehat{R}$ can also be used as an I-map for any robot model formed from $\mathcal{RP}$.

We now give several examples to illustrate the intuition of Definition 2. Examples 3–7 apply to a point robot with orientation in a bounded planar environment E, so $X = E \times S^1$. Illustrations of these primitives appear in Figures 3–5. We revisit these examples in Sections 6 and 7.

Example 2. Let $P_A = (S^1, \{0\}, f_A, h_A)$. Let f_A compute relative rotations, so that from a state $x = (x_1, x_2, \theta)$, we have $f_A(x, u) = (x_1, x_2, \theta + u)$. Since $Y_A = \{0\}$ contains only a dummy element, h_A is a trivial function always returning 0. This primitive can be implemented with an angular odometer on a mobile robot capable of rotating in place.

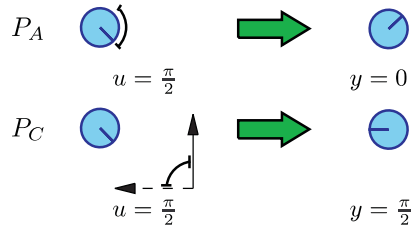

Fig. 3. Sample executions of the primitives of Examples 2 and 3. Top: P_A allows the robot rotate relative to its current orientation. Bottom: P_C allows the robot to rotate relative to a globally defined "north" direction.

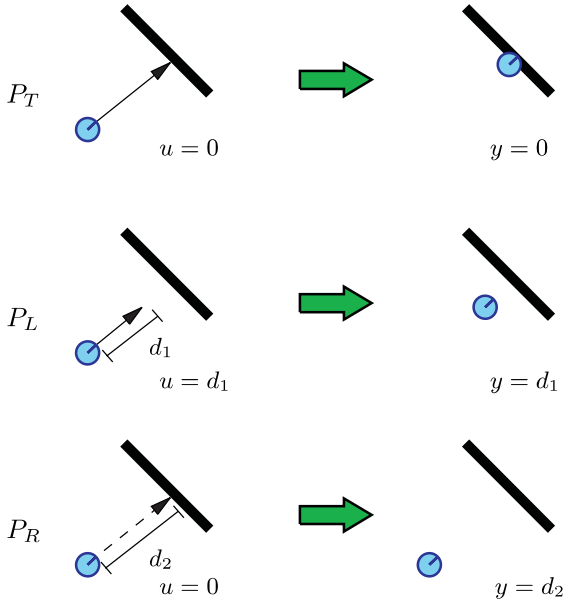

Fig. 4. Sample executions of the primitives of Examples 4–6. Top: P_T allows the robot to translate forward until it reaches an obstacle. Middle: P_L allows a robot to specify a distance to translate. Bottom: P_R allows the robot to measure the distance forward to the nearest obstacle, but does not change the robot's state.

Example 3. Let $P_C = (S^1 \sqcup \{0\}, S^1, f_C, h_C)$. Define $f_C(x, u)$ to set the rotation coordinate of x equal to u if $u \in S^1$ or to leave x unchanged if $u \in \{0\}$. The observation function h_C returns the robot's final orientation. This primitive amounts to allowing the robot to orient itself with respect to a global reference frame or to sense its current orientation without rotating. One might implement this primitive using a compass on a robot that can rotate in place.

Example 4. Let $P_T = (\{0\}, \{0\}, f_T, h_T)$. Define f_T to compute a forward translation to the obstacle boundary. This prim-

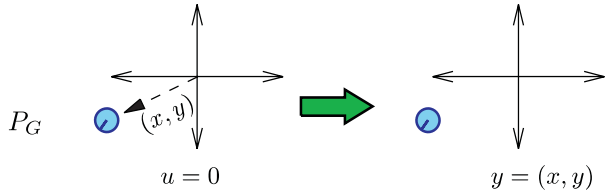

Fig. 5. A sample execution of the primitive of Example 7. The robot senses its position, but its state does not change.

itive can be implemented with a contact sensor on a mobile robot that can reliably move forward.

Example 5. Let $P_L = ([0, \infty), [0, \infty), f_L, h_L)$. For $x \in X$ and $u \in U$, define $f_L(x, u)$ to compute a forward translation of distance at most u, stopping short only if the robot reaches an obstacle first. The observation $h_L(x, u)$ is the actual distance traveled. This primitive can be implemented with a linear odometer on a robot that can move forward reliably. Depending on implementation issues, a contact sensor may also be needed.

Example 6. Let $P_R = (\{0\}, [0, \infty), f_R, h_R)$. For all $x \in X$, $f_R(x, 0) = x$, so that this primitive never changes the robot's state. The observation $h_R(x, u)$ is the distance to the nearest obstacle directly in front of the robot. This primitive models the capabilities of a forward-facing unidirectional range sensor.

Example 7. Let $P_G = (\{0\}, \mathbb{R}^2, f_G, h_G)$. Again, $f_G(x, u) = x$ for all x and u. For a state $x = (x_1, x_2, \theta)$, let $h_G(x, 0) = (x_1, x_2)$. This primitive roughly corresponds to a GPS device that the robot can periodically poll to determine its location in the plane.

Other possibilities for primitives include landmark detectors, wall followers, visibility sensors and so on. A more complete listing of sensors suitable for adaptation into robotic primitives appears in LaValle (2006, Section 11.5.1).

There are several benefits to modeling robot systems as collections of primitives. First, we claim that robotic primitives represent the right level of abstraction at which *planning* problems are interesting but manageable. If we consider sensors at too fine a level of detail, the problem takes on the character of a closed-loop control system. If the primitives are too sophisticated, we risk trivializing the planning problem while creating an unbearable modeling burden. Second, by dividing time into discrete stages, we avoid the technical difficulties of describing the robot's progression through $\mathcal{I}$ in continuous time. This consideration is increasingly important if we allow noise to affect state transitions or observations. We address issues related to the modeling of time more completely in Section 8.3.

5. The Information Preference Relation

Our goal is a dominance relation under which we can declare one robot "better than" another. To do so, we need a formal notion of one I-state being superior, in the sense of encoding better information, than another. To this end, choose a derived I-space $\mathcal{I}$ and an I-map κ into $\mathcal{I}$. Equip $\mathcal{I}$ with a partial order, which we call an *information preference relation*. Write $\kappa(\eta_1) \preceq \kappa(\eta_2)$ to indicate that $\kappa(\eta_2)$ is preferred to $\kappa(\eta_1)$. We require that for any $\eta_1, \eta_2 \in \mathcal{I}_{\text{hist}}$, and for any $u \in U$ and $y \in Y$,

$$\kappa(\eta_1) \preceq \kappa(\eta_2) \Rightarrow \kappa(\eta_1, u, y) \preceq \kappa(\eta_2, u, y). \tag{10}$$

This is a consistency property requiring preference for one I-state over another to be preserved across transitions in I-space.

Example 8. Regardless of $\mathcal{I}$ or κ, it is well defined (but perhaps unhelpful) to use a trivial relation under which $\kappa(\eta_1) \preceq \kappa(\eta_2)$ if and only if $\kappa(\eta_1) = \kappa(\eta_2)$.

Example 9. Under non-deterministic uncertainty, we can define $\kappa_{\text{ndet}}(\eta_1) \preceq \kappa_{\text{ndet}}(\eta_2)$ if and only if $\kappa_{\text{ndet}}(\eta_2) \subseteq \kappa_{\text{ndet}}(\eta_1)$. To show that (10) is satisfied, suppose $\kappa_{\text{ndet}}(\eta_1) \preceq \kappa_{\text{ndet}}(\eta_2)$. Let $x \in \kappa_{\text{ndet}}(\eta_2, u, y)$. The definition of κ_{ndet} ensures that there exists some $x' \in \kappa_{\text{ndet}}(\eta_2)$ such that $f(x', u) = x$ and $h(x', u) = y$. However, because $\kappa_{\text{ndet}}(\eta_2) \subseteq \kappa_{\text{ndet}}(\eta_1)$, we have $x' \in \kappa_{\text{ndet}}(\eta_1)$. It follows that $x \in \kappa_{\text{ndet}}(\eta_1, u, y)$.

The information preference relation we choose affects the goal regions that are sensible to consider. We should select a region in which, for every I-state in the region, we also include any I-states preferable to it. This formalizes the intuition that a robot in the goal region should not prefer to be outside the goal. Definition 3 codifies this idea of a sensible goal region.

Definition 3. Consider a set $I \subset \mathcal{I}$ of derived I-states. If, for any $\kappa(\eta_1) \in I$ and $\kappa(\eta_2) \in \mathcal{I}$ with $\kappa(\eta_1) \preceq \kappa(\eta_2)$, we have $\kappa(\eta_2) \in I$, then I is *preference closed*.

Alternatively, one can view preference closure as a constraint on $\preceq$. Fixing a space $\mathcal{G}$ of potential goal regions, we admit a partial order $\preceq$ only if every region in $\mathcal{G}$ is preference closed under $\preceq$. Note that the trivial definition of $\preceq$ in Example 8 always passes this test, regardless of $\mathcal{G}$.

6. A Dominance Relation over Robot Systems

Now we turn our attention to a definition of dominance of one robot system over another. This dominance relation induces a partial order over robot systems, according to their sensing and actuation abilities. The intuition is that dominance is based on one robot's ability to "simulate" another.

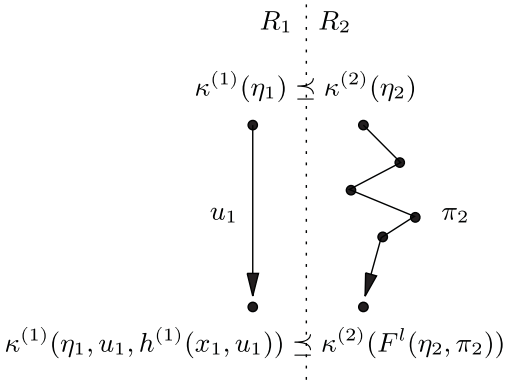

Fig. 6. An illustration of Definition 4. If R_2 can always reach an I-state better than that reached by R_1, then $R_1 \trianglelefteq R_2$.

Definition 4. (Robot dominance) Consider two robots

$$R_1 = (X^{(1)}, U^{(1)}, Y^{(1)}, f^{(1)}, h^{(1)})$$

and

$$R_2 = (X^{(2)}, U^{(2)}, Y^{(2)}, f^{(2)}, h^{(2)}).$$

Choose a derived I-space $\mathcal{I}$ and I-maps $\kappa^{(1)} : \mathcal{I}^{(1)}_{\text{hist}} \to \mathcal{I}$ and $\kappa^{(2)} : \mathcal{I}^{(2)}_{\text{hist}} \to \mathcal{I}$. If, for all:

- $\eta_1 \in \mathcal{I}^{(1)}_{\text{hist}}$;
- $\eta_2 \in \mathcal{I}^{(2)}_{\text{hist}}$ for which $\kappa^{(1)}(\eta_1) \preceq \kappa^{(2)}(\eta_2)$; and
- $u_1 \in U^{(1)}$;

there exists a policy $\pi_2 : \mathcal{I}^{(2)}_{\text{hist}} \to U^{(2)}$ such that for all $x_1 \in X^{(1)}$ consistent with η_1 and all $x_2 \in X^{(2)}$ consistent with η_2, there exists a positive integer l such that

$$\kappa^{(1)}(\eta_1, u_1, h^{(1)}(x_1, u_1)) \preceq \kappa^{(2)}(F^l(\eta_2, \pi_2, x_2)), \qquad (11)$$

then R_2 *dominates* R_1 under $\kappa^{(1)}$ and $\kappa^{(2)}$, denoted by $R_1 \trianglelefteq R_2$. If $R_1 \trianglelefteq R_2$ and $R_2 \trianglelefteq R_1$, then R_1 and R_2 are *equivalent*, denoted by $R_1 \equiv R_2$. If $R_1 \not\trianglelefteq R_2$ and $R_2 \not\trianglelefteq R_1$ then R_1 and R_2 are *incomparable*, denoted by $R_1 \boxtimes R_2$.

Informally, Definition 4 means that, for any transition made by R_1, there exists some strategy for R_2 to reach an information state at least as good, in the sense of information preference, as that reached by R_1. This is what we mean when we describe the statement $R_1 \trianglelefteq R_2$ as meaning that R_2 can simulate R_1. See Figure 6.

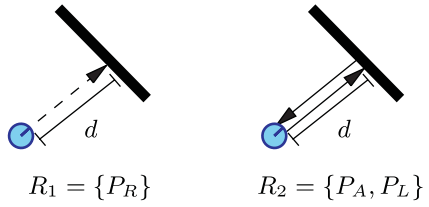

Fig. 7. An illustration of Example 10. The robot $R_2 = \{P_A, P_L\}$ dominates the robot $R_1 = \{P_R\}$ because the former can simulate the latter. (a) A distance measurement made directly by R_1. (b) Distance is measured indirectly by R_2 using its linear odometer.

6.1. Dominance Examples

The following examples help to clarify the definition.

Example 10. Let $R_1 = \{P_R\}$ and $R_2 = \{P_A, P_L\}$. Recall the definitions of these primitives from Examples 3, 5 and 6. We argue under non-deterministic uncertainty that $R_1 \trianglelefteq R_2$ by showing that R_2 can simulate R_1 in the precise sense of Definition 4. Let $\eta_1 \in \mathcal{I}^{(1)}_{\text{hist}}$ and $\eta_2 \in \mathcal{I}^{(2)}_{\text{hist}}$ with $\kappa_{\text{ndet}}(\eta_1) \preceq \kappa_{\text{ndet}}(\eta_2)$. Since $U^{(1)} = \{0\}$, there is only one choice for u_1. Let $l = 4$ and define π_2 so that R_2, starting from η_2, executes these actions in succession:

(1) Use P_L with a very large input to move forward to the nearest obstacle. Let $d = h(x, u)$ denote the distance moved.

(2) Use P_A with $u = 180^\circ$ to perform a half turn.

(3) Use P_L with $u = d$ to return the robot to its initial position.

(4) Use P_A with $u = 180^\circ$ to perform a half turn, returning the robot to its original orientation.

This policy is illustrated in Figure 7. It is easy to verify that from any $x \in X$, we have

$$\kappa_{\text{ndet}}(\eta_1, u_1, h(x, u_1)) \preceq \kappa_{\text{ndet}}(F^4(\eta_2, \pi_2, x))$$

and therefore $R_1 \trianglelefteq R_2$. Since R_1, which is completely immobile, cannot simulate the translations or rotations of R_2, we have $R_2 \not\trianglelefteq R_1$.

Note that these relationships are based on the robots' ability to move through $\mathcal{I}_{\text{ndet}}$ and do not consider any notion of the cost of motion or sensing. The introduction of such a cost function would likely lead to Pareto optima that express tradeoffs between the complexity of sensing built into the robot and the execution costs of particular plans executed by the robot. We do not consider such tradeoffs here.

Example 11. Let $R_1 = \{P_T\}$ and $R_2 = \{P_L\}$. We show under non-deterministic uncertainty that $R_1 \trianglelefteq R_2$. Let $\eta_1 \in \mathcal{I}^{(1)}_{\text{hist}}$ and $\eta_2 \in \mathcal{I}^{(2)}_{\text{hist}}$ with $\kappa(\eta_1) \preceq \kappa(\eta_2)$. There is only one choice for u_1. Choose $l = 1$ and define π_2 to choose an input for P_L larger than the diameter of the environment. This causes the motions of R_1 and R_2 to be identical. The resulting derived I-states $\kappa(\eta_1')$ and $\kappa(\eta_2')$ for R_1 and R_2 are the same, except that R_2 receives a meaningful sensor reading that may reduce the resulting non-deterministic I-state. This sensor information only makes $\kappa(\eta_2')$ smaller, so the preference $\kappa(\eta_1') \preceq \kappa(\eta_2')$ is maintained. Conclude that $R_1 \trianglelefteq R_2$.

This emphasizes that the relation induced by Definition 4 depends on the I-maps used. The next two examples illustrate this.

Example 12. Let $R_1 = \{P_A\}$ and $R_2 = \{P_C\}$. We argue that $R_1 \trianglelefteq R_2$ under the usual non-deterministic I-map with the initial condition of total uncertainty. Let $\eta_1 \in \mathcal{I}^{(1)}_{\text{hist}}$ and $\eta_2 \in \mathcal{I}^{(2)}_{\text{hist}}$ with $\kappa_{\text{ndet}}(\eta_1) \preceq \kappa_{\text{ndet}}(\eta_2)$. Let $u_1 \in U_1 = S^1$. Choose $l = 2$ and define π_2 to select the following two actions:

(1) Use P_C with $u = 0$ to sense the robot's orientation without changing the state. Let θ denote this orientation.

(2) Use P_C to rotate the robot to orientation $\theta + u$ in the global frame.

As in Example 11, the resulting states for R_1 and R_2 are identical but, since R_2 knows its orientation, it may be able to eliminate some candidate states that R_1 cannot. This establishes that $R_1 \trianglelefteq R_2$. Are R_1 and R_2 equivalent under this I-map? No, because R_2 can, with a single action, sense its orientation, but this information can never be gathered by R_1. Therefore, $R_2 \not\trianglelefteq R_1$ and $R_1 \not\equiv R_2$.

Example 13. Consider a situation identical to that of Example 12, but modify κ_{ndet} for a different initial condition $\kappa_{\text{ndet}}(\,) = \mathbb{R}^2 \times \{\pi/2\}$. That is, the robot begins its execution knowing its initial orientation. At every step, R_1 knows its orientation in the global frame and can simulate R_2 using angle addition. Therefore, we have $R_2 \trianglelefteq R_1$. However, using the same reasoning as in Example 12, we know $R_1 \trianglelefteq R_2$. Therefore, for this I-map, we have $R_1 \equiv R_2$.

6.2. Properties of the Dominance Relation

We conclude this section with some basic properties that follow from Definition 4.

Lemma 1. The dominance relation $\trianglelefteq$ is a partial order. Likewise $\equiv$ is indeed an equivalence relation.

Lemma 2. Consider three robots R_1, R_2 and R_3 formed from primitives in $\mathcal{RP}$ and an I-map κ for the master robot model $\widehat{R}$ of $\mathcal{RP}$. If $R_1 \trianglelefteq R_2$ under κ, we have:

(a) $R_1 \trianglelefteq R_1 \cup R_3$ (adding primitives never hurts);

(b) $R_2 \equiv R_2 \cup R_1$ (redundancy does not help);

(c) $R_1 \cup R_3 \trianglelefteq R_2 \cup R_3$ (no unexpected interactions).

Proof.

(a) Let $\eta_1 \in \mathcal{I}^{(1)}_{\text{hist}}$, $\eta_{13} \in \mathcal{I}^{(13)}_{\text{hist}}$, and $u_1 \in U_1$. Assume $\kappa(\eta_1) \preceq \kappa(\eta_{13})$. Choose $l = 1$ and $\pi_{13}(\eta) = u_1$ for all η. For all x, we have $\kappa(\eta_1, u_1, h(x, u_1)) \preceq \kappa(\eta_{13}, u_1, h(x, u_1)) = \kappa(F^l(\eta_{13}, \pi_{13}, x))$, completing the proof.

(b) It follows from part (a) that $R_2 \trianglelefteq R_1 \cup R_2$. It remains to show that $R_1 \cup R_2 \trianglelefteq R_2$. Let $\eta_{12} \in \mathcal{I}^{(12)}_{\text{hist}}$, $\eta_2 \in \mathcal{I}^{(2)}_{\text{hist}}$ and $u_{12} \in U_2 \cup U_1$. Assume $\kappa(\eta_{12}) \preceq \kappa(\eta_2)$. Either $u_{12} \in U_1$ or $u_{12} \in U_2$. If $u_{12} \in U_1$, then because $R_1 \trianglelefteq R_2$ there exist π_2 and l satisfying the definition for $R_1 \cup R_2 \trianglelefteq R_2$. If $u_{12} \in U_2$, choose $l = 1$ and $\pi_2(\eta) = u_{12}$ for all η. For all x, we have $\kappa(\eta_{12}, u_{12}, h(x, u_{12})) \preceq \kappa(\eta_2, u_{12}, h(x, u_{12})) = \kappa(F^l(\eta_2, \pi_2, x))$, completing the proof.

(c) Let $\eta_{13} \in \mathcal{I}^{(13)}_{\text{hist}}$, $\eta_{23} \in \mathcal{I}^{(23)}_{\text{hist}}$ and $u_{13} \in U_1 \sqcup U_3$. Assume $\kappa(\eta_{13}) \preceq \kappa(\eta_{23})$. Either $u_{13} \in U_1$ or $u_{13} \in U_3$. If $u_{13} \in U_1$, then because $R_1 \trianglelefteq R_2$ there exist π_{23} and l satisfying the definition for $R_1 \cup R_3 \trianglelefteq R_2 \cup R_3$. If $u_{13} \in U_3$, then choose $l = 1$ and $\pi_{23}(\eta) = u_{13}$ for all η. For all x, we have $\kappa(\eta_{13}, u_{13}, h(x, u_{13})) \preceq \kappa(\eta_{23}, u_{13}, h(x, u_{13})) = \kappa(F^l(\eta_{23}, \pi_{23}, x))$, completing the proof. ■

Corollary 3. If $R_1 \equiv R_2$, then $R_1 \cup R_3 \equiv R_2 \cup R_3$.

Proof. Apply Lemma 2(c) twice. ■

Lemma 2(c) might be misleading. Certainly, hardware components can be made to interact in interesting ways. For example, a control system might combine information from linear and angular odometers to execute circular arc motions. This apparent contradiction results from the definition of robotic primitives, which execute *serially*, rather than in parallel. In this sense, robotic primitives model sensing and actuation strategies as complete "packages", rather than the individual sensors or motors themselves.

Finally, we connect the idea of dominance to the ability of robots to complete tasks.

Lemma 4. (Solution by imitation) Consider two robots R_1 and R_2 with $R_1 \trianglelefteq R_2$ and a preference-closed goal region $\mathcal{I}_G$. If R_1 can reach $\mathcal{I}_G$, then R_2 can reach $\mathcal{I}_G$.

Proof. Use the policy π_2 implied by Definition 4 to complete the task with R_2. ■

This tight connection between dominance and task-completing ability provides some motivation for the form of dominance we propose.

7. Extended Example: Global Localization

In this section we present a detailed example using the definitions of Sections 5 and 6. We consider a *global localization* task, in which the robot has an accurate map of its environment but has no knowledge of its position within that environment. Many forms of the localization problem with varying sensing modalities have been studied in great detail. Some methods (Sugihara 1988; Avis and Imai 1990; Basye and Dean 1990; Cox 1991; Weiss et al. 1994; Guibas et al. 1995; Dellaert et al. 1999; Demaine et al. 2002; Ladd et al. 2004) passively observe the motions of the robot in order to draw conclusions about the robot's state. Others (Kleinberg 1994; Romanik and Schuierer 1996; Dudek et al. 1998; Rao et al. 2004; Koenig et al. 2006; O'Kane and LaValle 2007b) actively drive the robot to reduce uncertainty. The purpose of this example is to show how the results of Section 6 can be used to discover the information requirements of this particular problem in robotics. An analogy can be made to the classification of languages in the theory of computation. It has been shown, for example, that to accept the language of palindromes requires a machine with computation abilities at least as powerful as a pushdown automaton. In this section, we derive similar results regarding the sensing and motion abilities needed to complete the active global localization task.

7.1. Task Definition

Let $E \subseteq \mathbb{R}^2$ denote a planar environment in which a point robot moves. Assume that E is polygonal, bounded, closed and simply connected and that the rotational symmetry group of E is trivial. As in previous examples, the robot's state space is $X = E \times S^1$. We consider a catalog $\mathcal{RP} = \{P_A, P_C, P_T, P_L\}$ of four primitives from Examples 2–4. From these primitives we can form 15 distinct robots. For brevity, we use concatenation to indicate the primitives with which a robot is equipped, so that CT refers to a robot with primitive set $\{P_C, P_T\}$; similar names apply to the other 14 robot models.

Select $\mathcal{I} = \text{pow}(X) - \emptyset$. For κ, use the non-deterministic map defined in Example 1. The initial condition is total uncertainty, so $\kappa(\,) = X$. For the information preference relation, use the definition from Example 9, in which information preference is defined by subset containment. The goal region for the localization task is

$$\mathcal{I}_G = \{\eta \in \mathcal{I} \mid |\eta| = 1\}. \tag{12}$$

That is, we want to command the robot so that only a single final state is consistent with its history I-state. If the robot can complete the task for any E consistent with the assumptions above, we say that the robot can localize itself.

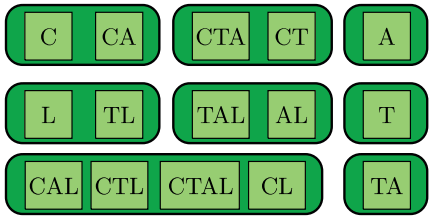

Fig. 8. The 15 robot models grouped into their eight equivalence classes.

7.2. Equivalences and Dominances

Although $\mathcal{RP}$ generates 15 robot models, we can use the results of Section 6 to group them into equivalence classes.

Lemma 5. The following equivalences hold:

(a) CA $\equiv$ C;

(b) CTA $\equiv$ CT;

(c) TL $\equiv$ L;

(d) TAL $\equiv$ AL;

(e) CAL $\equiv$ CTL $\equiv$ CTAL $\equiv$ CL.

The three remaining robot models, A, T and AT, are in singleton equivalence classes.

Proof. (a) Combine Example 12 and Lemma 2(b). (b) Combine Example 12, Lemma 2(b) and Corollary 3. (c) Combine Example 11 and Lemma 2(b). (d) Combine Example 11, Lemma 2(b) and Corollary 3. (e) Combine Examples 11 and 12, Lemma 2(b) and Corollary 3. ■

These equivalences are illustrated in Figure 8. From each, select the unique robot with the fewest primitives and discard the remaining seven robots. We can state several dominances between these classes.

Lemma 6. Between representatives of the equivalence classes from Lemma 5, the following dominances hold:

(a) C $\trianglelefteq$ CT $\trianglelefteq$ CL;

(b) A $\trianglelefteq$ AT $\trianglelefteq$ AL $\trianglelefteq$ CL;

(c) L $\trianglelefteq$ AL $\trianglelefteq$ CL;

(d) T $\trianglelefteq$ AT $\trianglelefteq$ CT $\trianglelefteq$ CL.

Proof. Combine Examples 11 and 12 with Lemma 2(a). ■

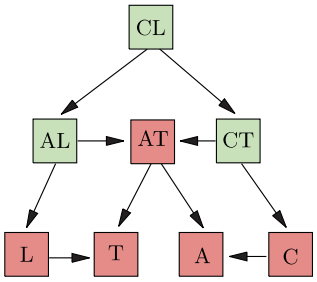

Fig. 9. Classification of robot models under which the localization task can be completed. Shaded models do not admit a solution. Arrows indicate dominances.

7.3. Completing the Localization Task

Which equivalence classes contain robots that can complete the localization task? First, note that some robot models are so simple that we can rule them out immediately.

Lemma 7. None of C, A, L and T can localize themselves.

Proof. For C and A, note that no action changes the robot's position and no observation is influenced by position. Therefore, neither robot can ever gather information about its position. For L and T, note that the robot can never change its orientation. Information available to the robot is limited to the ray extending from its initial state to the nearest obstacle forward. Since E may contain continua of starting states consistent with this information, neither robot can localize itself. ■

Prior results are helpful for the remaining cases.

Lemma 8. (O'Kane and LaValle 2007b) AL and CT can localize themselves but AT cannot.

Finally, we can finish the classification. The results of Lemmas 7–9 are summarized in Figure 9.

Lemma 9. CL can localize itself.

Proof. Combine Lemmas 4 and 8. ■

The result is a complete classification of the solvability of the localization problem over this hierarchy.

8. Extensions and Generalizations

This section contains a series of extensions and generalizations to the techniques presented in Sections 3–6. The intention is to illustrate that, although the preceding results are for a class of highly idealized systems, the general structure of our analysis is useful for a wider variety of problems with greater degrees of realism and generality. We propose methods for dealing with unknown environments (Section 8.1), with sensing and control uncertainty (Section 8.2) and with continuous time (Section 8.3). Although we present each method separately, the extensions are orthogonal in the sense that it is straightforward to apply all of them at once.

8.1. Unknown Environments

In the preceding analysis, we assumed that the robot moves in a fixed, known environment. What happens when the robot begins with limited or no knowledge about its environment, in the sense that positions and geometry of obstacles, map topology, navigability of terrain and so on are unknown? Imperfect knowledge about the environment is a more drastic instance of the general issue of state uncertainty. If the state is defined to include a description of the environment in addition to the robot's configuration, then uncertainty in the environment can be represented as an additional dimension of state uncertainty.

Concretely, choose an *environment space* $\mathcal{E}$ of which each element $E \in \mathcal{E}$ is a potential environment for the robot. Possibilities for $\mathcal{E}$ with varying degrees of realism, interest, practicality and amenability to analysis, include:

1. the set of bounded planar grids with occupancy maps;
2. the set of simple polygons in the plane;
3. the set of compact regions in $\mathbb{R}^2$ or $\mathbb{R}^3$ with connected interiors and piecewise analytic boundaries; and
4. the set of terrain maps from $\mathbb{R}^2$ to $\mathbb{R}$, giving the elevation or navigability at each point in the plane.

The state space is formed by combining the robot's configuration space $\mathcal{C}$ with $\mathcal{E}$, so that $X = \mathcal{C} \times \mathcal{E}$. See Figure 10. In the complete model, the true environment $E \in \mathcal{E}$ affects the robot by influencing the state transitions that the robot makes and the observations that the robot receives. Since the only change is to use a more complicated state space, Definition 4 need not change and the results of Section 6 still hold.

8.2. Imperfect Sensing and Control

We have assumed so far that the robot can execute all of its actions with perfect precision and complete reliability. The motions of real robots are imprecise and unpredictable. Moreover, although we have accounted for the importance of sensing by assuming that the robot is uncertain of its current state and must rely on sensing, we have assumed that sensor readings

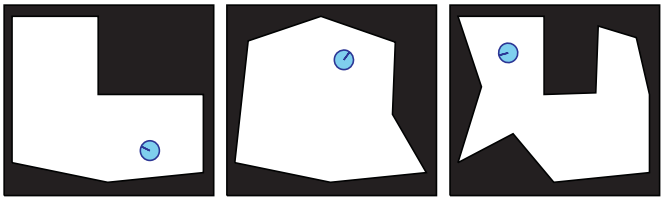

Fig. 10. Three states for an example system containing a mobile robot in the plane with environment uncertainty. When the environment is uncertain, the identity of the environment becomes part of the state of the system.

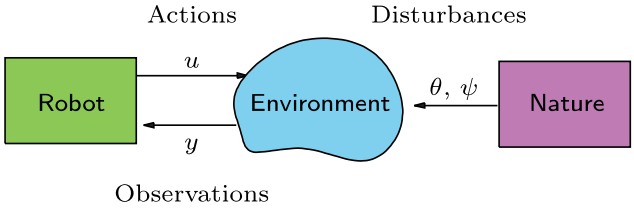

Fig. 11. As the robot interacts with its environment, an artificial decision maker nature generates disturbances.

are uncorrupted by noise. A more realistic sensor model would allow information from sensors to be subject to error.

We propose to follow the approach used in game theory (Blackwell and Girshik 1979; Papadimitriou 1985) and represent this uncertainty by envisioning an abstract external decision maker called "nature". The current state, the action chosen by the robot and the choices made by nature combine to determine how the state changes; given this information, the state trajectory is fully determined. See Figure 11. Formally, define a *nature action space* Θ and augment the state transition function f to depend on nature's choice of $\theta \in \Theta$ at each stage, so that $f : X \times U \times \Theta \to X$. Nature affects the robot's observations in a similar way. Define a *nature observation action space* Ψ and redefine the observation function $h : X \times U \times \Psi \to Y$. The policy application function F must be generalized to account for nature actions, so that

$$\eta_{m+k} = F^m(\eta_k, \pi, x_k, \theta_k, \ldots, \theta_{k+m}, \psi_k, \ldots, \psi_{k+m}). \quad (13)$$

Note that, in contrast to the simpler formulation of (6), the robot's current state, history I-state and policy are no longer sufficient to predict future history I-states.

The following examples illustrate how nature might interfere.

Example 14. Consider a point robot that can move freely in the plane by issuing displacement commands, but whose motion is subject to noise. Let $u_{\max}$ denote a bound on the magnitude of the displacement in each stage and let $\theta_{\max}$ denote a bound on magnitude of the error in this displacement. Let $X = \mathbb{R}^2$, $U = \{u \in \mathbb{R}^2 \mid \|u\| \leq u_{\max}\}$, $\Theta = \{\theta \in \mathbb{R}^2 \mid \|\theta\| \leq \theta_{\max}\}$ and $f(x, u, \theta) = x + u + \theta$. At stage k, the robot can be certain that its state lies within a closed disk of radius $k\theta_{\max}$, centered at the nominal (error-free) final point. See Figure 12.

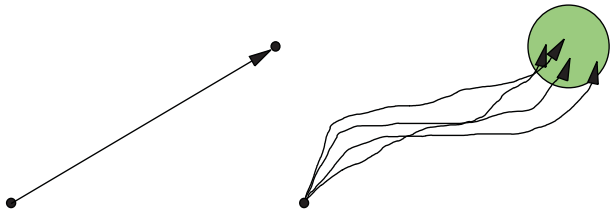

Fig. 12. (a) The robot in Example 14 gives displacement inputs that determine a nominal trajectory. (b) Nature interferes with this motion, but error bounds ensure that the final state is contained in a circle of radius $k\theta_{\max}$.

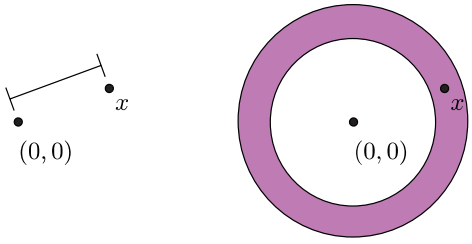

Fig. 13. (a) The robot in Example 15 has a sensor that reports a noisy estimate of the distance to the origin. (b) Accounting for noise bounded by $\psi_{\max}$, the observation confines the robot's state to an annulus of width $2\psi_{\max}$.

Example 15. Suppose that a mobile robot has a sensor that reports the distance to some landmark. Let $X = \mathbb{R}^2$ and $Y = [0, \infty)$. Without loss of generality, position the landmark at the origin. Assume that the sensor has bounded additive error, so that $\Psi = [-\psi_{\max}, \psi_{\max}]$ and $h(x, \psi) = \|x\| + \psi$. See Figure 13. At each stage, the robot knows that its state is within an annulus of width $2\psi_{\max}$, centered at the origin.

In the presence of interference from nature, there are at least two relevant solution concepts:

1. A strategy $\pi : \mathcal{I}_{\text{hist}} \to U$ is a *possible solution* if there exists some stage k and choices of $\theta_1, \ldots, \theta_k$ and $\psi_1, \ldots, \psi_k$ for which the robot reaches a derived I-state $\eta_k \in \mathcal{I}_G$. The robot may reach $\mathcal{I}_G$, but it is also possible that control or sensing errors will prevent it from achieving this goal.

2. A strategy $\pi : \mathcal{I}_{\text{hist}} \to U$ is a *guaranteed solution* if there exists some stage k such that for all choices of

$\theta_1, \ldots, \theta_k$ and $\psi_1, \ldots, \psi_k$, the robot reaches a derived I-state $\eta_k \in \mathcal{I}_G$. The robot can always reach its goal, regardless of any interference by nature.

Other solution concepts, such as those based on performance bounds or on probabilistic guarantees of reaching the goal, are possible but we do not consider them here. In this context, Definition 4 must be generalized to include universal quantifiers over nature's actions.

Definition 5. (Robot dominance with sensing and control error) Consider two robot systems

$$R_1 = (X^{(1)}, U^{(1)}, Y^{(1)}, \Theta^{(1)}, \Psi^{(1)}, f^{(1)}, h^{(1)})$$

and

$$R_2 = (X^{(2)}, U^{(2)}, Y^{(2)}, \Theta^{(2)}, \Psi^{(2)}, f^{(2)}, h^{(2)}).$$

Choose a derived I-space $\mathcal{I}$ and I-maps $\kappa^{(1)} : X^{(1)} \to \mathcal{I}$ and $\kappa^{(2)} : X^{(2)} \to \mathcal{I}$. If, for all:

- $\eta_1 \in \mathcal{I}^{(1)}_{\text{hist}}$;
- $\eta_2 \in \mathcal{I}^{(2)}_{\text{hist}}$ for which $\kappa^{(1)}(\eta_1) \preceq \kappa^{(2)}(\eta_2)$; and
- $u_1 \in U^{(1)}$,

there exists a policy $\pi_2 : \mathcal{I}^{(2)}_{\text{hist}} \to U^{(2)}$ such that for all $x_1 \in X^{(1)}$ consistent with η_1 and all $x_2 \in X^{(2)}$ consistent with η_2, there exists a positive integer l such that for all:

- $\theta_1 \in \Theta^{(1)}$;
- $\psi_1 \in \Psi^{(1)}$;
- $\theta_{2,1}, \ldots, \theta_{2,l} \in \Theta^{(2)}$;
- $\psi_{2,1}, \ldots, \psi_{2,l} \in \psi^{(2)}$;

we have

$$\begin{aligned} & \kappa^{(1)}(\eta_1, u_1, h^{(1)}(x_1, u_1, \psi_1)) \\ \preceq \;& \kappa(F^l(\eta_2, \pi_2, x_2, \theta_{2,1}, \ldots, \theta_{2,l}, \psi_{2,1}, \ldots, \psi_{2,l})) \end{aligned} \quad (14)$$

then R_2 *dominates* R_1 under $\kappa^{(1)}$ and $\kappa^{(2)}$, denoted $R_1 \trianglelefteq R_2$.

The next example demonstrates that Definition 5 behaves reasonably.

Example 16. (Varying error bounds) Recall the incompletely specified models in Examples 14 and 15. Consider two robot systems R_1 and R_2 with state transitions as in Example 14 and observations as in Example 15; R_1 and R_2 differ only in their error bounds $\theta^{(1)}_{\max}$, $\psi^{(1)}_{\max}$, $\theta^{(2)}_{\max}$ and $\psi^{(2)}_{\max}$. We compare these robots under κ_{ndet}. Comparing $\theta^{(1)}_{\max}$ to $\theta^{(2)}_{\max}$, and $\psi^{(1)}_{\max}$ to $\psi^{(2)}_{\max}$, there are four cases:

1. If $\theta^{(1)}_{\max} = \theta^{(2)}_{\max}$ and $\psi^{(1)}_{\max} = \psi^{(2)}_{\max}$, then $R_1 \equiv R_2$.
2. If $\theta^{(1)}_{\max} \leq \theta^{(2)}_{\max}$ and $\psi^{(1)}_{\max} \leq \psi^{(2)}_{\max}$, then $R_2 \trianglelefteq R_1$.
3. If $\theta^{(2)}_{\max} \leq \theta^{(1)}_{\max}$ and $\psi^{(2)}_{\max} \leq \psi^{(1)}_{\max}$, then $R_1 \trianglelefteq R_2$.
4. Otherwise, $R_2 \bowtie R_1$.

These results follow in a straightforward manner from Definition 5. The intuition is that one robot system dominates the other if and only if its error bounds are not larger.

8.3. Continuous Time

The models presented up to this point manage time in discrete stages, in which the robot makes a single decision at each stage. This discretization of time may be unsatisfactory for many kinds of systems, especially those that require complicated control strategies. Continuous-time models have a more direct correspondence with reality. To make the appropriate generalizations, we replace the discrete sequences of states, actions and observations with functions of a continuous-time parameter t.

The state space X, action space U and observation space Y remain unchanged from the discrete stage formulation. At each instant t, the robot chooses some $u(t) \in U$. Let $\widetilde{U}_t$ denote the space of all functions from $[0, t)$ into U and let $\widetilde{U} = \bigcup_{t \in [0,\infty)} \widetilde{U}_t$. For simplicity of notation, adopt the convention that $[0, 0) = \emptyset$. Define $\tilde{u} : [0, \infty) \to U$ as the robot's complete action history and let $\tilde{u}_t \in \widetilde{U}$ denote the robot's action history up to (but exclusive of) time t. We include a special *termination action* $u_T \in U$. The robot selects u_T to indicate that it has finished its task and intends to terminate execution. We require that if $u(t) = u_T$, then $u(t') = u_T$ for all $t' > t$. We describe changes in the state with a *state transition function*

$$\Phi : X \times \bigcup_{t \in [0,\infty)} \widetilde{U}_t \to X. \quad (15)$$

The intuition is that, given a starting state $x(0)$ and an action history $\tilde{u}_t$, the state transition function computes the resulting state

$$x(t) = \Phi(x(0), \tilde{u}_t). \quad (16)$$

This notation of a "black box" state transition function follows notation employed in control theory, for example, by Chen (1984).

Example 17. A familiar special case of (16) occurs if $\tilde{u}$ is a smooth function and there exists a function f such that

$$\Phi(x(0), \tilde{u}_t) = x(0) + \int_0^t f(x(s), u(s))\, ds. \quad (17)$$

In this case, the system dynamics can be described by the differential equation $\dot{x} = f(x, u)$.

As time passes, the robot's sensors provide feedback in the form of observations drawn from an observation space Y. Let $\widetilde{Y}_t$ denote the space of functions mapping $[0, t]$ into Y and let $\widetilde{Y} = \bigcup_{t\in[0,\infty)} \widetilde{Y}_t$. The robot's complete observation history is $\tilde{y} : [0, \infty) \to Y$. The observation history up to t (inclusive) is $\tilde{y}_t \in \widetilde{Y}_t$. The observations received by the robot are governed by the observation function $h : X \to Y$. The history I-state becomes

$$\mathcal{I}_{\text{hist}} = \bigcup_{t\in[0,\infty)} \widetilde{U}_t \times \widetilde{Y}_t, \tag{18}$$

and the history I-state at time t is $\eta(t) = (\tilde{u}_t, \tilde{y}_t) \in \mathcal{I}_{\text{hist}}$. A state x is consistent with an I-state $\eta(t) = (\tilde{u}_t, \tilde{y}_t)$ if and only if there exists some starting state $x(0)$ such that $\Phi(x(0), \tilde{u}_t) = x$ and $h(x(t')) = y(t')$ for $t' < t$.

In our discrete-stage formulation, we used a slightly different observation model, in which $h : X \times U \to Y$. In a continuous-time adaptation, the time period over which observations are available is the half-open interval $[0, t)$; $\tilde{y}_t$ would be undefined at t itself. As a result, the closest we could come to a memoryless strategy is to use the left-hand limit of $\tilde{y}_t$ at t, $\kappa_{\text{obs}}(\eta(t)) = \lim_{t'\to t^-} y(t')$, provided that the limit exists. (Compare with Example 19.) This technicality is part of the motivation for preventing y from depending directly on u, as we have done in this section. A more complete treatment of these kinds of sensor models appears in LaValle (2006, Section 11.1.1). We will not revisit this variation.

We describe the robot's strategy as a feedback policy $\pi : \mathcal{I}_{\text{hist}} \to U$ that specifies an action for each history I-state. We assume that a given strategy is executed until it selects u_T. The time when this occurs, the resulting final state and the observations received along the way are all affected by the strategy π itself and the starting state $x(0)$. Assuming that the robot executes π, the termination time is

$$T(\pi, x(0)) = \inf\{t \in [0, \infty) \mid \pi(\eta(t)) = u_T\}. \tag{19}$$

The final state, denoted $F(\pi, x(0))$, is

$$F(\pi, x(0)) = \Phi(x(0), \tilde{u}_{T(\pi,x(0))}). \tag{20}$$

The next three examples illustrate that feedback over a derived I-space can be a natural way to express familiar kinds of strategies.

Example 18. (Open loop strategy) Let $\mathcal{I}_{\text{time}} = [0, \infty)$ and consider the I-map $\kappa_{\text{time}}(\eta(t)) = t$. In this case, the derived I-state is simply the time elapsed. If the robot has an intended open loop action trajectory $\omega : [0, t_f) \to U$, a strategy to execute γ is $\pi(\eta(t)) = \omega(\kappa_{\text{time}}(\eta(t))$ if $t < t_f$ and $\pi(\eta(t)) = u_T$ otherwise.

Example 19. (Memoryless strategy) Another possibility is that it is enough to know the "most recent" observation, so $\mathcal{I}_{\text{obs}} = Y$ and $\kappa_{\text{obs}}(\eta(t)) = y(t)$. Given a memoryless plan $\gamma : Y \to U$, the composed function $\kappa_{\text{obs}} \circ \gamma : \mathcal{I}_{\text{hist}} \to U$ is a memoryless information feedback strategy.

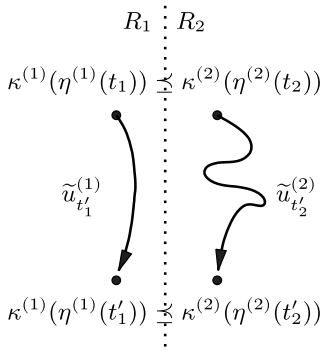

Fig. 14. An illustration of Definition 6. Compare with Figure 6.

Example 20. (Concatenating strategies) Given two strategies π_1 and π_2, a new strategy that concatenates them (that is, executes them in sequence) is expressed by $\pi(\eta(t)) = \pi_1(\eta(t))$ if $\pi_1(\eta(t)) \neq u_T$ and $\pi(\eta(t)) = \pi_2(\eta(t))$ otherwise. By nesting this construction, arbitrarily many strategies can be chained together.

Definition 4 can be generalized in a natural way.

Definition 6. (Robot dominance in continuous time) Consider two continuous-time robot systems $R_1 = (X^{(1)}, U^{(1)}, Y^{(1)}, \Phi^{(1)}, h^{(1)})$ and $R_2 = (X^{(2)}, U^{(2)}, Y^{(2)}, \Phi^{(2)}, h^{(2)})$. If, for all:

- $\eta^{(1)}(t_1) \in \mathcal{I}^{(1)}_{\text{hist}}$;
- $\eta^{(2)}(t_2) \in \mathcal{I}^{(2)}_{\text{hist}}$ for which $\kappa^{(1)}(\eta^{(1)}(t_1)) \preceq \kappa^{(2)}(\eta^{(2)}(t_2))$;
- $t'_1 \in [0, \infty)$; and
- $\tilde{u}^{(1)}_{t'_1} \in \widetilde{U}^{(1)}_{t'_1}$;

there exists an information feedback strategy $\pi_2 : \mathcal{I}^{(2)}_{\text{hist}} \to U^{(2)}$, such that for all $x^{(1)} \in X^{(1)}$ consistent with $\eta^{(1)}(t_1)$ and $x^{(2)} \in X^{(2)}$ consistent with $\eta^{(2)}(t_2)$, there exists $t'_2 \in [0, \infty)$ such that if R_1 executes $\tilde{u}^{(1)}_{t'_1}$ from time t_1 to t'_1 and R_2 executes $\pi^{(2)}$ from time t_2 to t'_2, we have

$$\kappa^{(1)}(\eta^{(1)}(t'_1)) \preceq \kappa^{(2)}(\eta^{(2)}(t'_2)), \tag{21}$$

then R_2 *dominates* R_1 under $\kappa^{(1)}$ and $\kappa^{(2)}$, denoted $R_1 \trianglelefteq R_2$.

See Figure 14. The next two examples illustrate some implications of Definition 6.

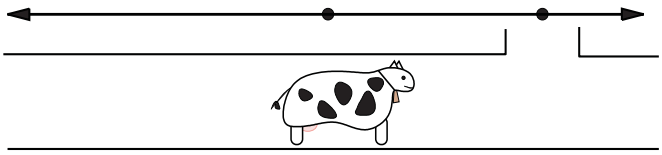

Fig. 15. The lost cow of Example 22 searching for a gate.

Example 21. (Omniscient sensing and perfect control) Consider a degenerate case with $Y = X$ and $h(x) = x$. This situation gives the robot complete information about its state. Choose $\mathcal{I} = X$ and $\kappa(\eta(t)) = y(t) = x(t)$. Let $\kappa(\eta_1) \preceq \kappa(\eta_2)$ if and only if $\kappa(\eta_1) = \kappa(\eta_2)$, as in Example 8. In this context, Definition 4 becomes a statement about the regions of state space reachable by different control systems.

Suppose that three such systems R_1, R_2 and R_3 differ only in their action spaces $U^{(1)}$, $U^{(2)}$ and $U^{(3)}$. Let $Z(A)$ denote the subset of state space reachable by a robot with action space A. Suppose $R_1 \trianglelefteq R_2$. The system R_3 need not be comparable to either R_1 or R_2. Note that additional robot models can be constructed from unions of $U^{(1)}$, $U^{(2)}$ and $U^{(3)}$. We have the following results:

$$Z(U^{(1)}) \subseteq Z(U^{(2)} \cup U^{(3)}) \quad (22)$$

$$Z(U^{(1)}) = Z(U^{(1)} \cup U^{(2)}) \quad (23)$$

$$Z(U^{(1)} \cup U^{(3)}) \subseteq Z(U^{(2)} \cup U^{(3)}) \quad (24)$$

These results are analogous to Lemma 2. Note that in combining action spaces in this way, we allow the robot to choose *sequentially* the action set from which to choose its action. The results fail if the robot is somehow allowed to choose actions from each constituent set in parallel.

Example 22. (A lost cow) A well-known problem in online algorithms is the *lost cow problem* (Baeza-Yates et al. 1993; Kao et al. 1993) in which a near-sighted cow moves along a fence searching for a gate, as illustrated in Figure 15. The difficulty under the standard sensing model is that the cow must systematically search in both directions from its initial position without any information about the distance or direction to the gate. The interest in this problem derives from potential applications in (or at least the potential for better understanding of) exploration in unbounded environments.

We formulate the lost cow problem and consider how the sensing model affects the cow's searching ability. Let $X = \mathbb{R}$, in which $x(t)$ is the position of the gate relative to the cow at time t. Let the action space be $U = [-1, 1]$ with $\Phi(x(0), \tilde{u}_t) = x(0) + \int_0^t u(s)\, ds$. We compare three distinct models C_1, C_2 and C_3 under κ_{ndet}:

C_1: Let $Y^{(1)} = \mathbb{R}$ and $h^{(1)}(x) = x$. Here the cow can determine both the direction and distance to the gate.

C_2: Let $Y^{(2)} = \{-1, 0, 1\}$ and $h(x) = \text{sign}(x)$. This allows the cow to determine the direction it must move to reach the gate, but not the distance.

C_3: Let $Y^{(3)} = \{0, 1\}$ and $h^{(2)}(x) = 1$ if $x = 0$ and $h^{(2)}(x) = 1$ otherwise. This is the standard lost cow sensing model, in which the cow cannot see the gate from a distance, but can detect the gate when it arrives.

Perhaps surprisingly, these three models are equivalent in the sense of Definition 6. This is because each can eventually determine its state (by finding the gate) and after the state is known, state uncertainty cannot recur. To simulate C_1 with C_3, first execute the algorithm of Baeza-Yates et al. (1993), then move to the state occupied by C_1.

We conclude our discussion of continuous-time models by showing how a discrete stage model in the form of Section 3 can be constructed from a continuous-time model in the form presented above. Consider a division of time into variable length stages, in which, in each stage, the robot executes a single information feedback strategy to completion. We require of each of these strategies the following special property.

Definition 7 (History invariance) If, for all $\eta(t) \in \mathcal{I}_{\text{hist}}$, all $x \in X$ consistent with $\eta(t)$, and all $y(0) \in Y$, we have $F(\pi, x, \eta(t)) = F(\pi, x, \eta(0))$, then π is a *history-invariant* strategy.

The intuition of the definition is that the robot executing π is free to use the observation and action history generated during its own execution, but it cannot peer into the past before its execution began in order to make decisions. Given a continuous-time robot system $R = (X, U, Y, \Phi, h)$ (as defined in this section) and a set Π of history-invariant information feedback strategies, construct a discrete-stage system (as in Section 3) $\bar{R} = (X, \bar{U}, \bar{Y}, \bar{f}, \bar{h})$ as follows:

1. The state space X is the same.
2. The action space is $\bar{U} = \Pi$.
3. The observation space is $\bar{Y} = \widetilde{Y}$.
4. The state transition function is $f : X \times \bar{U} \to X$, with $f(x, \pi) = F(\pi, x, \eta(0))$.
5. The observation function is $h : X \times \bar{U} \to \bar{Y}$.

The system starts at some (unknown) initial state $x_1 \in X$. Let $x_k \in X$, $u_k \in \bar{U}$ and $y_k \in \bar{Y}$ denote the appropriate values at stage k. These sequences are related to each other by $x_{k+1} = f(x_k, u_k)$ and $y_k = h(x_k, u_k)$. The history I-state consists of the action and observation histories: $\eta_k = (u_1, y_1, \ldots, u_{K-1}, y_{K-1})$. This construction gives a discrete-stage system faithful to the dynamics in both state space and I-space of the underlying continuous-time system.

Lemma 10. Any action sequence $u_1, \ldots, u_K$ executed by $\bar{R}$ reaches the same final state x and the analogous final history I-state as does R.

Note, however, that in making this transformation, we must choose a set Π of strategies and may therefore restrict the space of plans that the robot can execute. If Π does not contain a sufficiently rich selection of information feedback strategies, there may be regions of I-space that are no longer reachable under the discretized model. In this way, Π is analogous to the catalog of robotic primitives $\mathcal{RP}$ introduced in Section 4.

9. Discussion

The results of this paper are intended to lay a foundation for a sensor-centered theory for comparing robotic problems and systems. Great potential exists to build on this foundation, particularly by developing the analogy to the theory of computation even further.

The most obvious avenue for future work is to study a broader collection of problems. Although this paper considers an active global localization problem in detail, other fundamental robotics problems warrant similar analysis of their information requirements. For example, results exist for limited-sensing versions of navigation (Lumelsky and Tiwari 1994; Lumelsky and Stepanov 1987; Papadimitriou and Yannakakis 1991; Kamon et al. 1999), exploration (Dúnlaing and Yap 1982; Choset and Burdick 1995; Acar and Choset 2001a, Tovar et al. 2004) and manipulation (Akella and Mason 1998; Erdmann and Mason 1988; Goldberg 1993; Akella et al. 2000; Agarwal et al. 2001) tasks. Using the techniques we have presented, it should be possible to unify and extend these results to develop a more complete understanding of the sensing and motion abilities needed to solve these problems. Other problems and more complex sensing systems could also be investigated.

One of the most powerful ideas in the theory of computation that we have not explored here is the idea of *reductions*, which hold promise for comparing robotic problems themselves. The resulting statements would have the form "Problem A is at least as hard as Problem B". To make things more concrete, we might consider *decision problems*, in which the robot with a state space defined as in Section 8.1 must determine whether its environment $E \in \mathcal{E}$ has a certain property. Such problems can be expressed naturally as planning problems in I-space. To decide whether E has a property $\Xi : \mathcal{E} \to \{0, 1\}$, the robot must reach the goal region

$$\mathcal{I}_{G,\Xi} = \{\eta \in \mathcal{I}_{\text{hist}} \mid \forall (q, E) \in \kappa_{\text{ndet}}(\eta), \Xi(E) = 1\}$$
$$\cup \{\eta \in \mathcal{I}_{\text{hist}} \mid \forall (q, E) \in \kappa_{\text{ndet}}(\eta), \Xi(E) = 0\}. \quad (25)$$

An example is in Figure 16.

Another direction is to study the computational requirements of robotics problems. We expect that there exist rich tradeoffs between computation time, memory usage, sensing requirements and solution quality. There has been some research on certain tasks, for example exploration (Shannon 1952; Blum and Kozen 1978; Bender et al. 1998), pursuit-evasion (Yamashita et al. 2006) and coverage (Wagner et al. 1999), but very little is known in general. One way to deal with such issues is to study *sufficient I-maps* (LaValle 2006), which are I-maps for which transitions can be computed directly in the derived I-space, allowing the history to be discarded. For example, if a problem can be solved under a given robot model using a sufficient I-map into a derived I-space of finite cardinality n, the memory required to solve the problem is $O(\log n)$. The results of Blum and Kozen (1978), for example, can be characterized as showing how a discrete exploration problem can be solved in a derived I-space with cardinality linear in the height of the area to be explored, meaning that only logarithmic memory is required. These computational issues must be approached with care, especially if those computations involve real numbers (Blum et al. 1998).

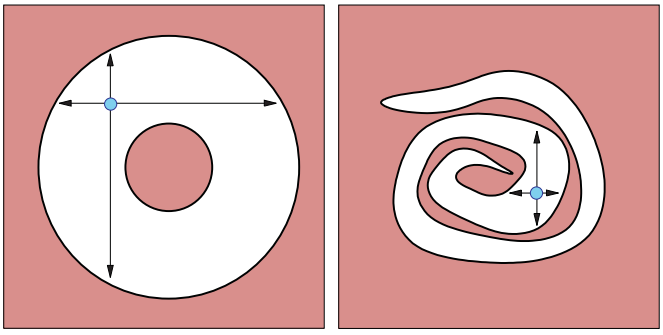

Fig. 16. A sample decision problem. What sensing is required to decide if a planar environment is simply connected? What robots can distinguish between (a) the annulus environment and (b) the helix?

In spite of these possibilities, there are important limitations to the idealized models we presented. Of the many issues remaining to be addressed, we mention a few here.

9.1. Probabilistic Uncertainty

We have focused our attention on non-deterministic uncertainty, but a large subset of contemporary work in robotics uses probabilistic models of uncertainty (Simmons and Koenig 1995; Shatkay and Kaelbling 1997; Fox et al. 1998; Austin and Jensfelt 2000; Jensfelt and Kristensen 2001; Thrun et al. 2005). Our results also apply, at least in principle, to probabilistic uncertainty. In this context, the relevant derived I-space is a space of probability distributions over X. However, it is not immediately clear what the "right" information preference relation over such a space would be. Depending on the models

used, it may also be necessary to relax Definition 4 to require only that R_2 can simulate R_1 with sufficiently high probability. More generally, the differences between non-deterministic and probabilistic uncertainty models warrant further exploration. For example, non-deterministic uncertainty has the property that sensing can only help. Actions from primitives such as P_R (Example 6) or P_G (Example 7) that do not change the state always lead to a derived I-state at least as good as the current one. Under probabilistic uncertainty, this property does not obviously hold, and sensing can sometimes increase uncertainty.

9.2. Selecting the Catalog of Primitives

Although we believe that our robotic primitives provide a useful abstraction, any results derived using our methods are only meaningful if $\mathcal{RP}$ is diverse enough to faithfully represent the underlying system. It remains an open problem to systematically find small (or at least succinctly described) sets of robotic primitives that are complete (or nearly complete) in the sense of not eliminating any reachable regions in I-space. There is, however, active interest in related problems for control systems (Frazzoli 2001; Girard and Pappas 2006; Mehta et al. 2006; Murphey 2006).

What happens if $\mathcal{RP}$ is not a finite set? For example, we may extend P_L (from Example 5) to a family $\{P_{L,\epsilon} = (S^1, \{0\}, f_{L_\epsilon}, h_{L_\epsilon}) \mid \epsilon \geq 0\}$ of primitives, each using a noisy linear odometer whose error is bounded by ϵ. If $\mathcal{RP}$ contains many such families of primitives and we assume that each robot has at most one primitive from each family, then the space of robot models is a cube in $\mathbb{R}^n$. The problem of identifying the region in which a given task can be solved is correspondingly more difficult.

9.3. Efficiency and Optimality

Throughout this paper, we have neglected the question of the robot's efficiency in completing its tasks. This weakness is particularly evident, for example, in Example 10, in which the statement of dominance does not consider the differences in execution cost, which in this case are likely to be prohibitively large. One established technique for taking such costs into account is to use *competitive ratios* (Icking et al. 2002; Gabriely and Rimon 2004), which compare the execution costs of online algorithms (which must gather information during their execution) to offline methods (which have complete information) for the same tasks. It may be fruitful to generalize this notion by considering "relative competitive ratios" that bound the additional cost accrued by replacing one robot system with another dominant robot system.

9.4. Parameterization of Time

In Section 8.3, we parameterized the robot's observations by time. In doing so, we implicitly assumed that the robot has an accurate clock. Although such an assumption is generally not technologically impractical, it requires care in abstract models to ensure that the robot cannot acquire extra information "for free". A robot might, for example, use this implicit clock to parlay an accurate velocity sensor into a perfect odometer. One solution is to express $\tilde{u}$ and $\tilde{y}$ as functions of some other abstract parameter p. To recover the original functions of time, the robot must determine a hidden mapping from $\mathbb{R}$ to $\mathbb{R}$ under which p maps to t. Such issues are addressed in detail by LaValle and Egerstedt (2007).

9.5. Cooperation and Coordination

In this work we have considered only a single independent robot. We might also consider the performance of teams of cooperative robots on the same tasks. Such work would require an investigation of the joint I-spaces that would arise from the interaction of multiple agents, each having only limited information. In particular, limited and possibly noisy communication between robots must be modeled. Many of the relevant issues are worked out in the game theory literature (Owen 1982; Başar and Olsder 1995).

Acknowledgments

This work is supported by ONR Grant N00014-02-1-0488 and DARPA grants #HR0011-05-1-0008 and #HR0011-07-1-0002.

References

Abate, A., Ames, A. D. and Sastry, S. (2006). Error-bounds based stochastic approximations and simulations of hybrid dynamical systems. *Proceedings of the American Control Conference*.

Acar, E. U. and Choset, H. (2001a). Complete sensor-based coverage with extended-range detectors: a hierarchical decomposition in terms of critical points and Voronoi diagrams. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Acar, E. U. and Choset, H. (2001b). Robust sensor-based coverage of unstructured environments. *Procedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Agarwal, P. K., Collins, A. D. and Harer, J. L. (2001). Minimal trap design. *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2243–2248.

Akella, S., Huang, W., Lynch, K. M. and Mason, M. T. (2000). Parts feeding on a conveyor with a one joint robot. *Algorithmica*, **26**(3): 313–344.

Akella, S. and Mason, M. (1998). Posing polygonal objects in the plane by pushing. *International Journal of Robotics Research*, **17**(1): 70–88.

Austin, D. J. and Jensfelt, P. (2000). Using multiple gaussian hypotheses to represent probability distributions for mobile robot localization. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1036–1041.

Avis, D. and Imai, H. (1990). Locating a robot with angle measurements. *Journal of Symbolic Computation*, **10**(3–4): 311–326.

Başar, T. and Olsder, G. J. (1995). *Dynamic Noncooperative Game Theory*, 2nd edn. London, Academic Press.

Baeza-Yates, R. A., Culberson, J. C. and Rawlins, G. J. E. (1993). Searching in the plane. *Information and Computation*, **106**: 234–252.

Barraquand, J. and Ferbach, P. (1995). Motion planning with uncertainty: the information space approach. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1341–1348.

Basye, K. and Dean, T. (1990) Map learning with indistinguishable locations. *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. Amsterdam, North-Holland, pp. 331–342.

Bender, M. A., Fernández, A., Ron, D., Sahai, A. and Vadhan, S. (1998). The power of a pebble: exploring and mapping directed graphs. *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pp. 269–278.

Bertsekas, D. P. (2001). *Dynamic Programming and Optimal Control*, vol. 1, 2nd edn. Belmont, MA, Athena Scientic.

Blackwell, D. and Girshik, M. A. (1979). *Theory of Games and Statisitical Decisions*. New York, Dover.

Blum, L., Cucker, F., Shub, M. and Smale, S. (1998). *Complexity and Real Computation*. Berlin, Springer.

Blum, M. and Kozen, D. (1978). On the power of the compass (or, why mazes are easier to search than graphs). *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pp. 132–142.

Borodin, A. and El-Yaniv, R. (1998). *Online Computation and Competitive Analysis*. Cambridge, Cambridge University Press.

Brafman, R. I., Halpern, J. Y. and Shoham, Y. (1998). On the knowledge requirements of tasks. *Artificial Intelligence*, **98**(1–2): 317–349.

Chen, C.-T. (1984). *Linear System Theory and Design*. New York, Holt, Rinehart and Winston.

Choset, H. and Burdick, J. (1995). Sensor based planning, part I: the generalized Voronoi graph. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1649–1655.

Cox, I. J. (1991) Blanche—an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transations on Robotics and Automation*, **7**(2): 193–204.

Dellaert, F., Fox, D., Burgard, W. and Thrun, S. (1999). Monte Carlo localization for mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*.

Demaine, E. D., López-Ortiz, A. and Munro, J. I. (2002). Robot localization without depth perception. *Proceedings of the Scandinavian Workshop on Algorithm Theory*.

Donald, B. R. (1995). On information invariants in robotics. *Artificial Intelligence*, **72**(1–2): 217–304.

Dudek, G., Romanik, K. and Whitesides, S. (1998). Localizing a robot with minimum travel. *SIAM Journal on Computing*, **27**(2): 583–604.

Egerstedt, M. (2002). Motion description languages for multimodal control in robotics. *Control Problems in Robotics* (**Springer Tracts in Advanced Robotics**), Bicchi, A., Cristensen, H. and Prattichizzo, D. (eds). Berlin, Springer, pp. 75–90.

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

Erdmann, M. (1995). Understanding action and sensing by designing action-based sensors. *International Journal of Robotics Research*, **14**(5): 483–509.

Erdmann, M. and Mason, M. T. (1988). An exploration of sensorless manipulation. *IEEE Transations on Robotics and Automation*, **4**(4): 369–379.

Fox, D., Burgard, W. and Thrun, S. (1998). Active markov localization for mobile robots. *Robotics and Autonomous Systems*, **25**: 195–207.

Frazzoli, E. (2001). Robust hybrid control of autonomous vehicle motion planning. *Ph.D. Thesis*, Cambridge, MA, Massachusetts Institute of Technology, June.

Gabriely, Y. and Rimon, E. (2004). Competitive complexity of mobile robot on line motion planning problems. *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*.

Girard, A. and Pappas, G. J. (2006). Hierarchical control using approximate simulation relations. *Proceedings of the IEEE Conference on Decision and Control*.

Girard, A. and Pappas, G. J. (2007). Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, **52**(2): 782–798.

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

Goldberg, K. Y. and Mason, M. T. (1990) Bayesian grasping. *Proceedings of the IEEE International Conference on Robotics and Automation*.

Guibas, L. J., Motwani, R. and Raghavan, P. (1995). The robot localization problem. *Proceedings of the Workshop on*

the Algorithmic Foundations of Robotics, Goldberg, K., Halperin, D., Latombe, J.-C. and Wilson, R. (eds). Wellesley, MA, A. K. Peters, pp. 269–282.

Hopcroft, J. E., Ullman, J. D. and Motwani, R. (2000). *Introduction to Automata Theory, Languages, and Computation*, 2nd edn. Reading, MA, Addison-Wesley.

Icking, C., Kamphans, T., Klein, R. and Langetepe, E. (2002). On the competitive complexity of navigation tasks. *Proceedings of Sensor Based Intelligent Robots*, pp. 245–258.

Jensfelt, P. and Kristensen, S. (2001). Active global localisation for a mobile robot using multiple hypothesis tracking. *IEEE Transations on Robotics and Automation*, **17**(5): 748–760.

Kamon, I., Rivlin, E. and Rimon, E. (1999). Range-sensor based navigation in three dimensions. *Proceedings of the IEEE International Conference on Robotics and Automation*.

Kao, M.-Y., Reif, J. H. and Tate, S. R. (1993). Searching in an unknown environment: an optimal randomized algorithm for the cow-path problem. *Proceedings of the ACM–SIAM Symposium on Discrete Algorithms*, pp. 441–447.

Karp, R. M. On-line algorithms versus off-line algorithms: how much is it worth to know the future? *Proceedings of the World Computer Congress*.

Kleinberg, J. M. (1994). The localization problem for mobile robots. *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pp. 521–531.

Koenig, S., Mudgal, A. and Tovey, C. (2006). An approximation algorithm for the robot localization problem. *Proceedings of the ACM–SIAM Symposium on Discrete Algorithms*.

Kuhn, H. W. (1953). Extensive games and the problem of information. *Contributions to the Theory of Games*, Kuhn, H. W. and Tucker, A. W. (eds) Princeton, NJ, Princeton University Press, pp. 196–216.

Kutulakos, K. N., Dyer, C. R. and Lumelsky, V. J. (1994). Provable strategies for vision-guided exploration in three dimensions. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1365–1371.

Ladd, A. M., Bekris, K. E., Rudys, A. P., Wallach, D. S. and Kavraki, L. E. (2004). On the feasibility of using wireless Ethernet for indoor localization. *IEEE Transactions on Robotics and Automation*, **20**(3): 555–559.

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge, Cambridge University Press. Also available at http://planning.cs.uiuc.edu/.

LaValle, S. M. and Egerstedt, M. B. (2007). On time: Clocks, chronometers, and open-loop control. *Technical Report UIUCDCS-R-2007-2861*, University of Illinois at Urbana-Champaign, May.

LaValle, S. M. and Hutchinson, S. A. (1998). An objective-based framework for motion planning under sensing and control uncertainties. *International Journal of Robotics Research*, **17**(1): 19–42.

Lenser, S. and Veloso, M. (2000). Sensor resetting localization for poorly modelled mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*.

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

Lumelsky, V. and Tiwari, S. (1994). An algorithm for maze searching with azimuth input. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 111–116.

Lumelsky, V. J. and Stepanov, A. A. (1987) Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, **2**: 403–430.

Mehta, T., Delmotte, F. and Egerstedt, M. (2006). Motion alphabet augmentation based on past experiences. *Proceedings of the IEEE Conference on Decision and Control*.

Moll, M. and Erdmann, M. (2002) Manipulation of pose distributions. *International Journal of Robotics Research*, **21**(3): 277–292.

Murphey, T. (2006). Motion planning for kinematically overconstrained vehicles using feedback primitives. *Proceedings of the IEEE International Conference on Robotics and Automation*.

Dúnlaing, C. Ó. and Yap, C. K. (1982). A retraction method for planning the motion of a disc. *Journal of Algorithms*, **6**: 104–111.

O'Kane, J. M. and LaValle, S. M. (2006). On comparing the power of mobile robots. *Proceedings of Robotics: Science and Systems*.

O'Kane, J. M. and LaValle, S. M. (2007a). Sloppy motors, flaky sensors, and virtual dirt: comparing imperfect ill-informed robots. *Proceedings of the IEEE International Conference on Robotics and Automation*.

O'Kane, J. M. and LaValle, S. M. (2007b). Localization with limited sensing. *IEEE Transations on Robotics*, **23**(4): 704–716.

Owen, G. (1982) *Game Theory*. New York, Academic.

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

Papadimitriou, C. H. and Yannakakis, M. (1991). Shortest paths without a map. *Theoretical Computer Science*, **84**: 127–150.

Rao, M., Dudek, G. and Whitesides, S. (2004). Randomized algorithms for minimum distance localization. *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, pp. 265–280.

Romanik, K. and Schuierer, S. (1996). Optimal robot localization in trees. *Proceedings of the Symposium on Computational Geometry*, pp. 264–273.

Shannon, C. E. (1952). Presentation of a maze solving machine. *Cybernetics: Circular, Casual, and Feedback Mechanisms in Biological and Social Systems, Transactions*

Eighth Conference, von Foerster, H., Mead, M. and Teuber, H. L. (eds). New York, Josiah Macy Jr. Foundation, pp. 169–181.

Shatkay, H. and Kaelbling, L. P. (1997). Learning topological maps with weak local odometric information. *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 920–927.

Simmons, R. and Koenig, S. (1995). Probabilistic robot navigation in partially observable environments. *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1080–1087.

Sipser, M. (1997). *Introduction to the Theory of Computation*. Boston, MA, PWS.

Sleator, D. D. and Tarjan, R. E. (1985). Amortized efficiency of list update and paging rules. *Communications of the ACM*, **28**: 202–208.

Soare, R. I. (1987). *Recursively Enumerable Sets and Degrees*. Berlin, Springer.

Sugihara, K. Some location problems for robot navigation using a simple camera. *Computer Vision, Graphics, and Image Processing*, **42**(1): 112–129.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics*. Cambridge, MA, MIT Press.

Tovar, B., Guilamo, L. and LaValle, S. M. (2004). Gap Navigation Trees: minimal representation for visibility-based tasks. *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*.

van der Stappen, A. F., Berretty, R.-P., Goldberg, K. and Overmars, M. H. (2000). Geometry and part feeding. *Proceedings of Sensor Based Intelligent Robots*, pp. 259–281.

Wagner, I. A., Lindenbaum, M. and Bruckstein, A. M. (1999). Distributed covering by ant-robots using evaporating traces. *IEEE Transations on Robotics and Automation*, **15**(5): 918–933.

Weiss, G., Wetzler, C. and von Puttkamer, E. (1994). Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Whitney, D. E. (1986) Real robots don't need jigs. *Proceedings of the IEEE International Conference on Robotics and Automation*.

Wiegley, J., Goldberg, K., Peshkin, M. and Brokowski, M. (1997). A complete algorithm for designing passive fences to orient parts. *Assembly Automation*, **17**(2): 129–136.

Kameda, T., Yamashita, M. and Suzuki, I. (2006). Online polygon search by a seven-state boundary 1-searcher. *IEEE Transations on Robotics*, **22**(3): 446–460.