

On Comparing the Power of Mobile Robots

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

Abstract—Minimalist models have been studied for a broad array of tasks in robotics. In this paper, we consider the task-completing power of robots in terms of the sensors and actuators with which the robot is equipped. Our goal is to understand the relative power of different sets of sensors and actuators and to determine which of these sets enable the robot to complete its task. We define robots as collections of *robotic primitives* and provide a formal method for comparing the sensing and actuation power of robots constructed from these primitives. This comparison, which is based on the 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 then apply it to a limited-sensing version of the global localization problem.

I. INTRODUCTION

Among many researchers in robotics, there is a movement toward *minimalism*. The minimalist philosophy is to use robots with very simple sensing and actuation capabilities. A given task will partition any sufficiently rich set of robots into two subsets, one containing robots that can complete the task and another containing robots that cannot. In this context, it is natural to identify the *simplest* robots that can complete the task. An alternative view of the same approach is to seek *necessary conditions* on robot systems that are able to complete the task.

To this end, we ask two central questions in this work. What combinations of sensors and actuators enable a robot to complete a given task? What does it mean for one robot to be “simpler” than another? We address these questions by carefully defining a set of robot systems and presenting a formal method for comparing the robots in this set.

We want to study a robot’s ability complete certain tasks as determined by the “complexity” of the robot. To be precise, we consider the capabilities of the robot in terms of the set of sensors and actuators to which the robot has access. We allow the robot to perform computations, but we do not restrict the time nor space complexity of this computation.

A. Overview

We are interested in problems for which sensing (or the lack of sensing) is important. At the heart of the issue is the idea that the robot does not know its true state. Instead, it must use its history of actions and sensor readings to draw conclusions about its state. Since different robot systems have distinct spaces of actions and sensor readings, the sensor-action histories cannot be directly compared. Therefore, we use a *derived information space* as the meeting ground on

which comparisons can be made. 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.

Our objective is to compare the power of robots with varying combinations of sensorimotor components. To formalize the set of robots to which our analysis applies, we introduce the idea of *robotic primitives*. 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.

The central idea is the notion of the *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 .

The meaning of this definition hinges on the idea of one derived information state giving the robot “at least as much” information as another. This leads us to augment the derived information space with a partial order that indicates preferences for some information states over others.

For concreteness, we apply these ideas to the task of *global localization*, in which the robot must move from total uncertainty to total certainty in its state. What sensor sets are minimal for localization, in the sense that eliminating any prevents the robot from localizing itself? Using the dominance idea described above, we give a partial answer to that question.

In this paper we make three primary contributions: First, we present the idea of robotic primitives for modeling robot systems as collections of independent components. Second, we give a definition for dominance of one robot system over another that formalizes the imprecise definition above. We prove some basic properties of this relation. Third, we apply these ideas to a limited-sensing version of the global localization problem. We define a catalog of 4 robotic primitives and fully determine which of the 15 robotic systems induced by these primitives can solve the localization problem.

B. Related work

Several lines of research are closely related. The minimalist approach has a long history, dating perhaps to Whitney [42]. Minimalist approaches have been used in manufacturing contexts for part orientation [2], [3], [17], [18], [19], [31], [40], [43] and in mobile robotics for navigation and exploration [1], [10], [25], [22], [30], [32], [39].

More directly, a few works have attempted to identify sensing requirements of tasks. Brafman et al. [9] consider very similar issues for distributed teams of physically identical robots in discrete state spaces. Donald [14] also performs a similar analysis, characterizing tasks according to their complexity measured in bit-seconds. Erdmann [16] applies preimage backchaining to the problem of sensor design. Although we are strictly interested in the physical abilities of the robot, other work has explored the computation power needed for various tasks [7], [8], [21]. Our work is distinguished from these antecedents in at least two important ways. First, rather than holding the robot’s motor capabilities fixed, we allow interactions between sensing and motion to be explicitly modeled with robotic primitives. Second, by considering the problem in the robot’s information space, we can, in a general way, explicitly reason about the robot’s state uncertainty.

Since we consider a global localization task in some detail, it is worth mentioning the enormous body of work on localization methods for robots with many different kinds of sensing systems. Some methods [5], [6], [11], [12], [13], [20], [37], [26], [41] passively observe the motions of the robot in order to draw conclusions about the robot’s state. Others [15], [23], [24], [33], [34], [35], [36] actively drive the robot to reduce uncertainty.

While the examples in this paper use nondeterministic uncertainty, which is based on set membership, the basic structure of our analysis is compatible with probabilistic uncertainty models like those of [38]. Many probabilistic methods (for example, [4], [28]) can be characterized as operating in an information space whose members are probability distributions over state space. In this sense, the model of uncertainty we use is orthogonal to the questions we address in this work.

C. Organization

This paper is organized as follows. Section II lays a foundation of basic definitions for robotic planning problems. Section III introduces the concept of a robotic primitive and defines the set of robots induced by a catalog of primitives. In Section IV we describe the information preference relation, a partial ordering over derived information space that formalizes the idea that some information states are better than others. The definition of dominance and some basic properties thereof appear in Section V. In Section VI, we apply the results from Sections III, IV, and V to the global localization task. Section VII contains discussion and conclusions.

II. BASIC DEFINITIONS

In this section we present some general definitions for robotic planning problems.

A. States, actions, and observations

We allow a robot to move in a state space X . Generally X will be the configuration space [29] of the robot. The examples in this paper are for a point robot with orientation in the plane. In these examples, we use $X = W \times S^1$, in which $W \subset \mathbb{R}^2$ is the robot’s environment and $S^1 = [0, 2\pi]/\sim$, where \sim is

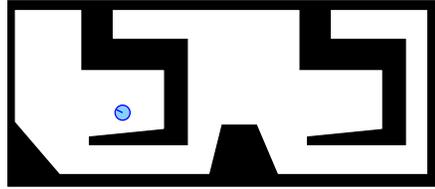


Fig. 1. A robot in a planar environment W . Its state space is $X = W \times S^1$.

an equivalence relation identifying 0 and 2π , represents the robot’s orientation.

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 $f : X \times U \rightarrow X$. At the conclusion of each stage, the robot’s sensors provide an observation y from an observation space Y , according to $h : X \times U \rightarrow Y$. Call h the robot’s *observation function*. Let x_k , u_k , and y_k denote respectively the state, action, and observation at stage k . These sequences are related to each other by f and h :

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

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

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

$$f^k(x, u_1, \dots, u_k) = f(\dots f(f(x, u_1), u_2) \dots), u_k). \quad (3)$$

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 sets.

Although we are assuming in this paper that both state transitions and observations are deterministic, we believe that in a broader context, managing unpredictability in motion and sensing is a crucial issue. We omit such uncertainty here only because it would unnecessarily complicate the analysis without providing many new insights. A discussion of the extensions needed to allow this kind of uncertainty appears in Section VII-A.

B. 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 \mathcal{I}_{hist} . After k stages, the robot’s *history information state* (history I-state) is a $2k$ -dimensional vector $\eta = (u_1, y_1, \dots, u_k, y_k)$. We occasionally abuse notation by writing (η, u_{k+1}, y_{k+1}) for the history I-state formed by appending u_{k+1} and y_{k+1} to η .

We may define a *policy* $\pi : \mathcal{I}_{hist} \rightarrow U$ over history I-space. As a shorthand, we define a function F that applies a policy several times in succession, starting with some state x :

$$F^0(\eta, \pi, x) = \eta \quad (4)$$

$$F^k(\eta, \pi, x) = (\eta_{k-1}, \pi(\eta_{k-1}), h(x, \pi(\eta_{k-1}))), \quad (5)$$

in which $\eta_{k-1} = F^{k-1}(\eta, \pi, x)$. Note that F^k depends on the true state x (which is unknown to the robot) because x influences the observation sequence that 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. For these reasons, we select a *derived information space* (derived I-space) \mathcal{I} and an *information mapping* (I-map) $\kappa : \mathcal{I}_{hist} \rightarrow \mathcal{I}$. 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} .

An important special case is the value of κ for an empty history, i.e. $\kappa(\cdot)$. This value gives an *initial condition* for the robot, reflecting any knowledge the robot may have before its execution begins. In principle, we may select \mathcal{I} and κ arbitrarily. The usefulness of a derived I-space lies in its ability to fully capture the information relevant to the task of interest.

Example 1: We define the *nondeterministic I-space* \mathcal{I}_{ndet} , in which derived I-states are nonempty subsets of X . The interpretation is that the robot’s derived I-state is a minimal set guaranteed to contain the true state. The I-map $\kappa_{ndet} : \mathcal{I}_{hist} \rightarrow \mathcal{I}_{ndet}$ can be defined recursively

$$\kappa_{ndet}(\cdot) = X \quad (6)$$

$$\kappa_{ndet}(\eta, u, y) = \{f(x, u) \mid x \in \kappa_{ndet}(\eta), y = h(x, u)\} \quad (7)$$

In Equation 6, we assume the robot initially has no information about its state. \square

C. Tasks and solutions

We define a *task* for the robot as a goal region in $\mathcal{I}_G \subseteq \mathcal{I}$ that the robot must reach. A *solution* is a policy π under which, for any $x \in X$, there exists l such that $F^l(\eta_1, \pi, x) \in \mathcal{I}_G$.

III. 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.

A. Robotic primitives

At the most concrete level, a robot is a conglomeration of motors and sensors connected to some sort of computer. Between these components there may be interactions via 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. Formally, we define robotic primitives in terms of the action and observation abilities they provide.

Definition 1: A *robotic primitive* (or simply a *primitive*) is a tuple

$$(U_i, Y_i, f_i, h_i) \quad (8)$$

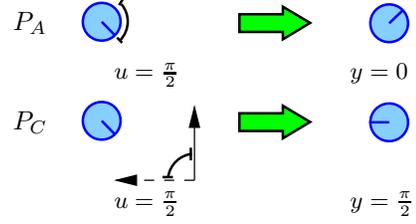


Fig. 2. Sample executions of the primitives of Examples 2 and 3. [top] P_A allows the robot rotate relative to its current orientation. [bottom] P_C allows the robot to rotate relative to a globally defined “north” direction.

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

We now give several examples to illustrate the idea. Examples 3-7 apply to a point robot with orientation in the plane, so $X = \mathbb{R}^2 \times S^1$. Illustrations of these primitives appear in Figures 2-4. We will revisit these examples in Sections V and VI.

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. \square

Example 3: Let $P_C = (S^1 \sqcup \{0\}, S^1, f_C, h_C)$. The \sqcup notation indicates a disjoint union operation, under which identical elements from different source sets remain distinct. Define $f_C(x, u)$ to set the rotation coordinate of x to equal 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. This primitive can be implemented using a compass on a robot that can rotate in place. \square

Example 4: Let $P_T = (\{0\}, \{0\}, f_T, h_T)$. Define f_T to compute a forward translation to the obstacle boundary. This primitive can be implemented with a contact sensor on a mobile robot that can reliably move forward. \square

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. Depending on implementation issues, a contact sensor may be needed as well. \square

Example 6: Let $P_R = (\{0\}, [0, \infty), f_R, h_R)$. Again $f(x, u) = x$ for all x and u . The observation $h(x, u)$ 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. \square

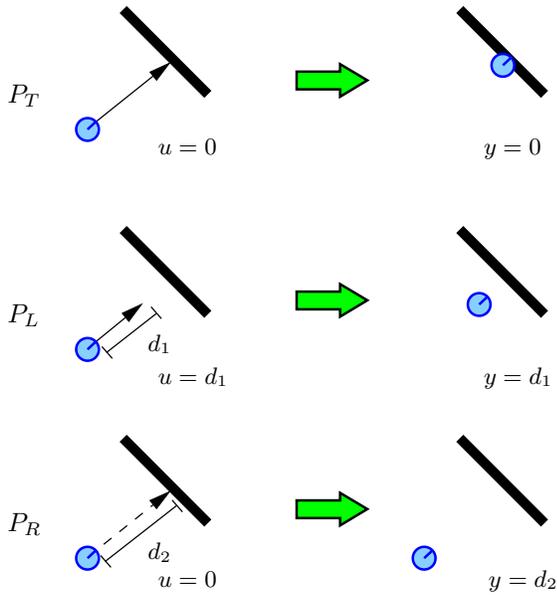


Fig. 3. Sample executions of the primitives of Examples 4-6. [top] P_T allows the robot to translate forward until it reaches an obstacle. [middle] P_L allows a robot to specify a distance to translate. [bottom] P_R allows the robot to measure the distance forward to the nearest obstacle, but does not change the robot's state.

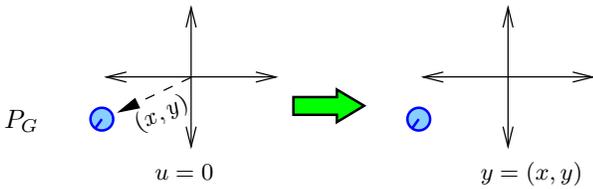


Fig. 4. A sample execution of the primitive of Example 7. The robot senses its position, but its state does not change.

Example 7: Let $P_G = (\{0\}, \mathbb{R}^2, f_G, h_G)$. For all $x \in X$, $f_G(x, 0) = x$, so that this primitive never changes the robot's state. For a state $x = (x_1, x_2, \theta)$, let $h(x, 0) = (x_1, x_2)$. This primitive roughly corresponds to a GPS device that the robot can periodically poll to determine its location in the plane. \square

Others possibilities for primitives include landmark sensors, wall followers, visibility sensors, etc. A more complete listing of sensors suitable for adaptation into robotic primitives appears in Section 11.5.1 of [27].

We believe there are several benefits to modeling robot systems as collections of primitives. First, we claim that robotic primitives represent roughly 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 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.

B. A catalog of primitives

Let $\mathcal{RP} = \{P_A, \dots, P_N\}$ denote a catalog of primitives. A *robot model* (or simply a *robot*) is a nonempty subset of \mathcal{RP} . A robot $R = \{P_{i_1}, \dots, P_{i_m}\} \subseteq \mathcal{RP}$ has action set $U_R = U_{i_1} \sqcup \dots \sqcup U_{i_m}$ and observation set $Y_R = Y_{i_1} \sqcup \dots \sqcup Y_{i_m}$. The state transition function $f_R : X \times U_R \rightarrow X$, and observation function $h_R : X \times U_R \rightarrow Y_R$, are formed by unioning the f and h maps from the relevant primitives. Since robots are defined by sets of primitives, it is meaningful to apply standard set operations such as union or intersection directly to robots. It is also helpful to define the *global history I-space* $\widehat{\mathcal{I}}_{hist}$, which contains all sensor-action histories that may be encountered by any robot constructed from primitives in \mathcal{RP} .

As we have already discussed, different types of robots have distinct action and observation sets. Therefore the histories cannot be directly compared. Instead, choose a derived I-space \mathcal{I} and an I-map $\kappa : \widehat{\mathcal{I}}_{hist} \rightarrow \mathcal{I}$ in which to make the comparisons. Since the domain of $\widehat{\mathcal{I}}_{hist}$ includes actions and observations for any primitive, κ fully defines a map from the history I-space of any robot model into \mathcal{I} . The choice of κ and \mathcal{I} is crucial because our definition of dominance (Definition 4) is parameterized by κ and \mathcal{I} . Our analysis is task-independent only to the extent that \mathcal{I} can encode the tasks (that is, the goal regions) in which we are interested.

IV. 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 "more informed than" another. To that end, equip \mathcal{I} with a partial order we call an *information preference relation*. Write $\eta_1 \preceq \eta_2$ to indicate that η_2 is a refinement of η_1 . We require \preceq to have the property that for any $\eta_1, \eta_2 \in \mathcal{I}_{hist}$, and for any $u \in U$ and $y \in Y$,

$$\kappa(\eta_1) \preceq \kappa(\eta_2) \implies \kappa(\eta_1, u, y) \preceq \kappa(\eta_2, u, y). \quad (9)$$

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 $\kappa(\eta_1) = \kappa(\eta_2)$. \square

Example 9: Under nondeterministic uncertainty, we can define $\kappa(\eta_1) \preceq \kappa(\eta_2)$ if and only if $\kappa(\eta_2) \subseteq \kappa(\eta_1)$. It is straightforward to show that Equation 9 is satisfied. \square

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. Definition 2 codifies this idea of a rational goal region.

Definition 2: Consider a set $I \subset \mathcal{I}$ of derived I-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*.

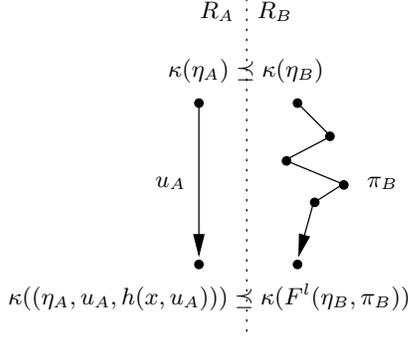


Fig. 5. An illustration of Definition 4. If R_2 can always reach an I-state better than the one reached by R_1 , then $R_1 \preceq R_2$.

Alternatively, we 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 . The trivial definition of \preceq in Example 8 always passes this test, regardless of \mathcal{G} .

V. 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. We begin with a technical definition.

Definition 3: A state $x \in X$ is *consistent* with a history I-state $\eta = (u_1, y_1, \dots, u_k, y_k)$ if there exists some $x_1 \in X$ such that $x = f^k(x_1, u_1, \dots, u_k)$ and $y_j = h(f^{j-1}(x_1, u_1, \dots, u_{j-1}), u_j)$ for each $j = 1, \dots, k$.

We can now define the dominance relation. The intuition is that dominance is based on one robot's ability to "simulate" another.

Definition 4 (Robot dominance): Consider two robots $R_1, R_2 \subseteq \mathcal{RP}$. If, for all

- $\eta_1 \in \mathcal{I}_{hist}^{(1)}$,
- $\eta_2 \in \mathcal{I}_{hist}^{(2)}$ with $\kappa(\eta_1) \preceq \kappa(\eta_2)$, and
- $u_1 \in U_1$,

there exists a policy $\pi_2 : \mathcal{I}_{hist}^{(2)} \rightarrow U$ and a positive integer l such that for all x consistent with both η_1 and η_2 ,

$$\kappa(\eta_1, u_1, h(x, u_1)) \preceq \kappa(F^l(\eta_2, \pi_2, x)) \quad (10)$$

then R_2 *dominates* R_1 under \mathcal{I} and κ , denoted $R_1 \trianglelefteq R_2$.

Informally, the definition says that R_2 can always select actions that take it to an I-state at least as good as the one reached by R_1 . See Figure 5. Definition 4 enables us to define robot equivalence in the natural way.

Definition 5 (Robot equivalence): For any $R_1, R_2 \subseteq \mathcal{RP}$, if $R_1 \trianglelefteq R_2$ and $R_2 \trianglelefteq R_1$ then R_1 and R_2 are *equivalent*, denoted $R_1 \equiv R_2$. The set of robots equivalent to R_1 is denoted $[R_1]$.

A. Dominance examples

Several examples will clarify the definition.

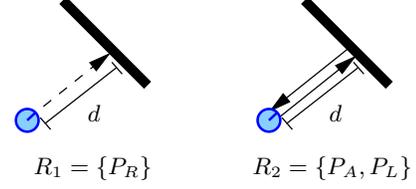


Fig. 6. An illustration of Example 10. The robot $R_2 = \{P_A, P_L\}$ dominates the robot $R_1 = \{P_R\}$ because the former can simulate the latter. [left] A distance measurement made directly by R_1 . [right] Distance is measured indirectly by R_2 using its linear odometer.

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 nondeterministic uncertainty that $R_1 \trianglelefteq R_2$ by showing that R_2 can simulate R_1 in the precise sense of Definition 4. Let $\eta_1 \in \mathcal{I}_{hist}^{(1)}$ and $\eta_2 \in \mathcal{I}_{hist}^{(2)}$ with $\kappa(\eta_1) \preceq \kappa(\eta_2)$. 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 6. From here it is easy to verify that from any $x \in X$, we have $\kappa(\eta_1, u_1, h(x, u_1)) \preceq \kappa(F^4(\eta_2, \pi_2, x))$. \square

Example 11: Let $R_1 = \{P_T\}$ and $R_2 = \{P_L\}$. We show under nondeterministic uncertainty that $R_1 \trianglelefteq R_2$. Let $\eta_1 \in \mathcal{I}_{hist}^{(1)}$ and $\eta_2 \in \mathcal{I}_{hist}^{(2)}$ with $\eta_1 \preceq \eta_2$. 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 η'_1 and η'_2 for R_1 and R_2 are the same, except that R_2 receives a meaningful sensor reading that may cause one or more candidates to be pruned. This sensor information only makes η'_2 smaller, so the preference $\eta'_1 \preceq \eta'_2$ is preserved. Conclude that $R_1 \trianglelefteq R_2$. \square

It bears emphasis that the relation induced by Definition 4 depends on κ . 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 nondeterministic I-map with the initial condition of total uncertainty. Let $\eta_1 \in \mathcal{I}_{hist}^{(1)}$ and $\eta_2 \in \mathcal{I}_{hist}^{(2)}$ with $\eta_1 \preceq \eta_2$. 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 candidates that R_1 cannot. This establishes that $R_1 \preceq R_2$.

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 $R_2 \not\equiv R_1$ and $R_1 \not\equiv R_2$. \square

Example 13: Consider a situation identical to that of Example 12, but modify κ for a different initial condition $\kappa(\cdot) = \mathbb{R}^2 \times \{\pi/2\}$. That is, the robot begins its execution knowing its initial orientation. Then at every step, R_1 knows its orientation in the global frame, and can simulate R_2 using angle addition. Therefore we have $R_2 \preceq R_1$. But using the same reasoning as in Example 12, we know $R_1 \preceq R_2$. Therefore, for this κ , we have $R_1 \equiv R_2$. \square

B. Properties of the dominance relation

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

Lemma 1: The dominance relation \preceq is partial order. Likewise \equiv is indeed an equivalence relation.

Lemma 2 (Adding primitives can only help): For any $R_1, R_3 \subseteq \mathcal{RP}$, $R_1 \preceq R_1 \cup R_3$.

Proof: Let $\eta_1 \in \mathcal{I}_{hist}^{(1)}$, $\eta_{13} \in \mathcal{I}_{hist}^{(13)}$, 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 . Then we have

$$\begin{aligned} \kappa(\eta_1, u_1, h(x, u_1)) &\preceq \kappa(\eta_{13}, u_1, h(x, u_1)) \\ &= \kappa(F^l(\eta_{13}, \pi_{13}, x)), \end{aligned} \quad (11)$$

completing the proof. \square

Lemma 3 (Redundancy doesn't help): If $R_1 \preceq R_2$, then $R_2 \equiv R_1 \cup R_2$

Proof: Since $R_2 \cup R_2 = R_2$, it follows from Lemma 2 that $R_2 \preceq R_1 \cup R_2$. It remains to show that $R_2 \cup R_1 \preceq R_2$. Let $\eta_2 \in \mathcal{I}_{hist}^{(2)}$, $\eta_{12} \in \mathcal{I}_{hist}^{(12)}$, and $u_2 \in U_2$. Assume $\kappa(\eta_2) \preceq \kappa(\eta_{12})$. Choose $l = 1$ and $\pi_{12}(\eta) = u_2$ for all η . Then we have

$$\begin{aligned} \kappa(\eta_B, u_B, h(x, u_B)) &\preceq \kappa(\eta_{12}, u_B, h(x, u_B)) \\ &= \kappa(F^l(\eta_{12}, \pi_{12}, x)), \end{aligned} \quad (12)$$

completing the proof. \square

Lemma 4 (No unexpected interactions): If $R_1 \preceq R_2$, then $R_1 \cup R_3 \preceq R_2 \cup R_3$.

Proof: Let $\eta_{13} \in \mathcal{I}_{hist}^{(13)}$, $\eta_{23} \in \mathcal{I}_{hist}^{(23)}$, 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 $R_1 \preceq R_2$ there exist π_3 and l satisfying the definition for $R_1 \cup R_3 \preceq R_2 \cup R_3$. If $u_{13} \in U_3$, then choose $l = 1$ and $\pi_{23}(\eta) = u_{13}$ for all η . Then for all x , we have

$$\begin{aligned} \kappa(\eta_{13}, u_{13}, h(x, u_{13})) &\preceq \kappa(\eta_{23}, u_{13}, h(x, u_{13})) \\ &= \kappa(F^l(\eta_{23}, \pi_{23}, x)), \end{aligned} \quad (13)$$

completing the proof. \square

Corollary 5: If $R_1 \equiv R_2$, then $R_1 \cup R_3 \equiv R_2 \cup R_3$.

Lemma 4 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 is an artifact of the selection of the primitive catalog. To be considered in this framework, low-level interactions should be modeled as members of \mathcal{RP} in their own right.

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

Lemma 6 (Solution by imitation): Consider two robots R_1 and R_2 with $R_1 \preceq R_2$ and a preference-closed goal region \mathcal{I}_G .

- (a) If R_1 can reach \mathcal{I}_G then R_2 can reach \mathcal{I}_G .
- (b) If R_2 cannot reach \mathcal{I}_G then R_1 cannot reach \mathcal{I}_G .

Proof: For part (a) use the policy π_2 implied by Definition 4 to complete the task with R_2 . Part (b) is the contrapositive of part (a). \square

VI. EXTENDED EXAMPLE: GLOBAL LOCALIZATION

In this section we present a detailed example illustrating the definitions of Sections IV and V. 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.

A. Task definition

Let $W \subseteq \mathbb{R}^2$ denote a planar environment in which a point robot moves. Assume that W is polygonal, bounded, closed, and simply-connected and that the rotational symmetry group of W is trivial. The robot's state space is $X = W \times S^1$, accounting for its position within W and its orientation.

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 a robot is equipped with, 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 nondeterministic map defined in Example 1. The initial condition is total uncertainty, so $\kappa(\cdot) = X$. For \preceq use the definition from Example 9. The goal region for the localization task is

$$\mathcal{I}_G = \{\eta \in \mathcal{I} \mid |\eta| = 1\}. \quad (14)$$

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 W consistent with the assumptions above, we say that the robot can localize itself.

B. Equivalences and dominances

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

Theorem 1: The following equivalences hold:

- (a) CA \equiv C
- (b) CTA \equiv CT
- (c) TL \equiv L

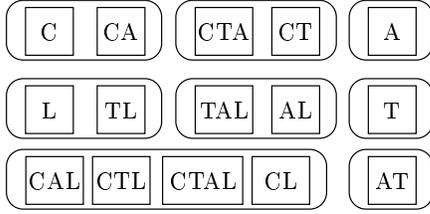


Fig. 7. Fifteen robot models grouped into their eight equivalence classes.

(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 3. (b) Combine Example 12, Lemma 3, and Corollary 5. (c) Combine Example 11 and Lemma 3. (d) Combine Example 11, Lemma 3, and Corollary 5. (e) Combine Examples 11 and 12, Lemma 3, and Corollary 5. \square

These equivalences are illustrated in Figure 7. From each, select the unique robot with the fewest primitives and discard the remaining 7 robots. We can state a number of dominances between these classes.

Theorem 2: Between representatives of the equivalence classes from Theorem 1, 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. \square

C. Completing the localization task

Which equivalence classes contain robots that can complete the localization task? First, notice that several robot models are so absurdly 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, notice 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, notice 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 W may contain continua of starting states consistent with this information, neither robot can localize itself. \square

Two prior results are helpful for the remaining cases.

Lemma 8 ([33], [34]): AL and CT can localize themselves but AT cannot.

Finally, we can finish the classification:

Lemma 9: CT can localize itself.

Proof: Combine Lemma 6 with Lemma 8. \square

The results of Lemmas 7-9 are summarized in Figure 8.

VII. DISCUSSION AND CONCLUSION

There are several issues that space limitations prevent us from fully describing.

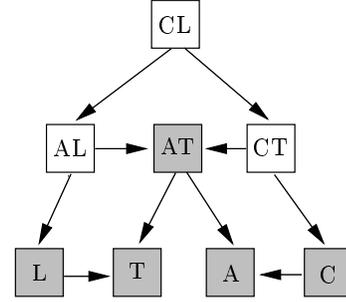


Fig. 8. Classification of robot models under which the localization task can be completed. Shaded models do not admit a solution. Arrows indicate dominances.

A. Uncertainty in transitions and sensing

First, as we have already mentioned, we require state transitions and observations to be deterministic. To relax this restriction, extend the state transition and observation functions to include a third parameter chosen by an external decision maker “nature”. This leads to $f : X \times U \times \Theta \rightarrow X$ and $h : X \times U \times \Psi \rightarrow Y$. In each stage, nature selects a $\theta \in \Theta$ and a $\psi \in \Psi$. Nature’s selections are governed by a nondeterministic or probabilistic uncertainty model which must be specified as part of the problem description. This extension can be incorporated into Definition 4 by the addition of universal quantifiers over Θ and Ψ .

B. Reachable sets and preference closure

The form of Definition 4 is local in an important sense. Comparisons are made based on a robot’s ability to simulate another robot’s trajectory in \mathcal{I} , step by step. An equivalent “global” definition can be defined in terms of reachable sets and preference closure. Let $Z(R_i) \subseteq \mathcal{I}$ denote the I-states reachable by R_i and for $N \subset \mathcal{I}$, let $Pc^-(N)$ denote the set of all I-states η_1 for which there exists an $\eta_2 \in N$ with $\eta_1 \preceq \eta_2$. Then $R_1 \trianglelefteq R_2$ if and only if $Z(R_1) \subseteq Pc^-(Z(R_2))$.

C. Probabilistic uncertainty

We have focused our attention on nondeterministic uncertainty. Our results also apply, at least in principle, to probabilistic uncertainty. In this context, the relevant derived I-space \mathcal{I}_{prob} is a space of probability distributions over X . 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 nondeterministic and probabilistic uncertainty models warrant further exploration. For example, nondeterministic uncertainty has the property that sensing can only help – actions from primitives like P_G (Example 7) or P_R (Example 6) that do not change the state will always lead to a derived I-state at least as good as the current one. Under probabilistic uncertainty, this property does not hold; sensing can sometimes increase uncertainty.

D. Finiteness of \mathcal{RP}

Another avenue for extension is to consider the case when \mathcal{RP} is not a finite set. For example, we may extend P_L (from Example 5) to a family

$$\{P_{L,\epsilon} = (S^1, \{0\}, f_{L,\epsilon}, h_{L,\epsilon}) \mid \epsilon \geq 0\} \quad (15)$$

of primitives, each using a noisy linear odometer whose error is bounded by ϵ . With reasonable choices for \mathcal{T} , κ , and \preceq , one can show that $P_{L,\epsilon_1} \preceq P_{L,\epsilon_2}$ if and only if $\epsilon_2 \leq \epsilon_1$. If \mathcal{RP} contains many such families of primitives, and we assume 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.

ACKNOWLEDGMENT

The authors thank Rob Ghrist and the anonymous reviewers for helpful suggestions. This work is supported in part by DARPA grant #HR0011-05-1-0008.

REFERENCES

- [1] E. U. Acar and H. Choset. Robust sensor-based coverage of unstructured environments. In *Proc. of IEEE IROS, Int'l Conference on Intelligent Robots and Systems*, 2001.
- [2] P. K. Agarwal, A. D. Collins, and J. L. Harer. Minimal trap design. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 3, pages 2243–2248, 2001.
- [3] S. Akella, W. Huang, K. M. Lynch, and M. T. Mason. Parts feeding on a conveyor with a one joint robot. *Algorithmica*, 26(3):313–344, March–April 2000.
- [4] D. J. Austin and P. Jensfelt. Using multiple gaussian hypotheses to represent probability distributions for mobile robot localization. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 1036–1041, 2000.
- [5] D. Avis and H. Imai. Locating a robot with angle measurements. *J. Symb. Comput.*, 10(3-4):311–326, 1990.
- [6] K. Basye and T. Dean. Map learning with indistinguishable locations. In *Proceedings Conference on Uncertainty in Artificial Intelligence*, pages 331–342. North-Holland, 1990.
- [7] 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*, pages 269–278, 1998.
- [8] 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*, pages 132–142, 1978.
- [9] R. I. Brafman, J. Y. Halpern, and Y. Shoham. On the knowledge requirements of tasks. *Artificial Intelligence*, 98(1-2):317–349, 1998.
- [10] H. Choset and J. Burdick. Sensor based planning, part I: The generalized Voronoi graph. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 1649–1655, 1995.
- [11] I. J. Cox. Blanche – an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, 1991.
- [12] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proceedings IEEE International Conference on Robotics and Automation*, 1999.
- [13] E. D. Demaine, A. López-Ortiz, and J. I. Munro. Robot localization without depth perception. In *Scandinavian Workshop on Algorithm Theory*, 2002.
- [14] B. R. Donald. On information invariants in robotics. *Artificial Intelligence*, 72(1-2):217–304, 1995.
- [15] G. Dudek, K. Romanik, and S. Whitesides. Localizing a robot with minimum travel. *SIAM J. Comput.*, 27(2):583–604, 1998.
- [16] M. Erdmann. Understanding action and sensing by designing action-based sensors. *International Journal of Robotics Research*, 14(5), 1995.
- [17] M. A. Erdmann and M. T. Mason. An exploration of sensorless manipulation. *IEEE Transactions on Robotics and Automation*, 4(4):369–379, August 1988.
- [18] K. Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10:201–225, 1993.
- [19] K. Y. Goldberg and M. T. Mason. Bayesian grasping. In *Proceedings IEEE International Conference on Robotics and Automation*, 1990.
- [20] L. J. Guibas, R. Motwani, and P. Raghavan. The robot localization problem. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *Proceedings Workshop on the Algorithmic Foundations of Robotics*, pages 269–282. A.K. Peters, Wellesley, MA, 1995.
- [21] T. Kameda, M. Yamashita, and I. Suzuki. On-line polygon search by a six-state boundary 1-searcher. April 2003. submitted for publication.
- [22] I. Kamon, E. Rivlin, and E. Rimon. Range-sensor based navigation in three dimensions. In *Proceedings IEEE International Conference on Robotics and Automation*, 1999.
- [23] J. M. Kleinberg. The localization problem for mobile robots. In *IEEE Symposium on Foundations of Computer Science*, pages 521–531, 1994.
- [24] S. Koenig, A. Mudgal, and C. Tovey. An approximation algorithm for the robot localization problem. In *Proceedings ACM-SIAM Symposium on Discrete Algorithms*, 2006.
- [25] K. N. Kutulakos, C. R. Dyer, and V. J. Lumelsky. Provable strategies for vision-guided exploration in three dimensions. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 1365–1371, 1994.
- [26] A. M. Ladd, K. E. Bekris, A. P. Rudys, D. S. Wallach, and L. E. Kavraki. On the feasibility of using wireless Ethernet for indoor localization. *IEEE Transactions on Robotics and Automation*, 20(3):555–559, June 2004.
- [27] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, UK, 2006. Also available at <http://msl.cs.uiuc.edu/planning/>.
- [28] S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proceedings IEEE International Conference on Robotics and Automation*, 2000.
- [29] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, 1979.
- [30] V. Lumelsky and S. Tiwari. An algorithm for maze searching with azimuth input. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 111–116, 1994.
- [31] M. Moll and M. Erdmann. Manipulation of pose distributions. *International Journal of Robotics Research*, 21(3):277–292, 2002.
- [32] C. Ó. Dúnlaing and C. K. Yap. A retraction method for planning the motion of a disc. *Journal of Algorithms*, 6:104–111, 1982.
- [33] J. M. O’Kane. Global localization using odometry. In *Proceedings IEEE International Conference on Robotics and Automation*, 2006.
- [34] J. M. O’Kane and S. M. LaValle. Almost-sensorless localization. In *Proceedings IEEE International Conference on Robotics and Automation*, 2005.
- [35] M. Rao, G. Dudek, and S. Whitesides. Randomized algorithms for minimum distance localization. In *Proc. Workshop on Algorithmic Foundations of Robotics*, pages 265–280, 2004.
- [36] K. Romanik and S. Schuierer. Optimal robot localization in trees. In *Proceedings Symposium on Computational Geometry*, pages 264–273, 1996.
- [37] K. Sugihara. Some location problems for robot navigation using a simple camera. *Comp. Vis., Graphics, & Image Proc.*, 42(1):112–129, 1988.
- [38] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.
- [39] B. Tovar, L. Guilamo, and S. M. LaValle. Gap Navigation Trees: Minimal representation for visibility-based tasks. In *Proceedings Workshop on the Algorithmic Foundations of Robotics*, 2004.
- [40] A. F. van der Stappen, R.-P. Berretty, K. Goldberg, and M. H. Overmars. Geometry and part feeding. In *Sensor Based Intelligent Robots*, pages 259–281, 2000.
- [41] G. Weiss, C. Wetzler, and E. von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 1994.
- [42] D.E. Whitney. Real robots don’t need jigs. In *Proceedings IEEE International Conference on Robotics and Automation*, 1986.
- [43] J. Wiegley, K. Goldberg, M. Peshkin, and M. Brokowski. A complete algorithm for designing passive fences to orient parts. *Assembly Automation*, 17(2), August 1997.