Sloppy motors, flaky sensors, and virtual dirt: Comparing imperfect ill-informed robots

Jason M. O'Kane and Steven M. LaValle

***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" 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. The basic idea is a dominance relation over robot systems that formalizes the idea that some robots are stronger than others. We show that this definition is directly related to the robots' ability to complete tasks. Our prior work in this area assumes perfect control and sensing, requires that the robot begin with a single fixed initial condition within a known environment, and models of time as a sequence of variable-length discrete stages, rather than as a continuum. In this paper, we substantially improve upon that earlier work by addressing these problems.**

I. INTRODUCTION

Suppose 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 can be (and 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 in answering 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.

A. Robots, sensors, and the theory of computation

This work is inspired in part by the theory of computation, which begins with precisely defined models of abstract machines, such as finite automata, pushdown automata, Turing machines, and so on [14]. 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 several kinds of basic questions about these machines and problems.

1) *Solvability*: Can a given machine can 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 can finite automata. 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 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. 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.

Research that studies the competitive ratios of online methods [16], [25], [26] is a step in the right direction. This work is useful for understanding how the quality of optimal solutions is affected by sensing complexity. However, online algorithms generally are concerned only with optimality, rather than feasibility. Moreover, research in this area generally does not consider imperfect sensing or control.

The aim of this paper is to develop a "sensor-centered" theory for analyzing and comparing robot systems. Our contribution is to develop such a theory more completely than in prior work and to illustrate its usefulness with examples.

B. Central Ideas

This section is an overview of the main ideas underlying our approach.

This work is supported by ONR Grant N00014-02-1-0488 and by the DARPA SToMP program. J. M. O'Kane and S. M. LaValle are with the Department of Computer Science, University of Illinois at Urbana-Champaign, 201 North Goodwin Avenue, Urbana, IL 61801, USA. {jokane, lavalle}@cs.uiuc.edu

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. Then the robot can treat this estimated state as its true state and ignore the uncertainty. We, however, are interested in a broader class of tasks for which accurate state estimation is impossible. Mason [22] gives an example of sweeping a floor on which dirt cannot be detected; the robot must sweep "virtual dirt" to ensure that the floor is eventually clean.

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 [17], but have been used in robotics for some time [2], [10], [13], [19]. The recent book by LaValle [18] explains and develops the idea thoroughly.

Of course, sensor data in its raw form is frequently uninformative. This motivates our study of *information mappings* into *derived information spaces*. Such a mapping could be based on probabilistic or nondeterministic models of the robot's sensors and environment. Other more drastic or problem-specific mappings are also possible. The choice of an information mapping determines the conclusions the robot can make about its state and indirectly the problems the robot can solve. Consequently, the choice of an information mapping and a derived information space are crucial modeling decisions.

2) Games against nature: Realistic models of robot motion and sensing include the possibility of errors and uncertainty. We follow the approach used in game theory [4], [24] 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.

3) Robot dominance: The centerpiece of this work is a preorder over robot systems. The intuition of this *dominance relation* is that one robot dominates another if and only if it can "simulate" the other in a certain way. The definition is based on a partial order, an *information preference relation* over information space that indicates which information states are "more informed" than 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 that can potentially defy comparison.

C. Organization

Section II is a brief survey of related work. In Section III we give a basic problem definition. Our definition of robot dominance and its properties are in Section IV. Section V relates the continuous-time model we introduce in this paper to our prior work that models time as a sequence of discrete stages. We make concluding remarks and discuss open problems in Section VI. We illustrate with examples throughout.

II. RELATED WORK

We partially address issues issues of robot comparison and dominance in prior work [23], in which we establish a *dominance* relation over robot systems. Although it was intended as a preliminary step toward a general theory of robots and sensors, that work has several important shortcomings that limit its applicability.

1) *Perfect control* – In [23], we assumed that the robot can execute all of its actions with perfect precision and complete reliability. The motions of real robots are imprecise and unpredictable.
2) *Perfect sensing* – Although [23] accounts for the importance of sensing by assuming that the robot is uncertain of its current state and must rely on sensing, it assumes that sensor readings are uncorrupted by noise. A more realistic sensor model would allow information from sensors to be subject to error.
3) *Modeling of time* – In [23], time is managed in discrete *stages*. 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.
4) *Fixed, known environment* – In [23], we assumed (tacitly) that the robot operates in a fixed, known environment. This assumption, which stems from the formulation of the current state as the robot's configuration within its environment, is unsatisfactory in all but the most structured contexts.
5) *Identical state spaces* – The dominance relation in [23] is only able to compare robots that share the same state space. To compare robots that are truly dissimilar, the framework must allow each robot to have a distinct state space.

In this paper, we present substantial revisions and extensions to the framework of [23] to remedy these shortcomings. These extensions illuminate several issues and subtleties not evident in the former paper.

Our goals are similar to those of Donald [8]. 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 for which sensing is a crucial issue, the information space is the space in which the problem can most naturally be posed.

A third line of related research is the work of Erdmann [11], which is itself grounded in the preimage planning ideas due Lozano-Perez, Mason, and Taylor [20]. In Erdmann's work, sensors are modeled by giving a partition of state space. The problem of sensor design is choose a partition so that from each region in the partition, the robot knows

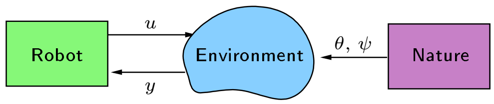

Fig. 1. As the robot interacts with its environment, an artificial decision maker nature generates disturbances.

what action to select in order to make progress toward its goal.

Others in artificial intelligence [6] and control theory [9], [12] have addressed related issues.

III. BASIC DEFINITIONS

This section contains basic definitions for planning with uncertainty in the robot's current state. In summary, the robot lives in some state space, beginning at an unknown start state and choosing actions that change the current state in some possibly noisy way. The current state is hidden from the robot, which must rely instead on observations that give incomplete and possibly noisy "hints" about the true state. The noise in control and sensing is generated by a fictional external decision maker we call *nature*, for which we assume some behavior model is known. See Figure 1. Details of the formulation follow.

A. State spaces and environment spaces

The robot moves in a state space X, which must be sufficiently expressive to encode all of the relevant information about the condition of the world. In a simple case, X might be defined as the configuration space [21] of the robot in a certain environment. Time proceeds continuously starting at $t = 0$ and continuing indefinitely. The robot's state at time t is denoted $x(t)$.

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, and
3) the set of compact regions in $\mathbb{R}^2$ or $\mathbb{R}^3$ with connected interiors and piecewise analytic boundaries.
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 $X = \mathcal{C} \times \mathcal{E}$. In our models, 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.

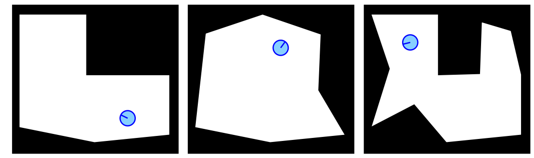

Fig. 2. When the environment is uncertain, the identity of the environment becomes part of the state of the system. Above are three states for an example system containing a mobile robot in the plane with environment uncertainty.

B. Actions and transitions

The robot influences its current state by choosing actions from some action space U. 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 $\widetilde{u} : [0, \infty) \to U$ as the robot's complete action history, and let $\widetilde{u}_t \in \widetilde{U}$ 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 $t' > t$.

How do these actions influence the state? Recall that we intend to model disturbances and unexpected events as interference from nature. Choices made by both the robot and by nature affect changes in the state. Let Θ denote a *nature action space*. Let $\widetilde{\Theta}_t$ denote the space of all functions mapping $[0, t)$ into Θ, and let $\widetilde{\Theta} = \bigcup_{t \in [0,\infty)} \widetilde{\Theta}_t$. Let $\widetilde{\theta} : [0, \infty) \to \Theta$ denote the complete history of nature actions and $\widetilde{\theta}_t \in \widetilde{\Theta}_t$ the nature action history up to (and including) t.

We describe changes in the state with a *state transition function*

$$\Phi : X \times \bigcup_{t \in [0,\infty)} (\widetilde{U}_t \times \widetilde{\Theta}_t) \to X. \tag{1}$$

The intuition is that, given a starting state $x(0)$, and action histories $\widetilde{u}_t$ and $\widetilde{\theta}_t$ of equal duration for the robot and nature respectively, the state transition function computes the resulting state

$$x(t) = \Phi(x(0), \widetilde{u}_t, \widetilde{\theta}_t). \tag{2}$$

This notation of a "black box" state transition function follows notation employed in control theory, for example by Chen [7].

Example 1: A familiar special case of (2) occurs if $\widetilde{u}$ and $\widetilde{\theta}$ are smooth functions and there exists a function f such that

$$\Phi(x(0), \widetilde{u}_t, \widetilde{\theta}_t) = x(0) + \int_0^t f(x(s), u(s), \theta(s)) \mathrm{d}s. \tag{3}$$

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

Example 2: Consider a point in the plane with velocity input, for which the motion is subject to noise. Let u_{max}

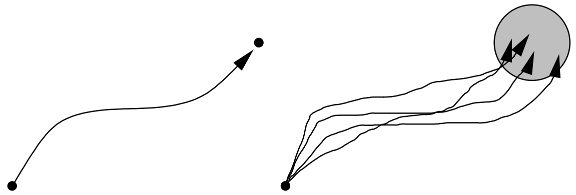

Fig. 3. [left] The robot in Example 2 gives velocity inputs that determine a nominal trajectory. [right] Nature interferes with this trajectory, but error bounds ensure that the final state is contained in a circle of radius $t\theta_{max}$.

denote a bound on the magnitude of the commanded velocity, and let θ_{max} denote a bound on magnitude of the error in the velocity. 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

$$\Phi(x(0), \widetilde{u}_t, \theta_t) = x(0) + \int_0^t (u(s) + \theta(s))\mathrm{d}s. \qquad (4)$$

At every time t, the robot can be certain that its state lies within a closed ball of radius $t\theta_{max}$, centered at the nominal (error free, i.e. $\widetilde{\theta} \equiv (0,0)$) final point. See Figure 3. □

C. Observations

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 $\widetilde{y} : [0,\infty) \to Y$. The observation history up to t (inclusive) is $\widetilde{y}_t \in \widetilde{Y}_t$.

Nature interferes with the observations by choosing a *nature observation action* from a space Ψ. Let $\widetilde{\Psi}_t$ denote the space of functions mapping $[0,t)$ into Ψ and let $\widetilde{\Psi} = \bigcup_{t\in[0,\infty)} \widetilde{\Psi}_t$. The robot's complete nature observation action history is $\widetilde{\psi} : [0,\infty) \to \Psi$; the nature observation action history up to time (but not including) t is $\widetilde{\psi}_t \in \widetilde{\Psi}_t$. The observations received by the robot are governed by the *observation function* $h : X \times \Psi \to Y$.

Example 3: Suppose the mobile robot has a sensor that detects the distance to some landmark. Let $X = \mathbb{R}^2$ and $Y = \mathbb{R}$. Without loss of generality, position the landmark at the origin. Assume that the sensor has bounded additive disturbance, so that $\Psi = [-\psi_{max}, \psi_{max}]$ and $h(x,\psi) = ||x|| + \psi$. See Figure 4. At each instant, the robot knows with certainty that its state is within an annulus of width $2\psi_{max}$ centered at the origin. □

D. Information spaces and information mappings

To inform its decisions, the robot has access only to the histories of actions it has selected and observations it has received so far. That is, to select $u(t)$, the robot can use $\widetilde{u}_t$ and $\widetilde{y}_t$. This motivates our definition of the *history information space*:

$$\mathcal{I}_{hist} = \bigcup_{t\in[0,\infty)} \widetilde{U}_t \times \widetilde{Y}_t \qquad (5)$$

The tuple $\eta(t) = (\widetilde{u}_t, \widetilde{y}_t) \in \mathcal{I}_{hist}$ containing the robot's action and sensing histories is the robot's *history information state*.

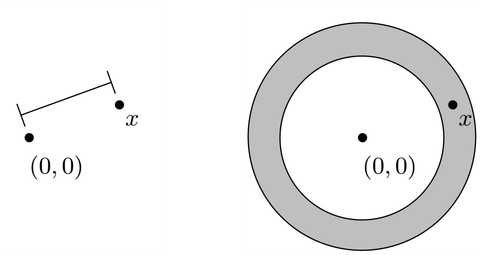

Fig. 4. [left] The robot in Example 3 has a sensor that reports a noisy estimate of the distance to the origin. [right] Accounting for noise bounded by ψ_{max}, the observation confines the robot's state to an annulus of width $2\psi_{max}$.

The history information state, since it is composed of functions of time, is unwieldy in isolation. As a result, we select a *derived information space* $\mathcal{I}$ and an information mapping $\kappa : \mathcal{I}_{hist} \to \mathcal{I}$. Informally, a derived information space represents "compression" or "interpretation" of the histories.

We say that a state x is *consistent* with an information state $\eta(t) = (\widetilde{u}_t, \widetilde{y}_t)$ if and only if there exists some starting state $x(0)$ and nature histories $\widetilde{\theta}$ and $\widetilde{\psi}$ such that $\Phi(x(0), \widetilde{u}_t, \widetilde{\theta}_t) = x$ and $h(x(t'), \widetilde{\psi}(t')) = y(t')$ for $t' < t$. The next example is an information mapping that arises directly from the notion of consistent states.

Example 4 (Nondeterministic information mapping): Let $\mathcal{I}_{ndet} = \mathrm{pow}(X) - \emptyset$. The relevant information mapping is $\kappa_{ndet} : \mathcal{I}_{hist} \to \mathcal{I}_{ndet}$, under which each history information state maps to the minimal subset of X consistent with it. The intuition is that $\eta(t)$ gives a set of "possible states" for the robot at time t. Because the robot's true state is always consistent with its history information state, this set is never empty. □

E. Information feedback strategies

How does the robot decide which actions to select? We describe the robot's strategy as a feedback strategy $\pi : \mathcal{I}_{hist} \to U$ that specifies an action for history information state. As the robot executes π, the actions are given by $u(t) = \pi(\eta(t))$. We call π an *information feedback strategy*.

Even though we define π as a feedback strategy over the history information space, the next two examples illustrate that feedback over a derived information space can sometimes be a natural way to express familiar kinds of strategies.

Example 5 (Open loop strategy): Let $\mathcal{I}_{time} = [0,\infty)$ and consider the information map $\kappa_{time}(\eta(t)) = t$. In this case, the derived information state is simply the time elapsed. Then if the robot has an intended open loop action trajectory $\omega : [0, t_f) \to U$, a strategy to execute γ is $\pi(\eta(t)) = \omega(\kappa_{time}(\eta(t))$ if $t < t_f$ and $\pi(\eta(t)) = u_T$ otherwise. □

Example 6 (Memoryless strategy): Another possibility is that it is enough to know the "most recent" observation, so $\mathcal{I}_{obs} = Y$ and $\kappa_{obs}(\eta(t)) = y(t)$. Given a memoryless plan $\gamma : Y \to U$, the composed function $\kappa_{obs} \circ \gamma : \mathcal{I}_{hist} \to U$ is a memoryless information feedback strategy.[1] □

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 itself π, the starting state $x(0)$, and the actions of nature $\widetilde{\theta}$ and $\widetilde{\psi}$. Assuming that the robot executes π, the termination time is $T(\pi, x(0), \widetilde{\theta}, \widetilde{\psi}) = \inf\{t \in [0,\infty) \mid \pi(\eta(t)) = u_T\}$, and the final state is $F(\pi, x(0), \widetilde{\theta}, \widetilde{\psi}) = \Phi(x(0), \widetilde{u}_{t_f}, \widetilde{\theta}_{t_f})$, in which $t_f = T(\pi, x(0), \widetilde{\theta}, \widetilde{\psi})$.

Example 7 (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. □

F. Tasks and solutions

A *task* (or problem) is defined by a goal region $\mathcal{I}_G \subset \mathcal{I}_{hist}$ in history information space. This notion is a generalization of the traditional idea of a goal state or goal region in state space. Generally, a *solution* is an information feedback strategy that reaches $\mathcal{I}_G$. In the presence of uncertainty in control and sensing, there are two relevant solution concepts.

1) A strategy is a *possible solution* if there exists some time t_g, some $\widetilde{\theta}_{t_g} \in \widetilde{\Theta}_{t_g}$ and some $\widetilde{\psi}_{t_g} \in \widetilde{\Psi}_{t_g}$, such that $\eta(t_g) \in \mathcal{I}_G$. 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 is a *guaranteed solution* if there exists some time t_g such that, for any $\widetilde{\theta}_{t_g}$ and any $\widetilde{\psi}_{t_g}$, $\eta(t_g) \in \mathcal{I}_G$. 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 will not consider them here.

IV. COMPARING ROBOT SYSTEMS

In this section, we show that the basic results of [23] still hold in our generalized framework. We define a dominance relation between robot systems to formalize the informal idea that some robots are "more powerful" than others, in the sense of having richer sensing and motion abilities. This relation has direct implications on the ability of robot systems to complete tasks.

A. Information preference relation

The first ingredient we need is some notion of when one derived information state is "better than" another. Fix a derived information state $\mathcal{I}$ and an information mapping $\kappa : \mathcal{I}_{hist} \to \mathcal{I}$. Equip $\mathcal{I}$ with a partial order *information preference relation* $\preceq$, under which $\eta_1 \preceq \eta_2$ means that η_2 is "more informed than" η_1. The only constraint on $\preceq$ is that is must be a partial order satisfying the following consistency property for any $t \in [0,\infty)$, $\widetilde{u}_t \in \widetilde{U}_t$ and $\widetilde{y}_t \in \widetilde{Y}_t$:

$$\kappa(\eta_1) \preceq \kappa(\eta_2) \implies \kappa(\eta_1, \widetilde{u}_t, \widetilde{y}_t) \preceq \kappa(\eta_2, \widetilde{u}_t, \widetilde{y}_t), \quad (6)$$

in which the concatenation on the right side indicates that the additional history information from $\widetilde{u}_t$ and $\widetilde{y}_t$ is appended to η_1 and η_2. The intuition is that information preference must be preserved if the same actions are selected and the same observations received from both η_1 and η_2.

Example 8: Recall κ_{ndet} from Example 4. Define $\preceq$ so that $\eta_1 \preceq \eta_2$ if and only if $\kappa_{ndet}(\eta_2) \subseteq \kappa_{ndet}(\eta_1)$. It is easy to verify that the consistency property holds. □

B. Definition of dominance

Our goal is a formal way to compare the power of robot systems. Consider two robot systems R_1 and R_2 defined as in Section III:

$$\begin{aligned} R_1 &= (X^{(1)}, U^{(1)}, Y^{(1)}, \Theta^{(1)}, \Psi^{(1)}, \Phi^{(1)}, h^{(1)}) \quad (7)\\ R_2 &= (X^{(2)}, U^{(2)}, Y^{(2)}, \Theta^{(2)}, \Psi^{(2)}, \Phi^{(2)}, h^{(2)}) \quad (8)\end{aligned}$$

Because $U^{(1)}$ need not have any special relationship to $U^{(2)}$, and likewise $Y^{(1)}$ need not be related to $Y^{(2)}$, the comparison cannot be made directly in the history information space, which simply records actions and observations. Instead, map the two history information spaces to the same derived information space. The corresponding information mappings are $\kappa^{(1)} : \mathcal{I}^{(1)}_{hist} \to \mathcal{I}$ and $\kappa^{(2)} : \mathcal{I}^{(2)}_{hist} \to \mathcal{I}$.

To compare distinct robot systems (perhaps with distinct state spaces) operating in the same family of environments, use the environment space construction described in Section III-A with R_1 and R_2 in the same environment space, so that $X^{(1)} = \mathcal{C}^{(1)} \times \mathcal{E}$ and $X^{(2)} = \mathcal{C}^{(2)} \times \mathcal{E}$.

Now we can state the dominance relation between robot systems.

Definition 1 (Robot dominance): Consider two robots R_1 and R_2. If, for all

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

there exists an information feedback strategy $\pi_2 : \mathcal{I}^{(2)}_{hist} \to U^{(2)}$, 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 for all

- $\widetilde{\theta}^{(1)}_{t'_1} \in \Theta^{(1)}_{t'_1}$,
- $\widetilde{\theta}^{(2)}_{t'_2} \in \Theta^{(2)}_{t'_2}$,
- $\widetilde{\psi}^{(1)}_{t'_1} \in \widetilde{\Psi}^{(1)}_{t'_1}$, and
- $\widetilde{\psi}^{(2)}_{t'_2} \in \widetilde{\Psi}^{(2)}_{t'_2}$,

if R_1 executes $\widetilde{u}^{(1)}_{t'_1}$ from time t_1 to t'_1 and R_2 executes $\pi^{(2)}$ from time t_2 to t'_2, we have

$$\kappa(\eta^{(1)}(t'_1)) \preceq \kappa(\eta^{(2)}(t'_2)) \quad (9)$$

[1] In [23], we used a slightly different observation model, in which $h : X \times U \to Y$. In this context, the time period over which observations are available is the half-open interval $[0,t)$; $\widetilde{y}_t$ is undefined at t itself. As a result, the closest we can come to a memoryless strategy is to use the left-hand limit of $\widetilde{y}_t$ at t, $\kappa_{obs}(\eta(t)) = \lim_{t' \to t^-} y(t')$, provided the limit exists. This technicality is part of the motivation for preventing y from depending directly on u, as we have done in this paper.

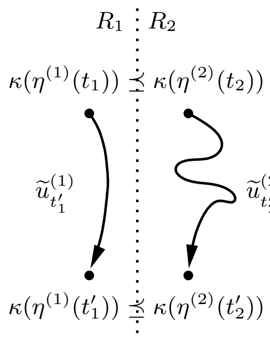

Fig. 5. An illustration of Definition 1.

then R_2 *dominates* R_1 under $\mathcal{I}$, κ, and $\preceq$, denoted $R_1 \trianglelefteq R_2$. If $R_1 \trianglelefteq R_2$ and $R_2 \trianglelefteq R_1$, then R_1 and R_2 are *equivalent*, denoted $R_1 \equiv R_2$. If $R_1 \ntrianglelefteq R_2$ and $R_2 \ntrianglelefteq R_1$ then R_1 and R_2 are *incomparable*, denoted $R_1 \boxtimes R_2$. ◦

Informally, Definition 1 means that, regardless of the transitions made by R_1 (and regardless of the interference from nature R_1 receives), 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. Figure 5 illustrates this intuition.

C. Dominance and solvability

Now we can establish the relationship between dominance and solvability. Fist, we define a class of "well-formed" tasks based on the information preference relation.

Definition 2: Consider a set $I \subset \mathcal{I}$ of derived information states. If, for any $\eta_1 \in I$ and $\eta_2 \in \mathcal{I}$ with $\eta_1 \preceq \eta_2$, we have $\eta_2 \in I$, then I is *preference closed*. ◦

For any preference closed goal region, we have the following result. A similar, but weaker (because of the limitations in robot models) result appeared in [23].

Lemma 1 (Solution by imitation): Consider two robot systems R_1 and R_2 with $R_1 \trianglelefteq R_2$ and a preference-closed goal region $\mathcal{I}_G$. If there exists a guaranteed solution for R_1 to reach $\mathcal{I}_G$, then also there exists a guaranteed solution for R_2 to reach $\mathcal{I}_G$.

Proof: Execute the strategy π_2 implied by Definition 1 with R_2. Because $R_1 \trianglelefteq R_2$, the final derived information state $\eta^{(2)}_{t'_2}$ reached by R_2 will be preferred to the final derived information state $\eta^{(1)}_{t'_1}$ reached by R_1. Because $\mathcal{I}_G$ is preference closed and $\eta^{(1)}_{t'_1} \in \mathcal{I}_G$, we have $\eta^{(2)}_{t'_2} \in \mathcal{I}_G$. □

D. Dominance examples

This section presents a few examples to illustrate the implications of Definition 1.

Example 9 (Omniscient sensing and perfect control): Consider a degenerate case with $Y = X$, and $h(x, \psi) = x$. Let $\Theta = \Psi = \{0\}$ be dummy singleton sets with no effect on state transitions or observations. This situation gives the robot perfect control and complete information about its state. Choose $\kappa(\eta(t)) = y(t) = x(t)$. Let $\eta_1 \preceq \eta_2$ if and only if $\eta_1 = \eta_2$. In this context, Definition 1 becomes a statement about the regions of state space reachable by different control systems.

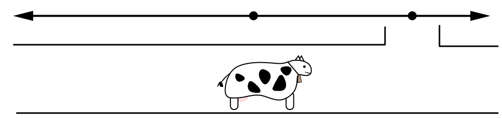

Fig. 6. The lost cow of Example 11 searching for a gate.

Suppose 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 $R_1 \trianglelefteq R_2$. 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 for which we omit the easy proofs because of space limitations:

$$Z(U^{(1)}) \subseteq Z(U^{(2)} \cup U^{(3)}) \tag{10}$$

$$Z(U^{(1)}) = Z(U^{(1)} \cup U^{(2)}) \tag{11}$$

$$Z(U^{(1)} \cup U^{(3)}) \subseteq Z(U^{(2)} \cup U^{(3)}) \tag{12}$$

These results are somewhat analogous to Lemmas 2-4 in [23]. 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 10 (Varying error bounds): Recall the incompletely specified models in Examples 2 and 3. Consider two robot systems R_1 and R_2 with state transitions as in Example 2 and observations as in Example 3; R_1 and R_2 differ only in the error bounds $\theta^{(1)}_{max}$, $\psi^{(1)}_{max}$, $\theta^{(2)}_{max}$, and $\psi^{(2)}_{max}$. We will compare these robots under κ_{ndet}.

Comparing $\theta^{(1)}_{max}$ to $\theta^{(2)}_{max}$, and $\psi^{(1)}_{max}$ to $\psi^{(2)}_{max}$, there are three cases:

1) If $\theta^{(1)}_{max} \le \theta^{(2)}_{max}$ and $\psi^{(1)}_{max} \le \psi^{(2)}_{max}$, then $R_2 \trianglelefteq R_1$.
2) If $\theta^{(2)}_{max} \le \theta^{(1)}_{max}$ and $\psi^{(2)}_{max} \le \psi^{(1)}_{max}$, then $R_1 \trianglelefteq R_2$.
3) If $\theta^{(1)}_{max} \le \theta^{(2)}_{max}$ and $\psi^{(2)}_{max} \le \psi^{(1)}_{max}$ or $\theta^{(2)}_{max} \le \theta^{(1)}_{max}$ and $\psi^{(1)}_{max} \le \psi^{(2)}_{max}$, then $R_2 \boxtimes R_1$.

This implies that $\theta^{(1)}_{max} = \theta^{(2)}_{max}$ and $\psi^{(1)}_{max} = \psi^{(2)}_{max}$ if and only if $R_1 \equiv R_2$. These results follow in a straightforward manner from Definition 1. The intuition of this (perhaps unsurprising) example is that one robot system dominates the other if its error bounds are smaller. □

Example 11 (A Lost Cow): A well-known problem in online algorithms is the *lost cow problem* [1], [15] in which a near-sighted cow moves along a fence searching for a gate, as illustrated in Figure 6. 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. For simplicity, assume perfect control and perfect sensing by setting $\Theta = \Psi = \{0\}$. The action space is $U = [-1, 1]$, with $\Theta = \{0\}$ and $\Phi(x(0), \widetilde{u}_t, \theta_t) = x(0) + \int_0^t u(s)\mathrm{d}s$. We compare three distinct models C_1, C_2, and C_3 under κ_{ndet}.

1) C_1: Let $Y^{(1)} = \mathbb{R}$ and $h^{(1)}(x, \psi) = x$. Here the cow can determine both the direction and distance to the gate.
2) C_2: Let $Y^{(2)} = \{-1, 0, 1\}$ and $h(x, \psi) = \mathrm{sign}(x)$. This allows the cow to determine the direction it must move to reach the gate, but not the distance.
3) C_3: Let $Y^{(3)} = \{0, 1\}$ and $h^{(2)}(x, \psi) = 1$ if $x = 0$ and $h^{(2)}(x, \psi) = 1$ 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 1. This comes about as a result of the fact that each can eventually determine its state (by finding the gate) and after the state is known, the state uncertainty cannot recur. To simulate C_1 with C_3, first execute the algorithm of [1], then move to the state occupied by C_1. □

V. A DISCRETE-STAGE MODEL

This section describes how the continuous-time model given in Section III is related to the discrete-stage formulation of [23].

A. Transforming from continuous time to discrete stages

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 3 (History invariance): If, for all $\eta(t) \in \mathcal{I}_{hist}$, all $x \in X$ consistent with $\eta(t)$, all $\widetilde{\theta} \in \widetilde{\Theta}$, all $\widetilde{\psi} \in \widetilde{\Psi}$, and all $y(0) \in Y$, we have

$$F(\pi, x, \eta(t), \widetilde{\theta}, \widetilde{\psi}) = F(\pi, x, \eta(0), \widetilde{\theta}, \widetilde{\psi}), \qquad (13)$$

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 $R = (X, U, Y, \Theta, \Psi, \Phi, h)$ as in Section III and a set Π of history-invariant information feedback strategies, construct a discrete-stage system $\overline{R} = (X, \overline{U}, \overline{Y}, \overline{\Theta}, \overline{\Psi}, \overline{f}, \overline{h})$ as follows:

1) The state space X is unchanged.
2) The action space is $\overline{U} = \Pi$.
3) The observation space is $\overline{Y} = \widetilde{Y}$.
4) The nature action space is $\overline{\Theta} = \widetilde{\Theta}$.
5) The nature observation action space is $\overline{\Psi} = \widetilde{\Psi}$.
6) The state transition function is $f : X \times \overline{U} \to X$, with $f(x, \pi) = F(\pi, x, \widetilde{\theta}, \eta(0))$.
7) The observation function is $h : X \times \overline{U} \times \overline{\Psi} \to \overline{Y}$.

The system starts at some (unknown) initial state $x_1 \in X$. Let $x_k \in X$, $u_k \in \overline{U}$, $y_k \in \overline{Y}$, $\theta \in \overline{\Theta}$, and $\psi_k \in \overline{\Psi}$ denote the appropriate values at stage k. These sequences are related to each other by $x_{k+1} = f(x_k, u_k, \theta_k)$ and $y_k = h(x_k, u_k, \psi_k)$. The history information state consists of the action and observation histories: $\eta_k = (u_1, y_1, \ldots, u_{K-1}, y_{K-1})$. We now argue that this discretized system faithfully represents the underlying continuous-time system.

Lemma 2: Any action sequence $u_1, \ldots, u_K$ executed by $\overline{R}$ reaches the same final state x and the analogous final history information state as does R.

Proof sketch: Use induction on k and the fact that the strategies in $\overline{U}$ are history invariant to show that for each $1 \leq k \leq K$, there exists t_k with the state x_k for $\overline{R}$ equal to the state $x(t_k)$ for R. □

Note, however, that in making this transformation, we may restrict the space of strategies that the robot can employ. If $\overline{U}$ does not contain a sufficiently rich selection of information feedback strategies, there may be regions of information space that are no longer reachable under the discretized model. It remains an open problem to find small (or at least succinctly described) sets of strategies that are complete or nearly complete in the sense of not eliminating any reachable regions in information space.

B. The role of robotic primitives

In [23], a universe of robot models is generated by a collection of *robotic primitives*, each of which gives partial action and observation sets. A complete model is formed by choosing a nonempty subset of primitives. How are they related to the continuous time models described in Section III? What role do these primitives play?

The robotic primitives serve two basic purposes. First, they provide a clean way of discretizing time. In the discrete-stage model, the physical time taken to execute each primitive is a concern secondary to the termination conditions under which the primitive terminates. This behavior is analogous to the termination action used in the current paper, and can be mimicked by concatenating motion strategies, as in Example 7. Second, a catalog of primitives is an effective way to generate a set of robot models to consider. Given nonempty sets of primitives, it is easy to combine, via unions of these sets, robots constructed from primitives, resulting in a sort of "calculus" over robot models in which individual components can be added or taken away. The appropriate analog for our new continuous time systems with nature is less clear.

VI. CONCLUSION

Although the results we present here are a substantial improvement over those of [23], there are still important pieces missing.

A. Computational issues

We have focused mostly on the sensing and motion requirements of tasks. An important related question is to determine the kinds of computation power these tasks require. What are the tradeoffs between computation time, memory

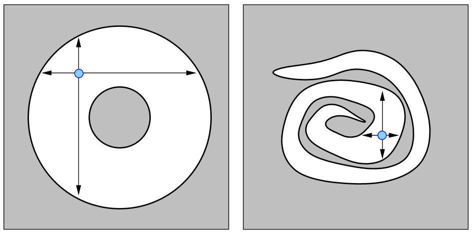

Fig. 7. A sample decision problem. What sensing is required to decide if a planar environment is simply connected? What robots can distinguish the annulus environment on the left from the helix on the right?

usage, sensing requirements and solution quality? Is there a satisfactory way to scalarize these competing objectives into a single-valued objective function, or should we expect a single problem will lead to many different Pareto optimal solutions? Some research has been done on computation requirements for certain tasks, for example [3], [5].

B. Reductions and decision problems

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 must determine if its environment $E \in \mathcal{E}$ has a certain property. Such problems fit naturally as planning problems in information space. To decide if E has a property $\Xi : \mathcal{E} \rightarrow \{0, 1\}$, the robot must reach the goal region

$$\begin{aligned} \mathcal{I}_{G\,\Xi} = \{\eta \in \mathcal{I}_{hist} \mid \forall (q, E) \in \kappa_{ndet}(\eta), \Xi(E) = 1\} \\ \cup \{\eta \in \mathcal{I}_{hist} \mid \forall (q, E) \in \kappa_{ndet}(\eta), \Xi(E) = 0\}. \end{aligned} \quad (14)$$

An example is in Figure 7.

C. Parameterization of time

By parameterizing the robot's observations by time, we have been implicitly assuming 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." For example, in the current scheme, a robot can use this implicit clock to parlay an accurate velocity sensor into a perfect odometer. One solution is to express $\widetilde{u}$ and $\widetilde{y}$ as functions of some other abstract parameter p. Then, 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.

D. Cooperation and coordination

In this work we consider 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 information spaces that would arise from the interaction of multiple agents, each having only limited information.

References

[1] R. A. Baeza-Yates, J. C. Culberson, and G. J. E. Rawlins, "Searching in the plane," *Information and Computation*, vol. 106, pp. 234–252, 1993.

[2] J. Barraquand and P. Ferbach, "Motion planning with uncertainty: The information space approach," in *Proceedings IEEE International Conference on Robotics & Automation*, 1995, pp. 1341–1348.

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

[4] D. Blackwell and M. A. Girshik, *Theory of Games and Statisitical Decisions*. New York: Dover, 1979.

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

[6] R. I. Brafman, J. Y. Halpern, and Y. Shoham, "On the knowledge requirements of tasks," *Artificial Intelligence*, vol. 98, no. 1-2, pp. 317–349, 1998.

[7] C.-T. Chen, *Linear System Theory and Design*. New York: Holt, Rinehart, and Winston, 1984.

[8] B. R. Donald, "On information invariants in robotics," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 217–304, 1995.

[9] M. Egerstedt, "Motion description languages for multi-modal control in robotics," in *Control Problems in Robotics*, ser. Springer Tracts in Advanced Robotics, A. Bicchi, H. Cristensen, and D. Prattichizzo, Eds., 2002, pp. 75–90.

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

[11] ——, "Understanding action and sensing by designing action-based sensors," *International Journal of Robotics Research*, vol. 14, no. 5, pp. 483–509, 1995.

[12] A. Girard and G. J. Pappas, "Approximation metrics for discrete and continuous systems," *IEEE Transactions on Automatic Control*, Mar. 2005, to appear.

[13] K. Y. Goldberg and M. T. Mason, "Bayesian grasping," in *Proceedings IEEE International Conference on Robotics and Automation*, 1990.

[14] J. E. Hopcroft, J. D. Ullman, and R. Motwani, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed. Reading, MA: Addison-Wesley, 2000.

[15] M.-Y. Kao, J. H. Reif, and S. R. Tate, "Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem," in *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 1993, pp. 441–447.

[16] R. M. Karp, "On-line algorithms versus off-line algorithms: How much is it worth to know the future?" in *Proceedings World Computer Congress*, 1992.

[17] H. W. Kuhn, "Extensive games and the problem of information," in *Contributions to the Theory of Games*, H. W. Kuhn and A. W. Tucker, Eds. Princeton, NJ: Princeton University Press, 1953, pp. 196–216.

[18] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[19] S. M. LaValle and S. A. Hutchinson, "An objective-based framework for motion planning under sensing and control uncertainties," *International Journal of Robotics Research*, vol. 17, no. 1, pp. 19–42, Jan. 1998.

[20] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.

[21] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, 1979.

[22] M. T. Mason, "The mechanics of manipulation," in *Proceedings IEEE International Conference on Robotics & Automation*, 1985, pp. 544–548.

[23] J. M. O'Kane and S. M. LaValle, "On comparing the power of mobile robots," in *Proc. Robotics: Science and Systems*, 2006.

[24] C. H. Papadimitriou, "Games against nature," *Journal of Computer and System Sciences*, vol. 31, pp. 288–301, 1985.

[25] C. H. Papadimitriou and M. Yannakakis, "Shortest paths without a map," *Theoretical Computer Science*, vol. 84, pp. 127–150, 1991.

[26] D. D. Sleator and R. E. Tarjan, "Amortized efficiency of list update and paging rules," *Communications of the ACM*, vol. 28, pp. 202–208, 1985.