A Logical Theory of Robot Localization*

Vaishak Belle and **Hector J. Levesque**
Dept. of Computer Science
University of Toronto
Toronto, Ontario M5S 3H5, Canada
{vaishak, hector}@cs.toronto.edu

Abstract

A central problem in applying logical knowledge representation formalisms to traditional robotics is that the treatment of belief change is categorical in the former, while probabilistic in the latter. A typical example is the fundamental capability of localization where a robot uses its many probabilistic sensors to situate itself in a dynamic world. Domain designers are then left with the rather unfortunate task of abstracting probabilistic sensors in terms of categorical ones, or more drastically, completely abandoning the inner workings of sensors to black-box probabilistic tools and then interpreting their outputs in an abstract way. Building on a first-principles approach by Bacchus, Halpern and Levesque, and a recent continuous extension to it by Belle and Levesque, we provide an axiomatization that shows how localization can be realized as a basic action theory, thereby demonstrating how such capabilities can be enabled in a single logical framework.

Introduction

For the past several years, action formalisms such as the situation calculus (Reiter 2001) and dynamic logic (Demolombe 2003) have received considerable attention as a means of providing a theoretical and computational foundation for autonomous agents situated in dynamic worlds. It is a challenging problem: in the least, reasonable features of action and change such as the frame and ramification problems need addressing, but if the robot has limited information then acting, sensing, knowledge and belief change also need to be taken into account. To this end, in the case of the situation calculus, for example, one usually provides a set of logical sentences called a *basic action theory* (Reiter 2001) which explicates in a precise way the properties of the world and their relation to the agent's sensors and effectors. When that is further supported using complex actions and procedures, one obtains a powerful and general paradigm for designing intelligent agents, seen for example in (Giacomo et al. 1996; Burgard et al. 1999; Lakemeyer and Levesque 2007).

Although a tight pairing of sensor data and high-level control is indeed what is desired, typical sensor data is best treated *probabilistically* (Thrun, Burgard, and Fox 2005) while many knowledge change accounts are *categorical* (De Giacomo and Levesque 2000). A domain designer is now left with the rather unfortunate task of abstracting probabilistic sensors in terms of categorical ones, or more drastically, completely abandoning the inner workings of sensors to black-box probabilistic tools and then interpreting their outputs in an abstract way. Regardless of application domains where such a move might be appropriate, for reasons computational or otherwise, both of these limitations are very serious since they challenge the underlying theory as a genuine characterization of the agent. Other major concerns include: (a) the loss of granularity, as it is not clear at the outset which aspect of the sensor data is being approximated and by how much, and (b) the domain designer is at the mercy of her intuition to imagine the various ways sensors might get used.

A first-principles proposal by Bacchus, Halpern and Levesque (Bacchus, Halpern, and Levesque 1999), BHL henceforth, is perhaps the most general account to rectify this problem. Embedded in the usual machinery of a basic action theory, the BHL scheme enriches the situation calculus with an account of probabilistic nondeterminism. The enrichment allows us to talk about belief change in the formalism, which is compatible with earlier accounts on knowledge (Scherl and Levesque 2003) while also following Bayesian conditioning (Pearl 1988). In contrast to many probabilistic formalisms (see the penultimate section for more on this), it allows for partial specifications, *i.e.* distributions where only some of the fluents in the domain may be provided, as well as strict uncertainty (disjunctions and quantification). Recently, we (Belle and Levesque 2013a) have further extended the framework to reason about noise that is *continuous*. We take these results as encouragement to now consider the most basic capability needed for an autonomous agent to situate itself: the *localization* problem. Roughly speaking, given a spatial characterization of the robot's environment, the robot is to identify its pose (location, orientation) to a reasonable certainty using its sensors.

Localization has been addressed using a number of algorithmic techniques for more than two decades in the robotics literature (Cox 1991; Thrun, Burgard, and Fox 2005). Our objective will not be to compete with these techniques; in fact, this paper will not concern itself with algorithms at all. Rather, we want to show how localization can be understood

*This paper is a slightly abridged version of one appearing in the 2014 proceedings of the AAMAS conference.

as part of a larger effort in a single logical framework. To the best of our knowledge, this has not been attempted before. Nevertheless, we remark that owing to the first-order nature of the formalism, our account of localization, among other capabilities, is significantly more general than most, if not all, probabilistic formalisms.

The agenda for this paper will be as follows. We first introduce the preliminaries for reasoning about degrees of belief in the logical language of the situation calculus. We then iteratively develop the steps needed to localize a robot in an uncertain world. As one would expect (and desire), given the domain axiomatization, we show that localization is realized entirely within the logic in terms of belief change. Perhaps most significantly, we demonstrate how the framework subsumes probabilistic formalisms by using the full range of situation calculus successor state axioms and sensing axioms. We then discuss related work and conclude.

The Situation Calculus

The language $\mathcal{L}$ of the situation calculus (McCarthy and Hayes 1969) is a many-sorted dialect of predicate calculus, with sorts for *actions*, *situations* and *objects*. A situation represents a world history as a sequence of actions. A set of initial situations correspond to the ways the world might be initially. Successor situations are the result of doing actions, where the term $do(a, s)$ denotes the unique situation obtained on doing a in situation s. The term $do(\alpha, s)$, where α is the sequence $[a_1, \ldots, a_n]$ abbreviates $do(a_n, do(\ldots, do(a_1, s) \ldots))$. Initial situations are defined as those without a predecessor:[1]

$$Init(s) \doteq \neg\exists a, s'.\ s = do(a, s').$$

We let the constant S_0 denote the actual initial situation, and we use the variable ι to range over initial situations only.

In each model of $\mathcal{L}$, the situations can be structured into a set of trees, where the root of each tree is an initial situation and the edges are actions. In dynamical domains, we want the values of predicate and functions to vary from situation to situation. For this purpose, $\mathcal{L}$ includes *fluents* whose last argument is always a situation. Here we assume without loss of generality that all fluents are functional.

We follow some notational conventions. Free variables are assumed to be implicitly quantified from the outside. We often suppress the situation argument in a formula ϕ, or use a distinguished variable *now*. Either way, $\phi[t]$ is used to denote the formula with that variable replaced by t.

Basic action theory

Following (Reiter 2001), we model dynamic domains in $\mathcal{L}$ by means of a *basic action theory* $\mathcal{D}$ that consists of

1. sentences $\mathcal{D}_0$ that describe what is true in the initial states, including S_0;
2. precondition axioms that describe the conditions under which actions are executable;
3. successor state axioms that describe the changes to fluents on executing actions;
4. domain-independent *foundational* axioms, the details of which need not concern us here. See (Reiter 2001).

An agent reasons about actions by means of the entailments of $\mathcal{D}$, for which standard Tarskian models suffice. We assume henceforth that models *also* assign the usual interpretations to $=, <, >, 0, 1, +, \times, /, -, e, \pi$ and x^y (exponentials).[2]

Following (Belle and Levesque 2013a), in the sequel, we will be assuming that $f_1, \ldots, f_k$ are all the fluents in $\mathcal{L}$, and that they only take a single situation term as an argument. See (Belle and Levesque 2013a) for a discussion on this limitation. Note that we still allow these fluents to range over any set, including the reals.

Belief, likelihood and continuous noise

The BHL model of belief enriches the standard situation calculus to reason about noisy sensors and belief change, by building on a treatment of knowledge by Scherl and Levesque (2003). A major limitation of their work is the restriction to discrete noise, in contrast to the continuous noise usually encountered in robotics (Thrun, Burgard, and Fox 2005). This limitation has been recently lifted in (Belle and Levesque 2013a), which we briefly review below and is based on two distinguished binary fluents l and p.

The term $l(a, s)$ is intended to denote the likelihood of action a in situation s. The axioms for l vary from domain to domain (we will see an example shortly), but they have the general form of $l(A(\vec{x}), s) = u \equiv \phi_A(\vec{x}, u, s)$ which characterizes the conditions under which action type A has likelihood u in s. Readers may note that they have a form paralleling the precondition axioms.

Next, the p fluent determines a probability distribution on situations. The term $p(s', s)$ denotes the relative *density* accorded to situation s' when the agent happens to be in situation s. The properties of p in initial states, which vary from domain to domain, are specified by axioms as part of $\mathcal{D}_0$, as one would for any other functional fluent. Now, to give p the required properties, so that it behaves like a probability density, three axioms (listed in Table 1) are needed:

– (i). Assumed to be part of $\mathcal{D}_0$, this is a nonnegative constraint on p. While this is indeed a stipulation about initial states ι only, by means of the next item, the nonnegative constraint continues to hold everywhere.

– (ii). This successor state axiom states that, given an appropriate action likelihood axiom, the density of situations s' relative to $do(a, s)$ is the density of their predecessors s'' times the likelihood of a contingent on the successful execution of a at s''. One consequence of (i) and (ii) is that

[1]The formalism used in this paper is the situation calculus as characterized by Reiter in (Reiter 2001). Nevertheless, for convenience, we often introduce formula and term abbreviations that are meant to expand as $\mathcal{L}$-formulas. For example, we might introduce a new formula A by $A \doteq \phi$, where $\phi \in \mathcal{L}$. Then any expression $E(A)$ containing A is assumed to mean $E(\phi)$. Analogously, if we introduce a new term t by $t = u \doteq \phi(u)$ then any expression $E(t)$ is assumed to mean $\exists u(E(u) \wedge \phi(u))$.

[2]Alternatively, one could specify axioms for characterizing the field of real numbers in $\mathcal{D}$. Whether or not reals with exponentiation is *first-order* axiomatizable remains a major open question.

($p(s', s) > 0$) will be true only when s' and s share the same history of actions. Both of these items, in fact, are inherited from BHL.

– (iii). This sentence is to be included in $\mathcal{D}_0$ to impress exactly one initial situation for any vector of fluent values, which follows (Levesque, Pirri, and Reiter 1998) for realizing a precise space of initial situations.

In (Belle and Levesque 2013a), we showed that these 3 axioms are all that is needed to define belief and belief change in presence of continuity. If ϕ is a formula with a single free variable of sort situation, then the *degree of belief* in ϕ is simply defined as a logical term by the following abbreviation:

$$Bel(\phi, s) = u \doteq u = \frac{1}{\gamma} \int_{\vec{x}} Density(\vec{x}, \phi, s)$$

where the normalization factor γ is understood throughout as the same expression as the numerator but with ϕ replaced by *true*, $\int_x$ is a logical term formalized using second-order logic that corresponds to mathematical integration (see (Belle and Levesque 2013a)), and $Density(\vec{x}, \phi, s)$ is an abbreviation that returns the *density* associated with ϕ at s:

$$\begin{aligned}&Density(\vec{x}, \phi, do(\alpha, s)) = u \doteq\\ &\quad \exists\iota. \bigwedge f(\iota) = x \wedge \phi[do(\alpha, \iota)] \wedge u = p(do(\alpha, \iota), do(\alpha, S_0)) \vee\\ &\quad \neg[\exists\iota. \bigwedge f(\iota) = x \wedge \phi[do(\alpha, \iota)]] \wedge u = 0.\end{aligned}$$

The intuition is as follows. Using (iii), we obtain a bijection between initial situations and fluent values. By integrating over $\vec{x}$ in the usual mathematical sense, we simply pick the appropriate initial situation, test whether ϕ holds after doing α and use the corresponding p value. In this presentation, we have assumed for simplicity that all fluents take values over $\mathbb{R}$, and so for discrete fluents, one would simply replace the integral with a summation (over its possible values) where appropriate. This, then, summarizes the proposal. Basically, the following components were needed:

- abbreviations *Bel* and *Density* that expand as $\mathcal{L}$-expressions;
- an initial theory about S_0, including (iii) to accommodate multiple initial situations and p's initial constraint (i);
- action likelihood axioms using l;
- successor and precondition axioms as usual, including (ii) for p.

In the sequel, we assume that all basic action theories will include (i), (ii) and (iii).

It is worth noting that the account of belief change using *Bel* follows Bayesian conditioning (Belle and Levesque 2013a), which will be demonstrated below.

Axiomatizing Localization

One of the significant features about the BHL scheme and its continuous variant is that robot localization, among other capabilities, follows logically from a basic action theory. No new foundational axioms are necessary. In fact, localization is a certain degree of belief regarding position and orientation, and so by reasoning about belief change in terms of *projection* (Reiter 2001), the robot would get localized. On the one hand, this is perhaps expected as many state estimation techniques in robotics are based on Bayesian conditioning, but on the other, we are demonstrating this capability in a very rich first-order framework.

i. $\forall\iota, s.\, p(s, \iota) \geq 0 \;\wedge\; (p(s, \iota) > 0 \supset Init(s))$.

ii. $p(s', do(a, s)) = u \;\equiv$
$$\begin{aligned}&\exists s''\, [s' = do(a, s'') \wedge Poss(a, s'') \wedge\\ &\qquad u = p(s'', s) \times l(a, s'')]\\ &\vee \neg\exists s''\, [s' = do(a, s'') \wedge Poss(a, s'') \wedge u = 0].\end{aligned}$$

iii. $(\forall\vec{x}\exists\iota \bigwedge f_i(\iota) = x_i) \;\wedge\; (\forall\iota, \iota'. \bigwedge f_i(\iota) = f_i(\iota') \supset \iota = \iota')$.

Table 1: Axioms in $\mathcal{D}$ for p.

In this section, we develop a simple example and a basic action theory corresponding to this example. Localization will then be demonstrated in terms of logical entailments of the action theory. We think many of the features of our example are suggestive of how one would approach more complex domains. In the main, the example involves the following steps:

- a characterization of the environment (walls, doors, etc.);
- a characterization of the uncertainty of the robot about this environment (its position and orientation); and
- a characterization of the robot's actions and sensors, and how they depend on and affect the environment.

The basic action theory $\mathcal{D}$ developed for these characterizations will be built using three fluents h, v and θ that will determine the pose of the robot, a single rigid predicate *Solid* used to axiomatize the environment, two action types $move(z, w)$ and $rotate(z)$ that determine how the robot moves and how these affect the fluents using successor state axioms, a single sensing action $sonar(z)$, and convenient abbreviations that expand into formulas involving the aforementioned logical symbols. Of course, we assume $\mathcal{D}$ to also mention $Poss, l$ and p, which are distinguished $\mathcal{L}$-symbols. We reiterate that we will not need any machinery beyond Reiter's situation calculus (Reiter 2001).

Environment

The very first item on the agenda is the notion of a *map*, which for our purpose will simply mean an axiomatic formulation of the physical space. Our example is as follows. We imagine two walls that are parallel to each other and 10 units long, as in Figure 1. The one on the *extreme* left of the robot, which we refer to as WALL-E in the sequel, is without any doors, while the one that is adjacent to the robot, referred to as WALL-A, has 3 open doors. The doors extend for one unit each. We are imagining a coordinate system that has WALL-E on the Y-axis, and puts the bottom edge of WALL-E at the origin.

We develop a simple axiomatization to describe this physical space. (For more general formalizations, see (Ge and Renz 2013; Lee, Renz, and Wolter 2013), and references therein.) We think of the walls in terms of continuous solid segments, that is, WALL-E is considered to be a single chunk,

iv. $\{Solid(0, 0, 10), Solid(5, 0, 1), Solid(5, 2, 1),$ $Solid(5, 4, 3), Solid(5, 8, 2)\}$.

v. $Poss(a) \equiv true$.

vi. $h(do(a, s)) = u \equiv$
$\neg\exists z, w(a = move(z, w)) \wedge u = h(s) \ \vee$
$\exists z, w(a = move(z, w) \wedge u = \max(\delta(s), h(s) - z \cdot \cos(w)))$.

vii. $v(do(a, s)) = u \equiv$
$\neg\exists z, w(a = move(z, w)) \wedge u = ypos(s) \vee$
$\exists z, w(a = move(z, w) \wedge u = v(s) + z \cdot \sin(w))$.

viii. $\theta(do(a, s)) = u \equiv$
$\neg\exists z(a = rotate(z) \wedge u = \theta(s)) \vee$
$\exists z(a = rotate(z) \wedge u = (((\theta(s) + z) \bmod 360) - 180))$.

ix. $\{l(move(z, w), s) = 1, l(rotate(z), s) = 1\}$.

x. $l(sonar(z), s) = u \equiv$
$Blocked(s) \wedge u = \mathcal{N}(\delta/\cos(\theta) - z; 0, 1)[s] \vee$
$\neg Blocked(s) \wedge u = \mathcal{N}((\delta + \lambda)/\cos(\theta) - z; 0, 1)[s]$.

Table 2: A basic action theory for the domain.

while WALL-A is thought of as 4 components. We will be ignoring the thickness of walls for simplicity. In precise terms, let $Solid(x, y, d)$ indicate that beginning at the coordinate (x, y), one finds a solid structure of length d extending from (x, y) to $(x, y + d)$. Of course, we are using a rigid predicate because walls are stationary; for dynamic objects, such as the robot, fluents will be used. With this idea, we could characterize (say) WALL-E by including $Solid(0, 0, 10)$ in $\mathcal{D}_0$. For both walls, then, $\mathcal{D}_0$ is assumed to include the formulas (iv) from Table 2.

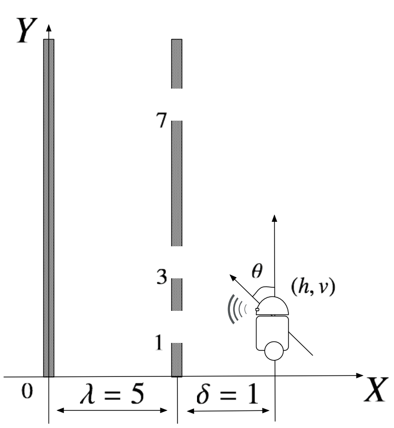

Figure 1: Two walls and a robot.

It should be clear that one may easily extract various directional and spatial relationships between such objects as appropriate. For example, although entirely obvious here, to calculate the distance between the walls, one may define an abbreviation λ as follows:

$$\lambda = u \doteq \exists x, y, d, x', y', d'.\ Solid(x, y, d) \wedge Solid(x', y', d') \wedge x \neq x' \wedge u = |x - x'|.$$

Robot: physical actions

Here, we characterize the robot's position, its world-changing actions, and their relationships.

The pose of the robot is given by three fluents: h, v and θ, where (h, v) is the robot's location, and θ is the orientation. We let θ range from -180 to 180 (degrees), with $\theta = 0$ indicating that the robot is perpendicular to WALL-E and directed towards it, and $\theta = 90$ indicating that the robot is perpendicular to the X-axis and directed towards the positive half of the Y-axis.

We imagine two physical action types at the robot's disposal, $move(z, w)$ and $rotate(z)$. We are thinking that the robot is capable of moving z units along the orientation w (degrees) wrt its angular frame. That is, for $w = 0$, the robot move would z units towards WALL-E, and for $w = 90$, the robot would move z units along the positive Y-axis, *i.e.* parallel to WALL-E. The robot can also orient itself in-place, using $rotate(z)$. For these actions, one also needs to specify their preconditions, and their likelihood axioms. For simplicity, we assume these and all other actions in domain (including the sensing action to be discussed shortly) are always executable, given by (v). Likelihood axioms may be used to specify probabilistic nondeterminism. Again, for simplicity, we consider probabilistic nondeterminism only with sensing actions, and so these physical actions are assumed to be deterministic, given by (ix).

The values of fluents change after actions, of course. The formula (ii) already specifies how p behaves in successor situations. We now do the same for h, v and θ. Since $move(z)$ and $rotate(z)$ are the only physical actions, the successor state axioms for h, v and θ will only mention these actions. They are given as (vi), (vii) and (viii) respectively. Let us consider them in order.

In the case of h, we would like (say) $move(z, 0)$ to bring the robot z units towards the wall on its left, but that motion should stop if the robot hits the wall. For this, it is perhaps easiest to first infer the distance between the robot and the closest wall on its left. This can be done as follows. For an arbitrary coordinate (x^*, y^*), we define an abbreviation for the nearest wall on its left:

$$NearestLeft(x^*, y^*) = d \doteq \exists x, y, d.\ Solid(x, y, d) \wedge y^* \in [y, y + d] \wedge \ldots \ \wedge d = (x^* - x).$$

We use $u \in [v, w]$ to mean $u \geq v \wedge u \leq w$, and the ellipsis stands for

$$\neg\exists x', y', d'.\ Solid(x', y', d') \wedge y^* \in [y', y' + d'] \ \wedge (x^* - x') < (x^* - x).$$

To now extract the distance between the *robot* and the nearest wall on its left, simply define an abbreviation δ as follows:

$$\delta(s) = u \doteq u = NearestLeft(h(s), v(s)).$$

This now allows us to dissect (vi). It says that $move(z, w)$ is the only action affecting h, thereby incorporating Reiter's monotonic solution the frame problem, and it decrements h by $z \cos(w)$ units but stops if the robot hits the nearest wall on its left. Note that, then, the value of h will become δ. For example, if $\theta = 0$, then the new value of h is simply decremented by z, and if $\theta = 180$, which would mean the robot is facing away from WALL-A then h would be incremented by z (since cos(180) is -1.)

For the fluent v, the treatment is analogous, as shown in (vii). That is, $move(z, w)$ would increment v by $z \cdot \sin(\theta)$. For example, if $z = 90$, then the move action would simply increment v since the motion would be along the Y-axis in an incremental fashion. Naturally, if one were to give a negative argument, say -3, to *move*, then the robot would move from (h, v) to $(h, v - 3)$.

Finally, θ is manipulated using $rotate(z)$ in an incremental manner while keeping its range in $[-180, 180]$ in (viii).

Robot: sensors

The robot is assumed to have a sonar unit on its frontal surface, that is, along θ. We take this sensor to be noisy. What this means is that if the robot is facing WALL-A, then a reading z from the sensor may *differ* from δ, but perhaps in some reasonable way. Most sensors have additive Gaussian noise (Thrun, Burgard, and Fox 2005), which is to say the likelihood of z is obtained from a normal curve whose mean is δ.

The complication here is that there are two walls and depending on the robot's pose, the sensor might be measuring either δ or $\lambda + \delta$. For example, if $h \in [0, 1]$ and $\theta = 0$, we understand that the sonar's signals would likely be centered around δ. However, if $v < 1$ but the robot's orientation is such that the sonar's signals advance through the gap at $[1, 2]$, then the robot's sonar unit would suggest values closer to $\delta + \lambda$ rather than δ alone. To provide a satisfactory l axiom for a sensor, let us first introduce an abbreviation for what it means for a sensor's signals to stop at WALL-A:

$$Blocked(s) \doteq \exists x, y, d.\ Solid(x, y, d) \wedge h(s) = x + \delta(s) \wedge (v + \delta \cdot \tan(\theta))[s] \in [y, y + d].$$

To make sense of this in (converse) terms of when signals would reach WALL-E, note that if $v < 1$ and yet $v + \tan(\theta) \in [1, 2]$, then the signal advances through the gap. Analogously, if $\theta < 0$ and $v > 2$ and yet $v + \tan(\theta) \in [1, 2]$, then the signal advances through as well. This then allows us to define an l axiom for the sonar in (x). Intuitively, when *Blocked* holds at situation s, we assume the sonar's reading to have additive Gaussian noise (with unit variance) centered around δ, but when the sonar's signals can reach WALL-E, we assume its reading to have additive Gaussian noise (with unit variance) centered around $\delta + \lambda$.[3]

Initial constraints

The final step is to decide on a p specification for the domain. Recall that the p fluent is used to formalize the (probabilistic) uncertainty that the robot has about the domain. This perhaps accounts for a major difference between the work here and almost all probabilistic formalisms. For us, in a sense, p is just another fluent function, allowing the domain modeler to provide incomplete and partial specifications. But since our objective in this paper will be to show, in the least, that robot localization behaves as it does in standard probabilistic formalisms, we now discuss an example with a fully known joint distribution. There are other possibilities still, a discussion of which we defer to later.

[3]The $\mathcal{N}$ term is an abbreviation for the mathematical formula defining a Gaussian density.

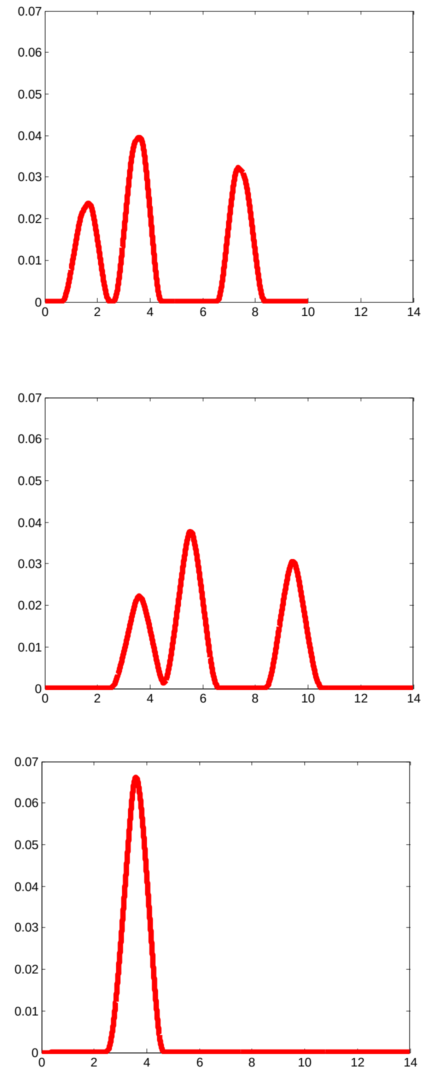

Figure 2: Belief change of v with normally distributed θ: after sensing 5.9, moving 2 units, and sensing 5.83.

Properties

Before looking at the example, let us briefly reflect on what is expected. A reasonable belief change mechanism would support the following.

- Suppose the agent believes v to be uniformly distributed on [0,10]. If the robot then uses its sonar and senses a value close to $\lambda + \delta$ say 5.9, it should come to believe that it is located at a door, which would deflate its beliefs about every point not in $[1, 2] \vee [3, 4] \vee [7, 8]$ (*i.e.* open gaps in

WALL-A.).

- Suppose the robot moves 2 units away from the X axis and then uses its sonar obtaining a reading of 5.8. It should then believe, rather confidently, that it must be in $[3,4]$ since that is the only trajectory that supports a door initially and a second door after 2 units.

  We now confirm these intuitions below.

$$\text{xi.}\quad p(\iota, S_0) = \begin{cases} .1 \times \mathcal{N}(\theta(\iota); 0, 9) & \text{if } (h = 6 \wedge v \in [0,10])[\iota] \\ 0 & \text{otherwise} \end{cases}$$

Table 3: Uncertainty about θ.

Example

We now consider a p specification where the agent will be uncertain about θ. The p we are thinking of is the one specified in Table 3. Here, the agent has the knowledge that $h = 6$, believes v is uniformly distributed on [0,10], and θ is believed to be normally distributed around 0 with a variance of 9.[4] This is a complete specification, in the sense that there is a unique joint distribution corresponding to the p axiom.

Consider for the moment what would happen after sensing once. As there is uncertainty regarding θ, it means that sensing (say) 5.9 will not imply full confidence in v being in [1,2], or [3,4], or [7,8]. Indeed, as discussed earlier, even for v values less than 1, the orientation may cause the sonar to sense WALL-E. Moreover, a larger range of θ values may cause the sonar to sense WALL-E in the [3,4] interval rather than the [1,2] interval due to its lack of wall obstructions, causing a belief density change as shown in Figure 2. After moving (say) 2 units and sensing values closer to $\lambda + \delta$ will lead to a more definite localization, as also shown in Figure 2. Here are some properties of this basic action theory:

Theorem 1: *Let $\mathcal{D}$ be a basic action theory that includes the sentences in Table 2 and Table 3. Then:*

1. $\mathcal{D} \models Bel(v \in [3, 4.57], S_0) = .157$

   Intuitively, for the numerator of *Bel*, we are to integrate a function $q(x, y, z)$ (where x corresponds to the fluent h, y corresponds to v and z corresponds to the fluent θ) that is $.1 \cdot \mathcal{N}(z; 0, 1)$ when $y \in [3, 4.57]$ and 0 otherwise. We get $\int_3^{4.57}\int_{\mathbb{R}} .1 \cdot \mathcal{N}(z; 0, 1)\, dz\, dy = .157$. The denominator is 1. (Strictly speaking, we would need to perform integration over x, y and z, but we avoided clutter and range only over y and z as the x value is fixed.)

2. $\mathcal{D} \models Bel(v \in [3, 4], do(sonar(5.9), S_0)) \approx .31$
   $\mathcal{D} \models Bel(v \in [2.8, 4.2], do(sonar(5.9), S_0)) \approx .33$

   It is worth developing this in detail. Picking the second, *Bel* expands as:

$$\frac{1}{\gamma}\int_{y\in\mathbb{R}}\int_{z\in\mathbb{R}} .1\cdot\mathcal{N}(z;0,9)\cdot\begin{cases}\mathcal{N}(.1;0,1) & \text{if } \exists\iota(\ldots\wedge\psi)[\iota]\\ \mathcal{N}(4.9;0,1) & \text{if } \exists\iota(\ldots\wedge\neg\psi)[\iota]\\ 0 & \text{otherwise}\end{cases}$$

   where the ellipses stands for:

$$h = 6 \wedge v \in [0,10] \wedge \theta = z \wedge v = y \wedge$$
$$v(do(sonar(5.9), now)) \in [2.8, 4.2]$$

   and ψ is:

$$(v+\tan\theta \in [1,2]) \vee (v+\tan\theta \in [3,4]) \vee (v+\tan\theta \in [7,8]).$$

   Note the simplification of the l values for the sensing action: $\mathcal{N}(\delta+\lambda-5.9; 0, 1) = \mathcal{N}(.1; 0, 1)$ and $\mathcal{N}(\delta-5.9; 0, 1) = \mathcal{N}(4.9; 0, 1)$.

   As pointed out earlier, what is interesting about *Bel*'s expansion is that since $\theta \neq 0$, the sensor may read $\delta + \lambda$ even if the robot is not located in $[1,2]$, $[3,4]$ and $[7,8]$. The reader may verify that if the agent believes $\theta = 0$, belief in $[3,4]$ after sensing 5.9 would be 1/3, but now, because of the uncertainty regarding θ, it is slightly less than 1/3.

3. $\mathcal{D} \models Bel(v \in [3, 4],$
   $do([sonar(5.9), move(2, 90)], S_0)) \approx .32$

   Here, the belief in $[1,2]$ after sensing 5.9, which is also .33 owing to the open door at [1,2], is transferred to $[3,4]$ after moving laterally by 2 units.

4. $\mathcal{D} \models Bel(v \in [3, 4],$
   $do([sonar(5.9), move(2, 90), sonar(5.83)], S_0)) \approx .96$

   Only v values in the vicinity of [3,4] support a second sensing reading of 5.83, which has the intended effect.

Discussions

As seen in much of the work in cognitive robotics (Lakemeyer and Levesque 2007; Reiter 2001), a logical language like the situation calculus allows for actions with complex context-dependent prerequisites and effects. But in comparison to standard (non-logical) probabilistic formalisms, the advantages of our proposal are perhaps most evident in terms of what is allowed in the initial specification of the p fluent. The example used in the paper is comparable to a unique joint probability distribution, which is standard. But that is not the case for one of the form:

$$\forall\iota(p(\iota, S_0) = \mathcal{U}(v; 0, 10)[\iota])$$
$$\vee$$
$$\forall\iota(p(\iota, S_0) = \mathcal{U}(v; 13, 23)[\iota])$$

This says that the agent believes v to be uniformly distributed on $[0, 10]$ or on $[13, 23]$, without being able to say which. As one would expect (in logic), appropriate beliefs will still be entailed, but perhaps they will not function as straightforwardly as in Table 3. For example:

- initially, it will follow that the robot is certain that $v \notin [30, 40]$, but will believe $v \in [7, 16]$ with a .3 probability;
- if sensors indicate that v is within (say) the range of [7,8], then the disjunctive uncertainty about v will no longer be significant.

[4] We use the usual "case" notation with curly braces:

$$z = \begin{cases} t_1 & \text{if } \psi \\ t_2 & \text{otherwise} \end{cases} \quad \doteq \quad (\psi \supset z = t_1) \wedge (\neg\psi \supset z = t_2).$$

Much weaker specifications are possible still, where the modeler may leave the nature of the distribution of some fluents completely open, which would correspond more closely to incomplete information in the usual non-probabilistic sense, among others. All of these are supported here.

Related Work

There are three main strands of related work from the representational aspect. They are probabilistic formalisms, relational probabilistic languages and finally action languages. We discuss them in turn, but focus our attention on robot localization where possible.

There are numerous probabilistic formalisms, see (Thrun, Burgard, and Fox 2005) for a comprehensive overview, some of which are at the heart of most traditional robotic systems. Much of the results are algorithmic in nature (Dellaert et al. 1999), in the sense of investigating sampling-based techniques, approximating domains with Gaussian distributions, and so on. At the outset, we mentioned already that this paper is about a specification. So, wrt the underlying formal characterization, almost all of these are based on Bayesian conditioning (Pearl 1988). They also assume a full specification of a joint distribution, specified compactly in the form of (say) conjugate distributions such as Gaussians or dependency structures such as Bayesian networks. Thus, in terms of methodology, none of these are geared to handle strict uncertainty, logical connectives, and partial specifications. Similar limitations also apply to early work on diagnosis in hybrid systems (McIlraith et al. 2000). Moreover, apart from a few cases such as (Darwiche and Goldszmidt 1994) and (Hajishirzi and Amir 2010) that are propositional, they do not reason about rich actions explicitly.

Logical formalisms for probabilistic reasoning, such as (Halpern 1990; Bacchus 1990), are equipped to handle features such as disjunctions and quantifiers, but they do not explicitly address actions. Relational probabilistic languages and Markov logics (Ng and Subrahmanian 1992; Richardson and Domingos 2006) also do not model actions. Recent temporal extensions, such as (Choi, Guzman-Rivera, and Amir 2011), treat special cases such as Kalman filtering, but not complex actions. Similar limitations apply to certain fuzzy logic approaches for Bayesian filtering (Jetto, Longhi, and Vitali 1999).

In this regard, action logics such as dynamic and process logics are closely related. These, and others based on the situation calculus and the fluent calculus (Thielscher 2001), in fact, are precisely the kind of logical languages we expect to be used for high-level control. But most of the work in the area, to the best of our knowledge, is limited in terms of one or more of the following: (a) they are propositional, (b) they have not been extended to handle noise that is continuous, and (c) they have not formalized and studied how localization can be realized. For example, in the area of dynamic logic, (Van Benthem, Gerbrandy, and Kooi 2009) treat probabilistic nondeterminism, but (a), (b) and (c) hold here. Related frameworks (Halpern and Tuttle 1993), including recent probabilistic planning languages (Kushmerick, Hanks, and Weld 1995; Younes and Littman 2004; Sanner 2011), are also ones where (a), (b) and (c) hold. Finally, proposals based on the situation and fluent calculi are first-order (Bacchus, Halpern, and Levesque 1999; Poole 1998; Boutilier et al. 2000; Mateus et al. 2001; Shapiro 2005; Gabaldon and Lakemeyer 2007; Fritz and McIlraith 2009; Belle and Lakemeyer 2011; Thielscher 2001), but none of them deal with *continuous sensor noise*, with the exception of (Belle and Levesque 2013a) that we build on. Also, (c) holds for these.

It is worth also emphasizing the differences between our work and recent work in *Symbolic POMDPs*, which are also based on first-order formalisms such as the situation calculus (Boutilier, Reiter, and Price 2001). First, note that the most recent results on such symbolic representations (Sanner, Delgado, and de Barros 2011) assume either that the state is discrete but observations are continuous, or that the state is continuous but observations are discrete. (We allow for both states and observations to be continuous.) But there is a more fundamental difference: on the one hand, MDPs are essentially about providing a policy to be executed. We have only discussed projection. On the other hand, the framework here *explicitly* reasons about belief change in presence of strict uncertainty, thereby offering a rich account of the agent's evolving knowledge state. Thus, one should view the framework here as an underlying logic of belief upon which reward structures and such could be further specified (Boutilier et al. 2000).

Conclusions and Outlook

This paper addresses a fundamental limitation when applying logical knowledge representation formalisms to robotics. One is forced to abstract the sensing results in a categorical fashion, or much worse, abandon its inner workings. In that regard, this paper's essential contribution was to explain and suggest how the modeler may represent her domain in a basic action theory, and how that gets further used to localize a mobile robot. We think this clarification and logical study is original, and not only is it fully compatible with existing probabilistic formalisms, but goes well beyond by allowing complex action types and partial specifications. These expressive capabilities are significant, because they are the very reason why (first-order) logical languages are chosen for modeling and reasoning in the first place. Giving them up would not be preferable for many domain modelers.

There are many avenues for future work, and we highlight two new directions. First, computation. It may seem that semantic characterizations of the form offered in this paper only serve as specifications, and may not play a role in reasoning. This may be true for arbitrary action theories, but recently, we (Belle and Levesque 2013b) have shown how regression can be formulated for this extension, by means of which projection queries reduce to formulas about the initial situation. Most significantly, the dynamic components of a basic action theory will not be needed. Because of this, perhaps, one may study Monte Carlo or other sampling-based methods (Thrun, Burgard, and Fox 2005) to reason about beliefs under certain restrictions to the language. Incidentally, algorithmic techniques such as particle filters (Thrun, Burgard, and Fox 2005) are precisely the ones used for capabilities such as localization, and so perhaps investigations along the line just mentioned may serve to algorithmically relate the two frameworks in addition to a semantic relation as considered here.

Second, we only discussed projection in this paper. If one seeks to employ a formalism such as this one on an agent, one would need syntactic structures to represent complex actions and procedures. For the standard situation calculus, a programming language called GOLOG has been proposed (Reiter 2001). Only categorical beliefs are treated there, and so, an extension for probabilistic beliefs is an exciting avenue for future work.

References

Bacchus, F.; Halpern, J. Y.; and Levesque, H. J. 1999. Reasoning about noisy sensors and effectors in the situation calculus. *Artificial Intelligence* 111(1–2):171 – 208.

Bacchus, F. 1990. *Representing and Reasoning with Probabilistic Knowledge*. MIT Press.

Belle, V., and Lakemeyer, G. 2011. A semantical account of progression in the presence of uncertainty. In *Proc. AAAI*, 165–170.

Belle, V., and Levesque, H. J. 2013a. Reasoning about continuous uncertainty in the situation calculus. In *Proc. IJCAI*.

Belle, V., and Levesque, H. J. 2013b. Reasoning about probabilities in dynamic systems using goal regression. In *Proc. UAI*.

Boutilier, C.; Reiter, R.; Soutchanski, M.; and Thrun, S. 2000. Decision-theoretic, high-level agent programming in the situation calculus. In *Proc. AAAI*, 355–362.

Boutilier, C.; Reiter, R.; and Price, B. 2001. Symbolic dynamic programming for first-order MDPs. In *Proc. IJCAI*, 690–697.

Burgard, W.; Cremers, A. B.; Fox, D.; Hähnel, D.; Lakemeyer, G.; Schulz, D.; Steiner, W.; and Thrun, S. 1999. Experiences with an interactive museum tour-guide robot. *Artif. Intell.* 114(1-2):3–55.

Choi, J.; Guzman-Rivera, A.; and Amir, E. 2011. Lifted relational kalman filtering. In *Proc. IJCAI*, 2092–2099.

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

Darwiche, A., and Goldszmidt, M. 1994. Action networks: A framework for reasoning about actions and change under uncertainty. In *Proc. UAI*, 136–144.

De Giacomo, G., and Levesque, H. 2000. Two approaches to efficient open-world reasoning. In *Logic-based artificial intelligence*. Norwell, MA, USA: Kluwer Academic Publishers. 59–78.

Dellaert, F.; Fox, D.; Burgard, W.; and Thrun, S. 1999. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, 1322–1328. IEEE.

Demolombe, R. 2003. Belief change: from situation calculus to modal logic. In *Proc. Nonmonotonic Reasoning, Action, and Change (NRAC)*.

Fritz, C., and McIlraith, S. A. 2009. Computing robust plans in continuous domains. In *Proc. ICAPS*, 346–349.

Gabaldon, A., and Lakemeyer, G. 2007. ESP: A logic of only-knowing, noisy sensing and acting. In *Proc. AAAI*, 974–979.

Ge, X., and Renz, J. 2013. Representation and reasoning about general solid rectangles. In *IJCAI*.

Giacomo, G. D.; Iocchi, L.; Nardi, D.; and Rosati, R. 1996. Moving a robot: The kr & r approach at work. In *Proc. KR*, 198–209.

Hajishirzi, H., and Amir, E. 2010. Reasoning about deterministic actions with probabilistic prior and application to stochastic filtering. In *Proc. KR*.

Halpern, J. Y., and Tuttle, M. R. 1993. Knowledge, probability, and adversaries. *J. ACM* 40:917–960.

Halpern, J. 1990. An analysis of first-order logics of probability. *Artificial Intelligence* 46(3):311–350.

Jetto, L.; Longhi, S.; and Vitali, D. 1999. Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted kalman filter. *Control Engineering Practice* 7(6):763–771.

Kushmerick, N.; Hanks, S.; and Weld, D. 1995. An algorithm for probabilistic planning. *Artificial Intelligence* 76(1):239–286.

Lakemeyer, G., and Levesque, H. J. 2007. Cognitive robotics. In *Handbook of Knowledge Representation*. Elsevier. 869–886.

Lee, J.; Renz, J.; and Wolter, D. 2013. Starvars - effective reasoning about relative directions. In Rossi, F., ed., *IJCAI*. IJCAI/AAAI.

Levesque, H. J.; Pirri, F.; and Reiter, R. 1998. Foundations for the situation calculus. *Electron. Trans. Artif. Intell.* 2:159–178.

Mateus, P.; Pacheco, A.; Pinto, J.; Sernadas, A.; and Sernadas, C. 2001. Probabilistic situation calculus. *Annals of Math. and Artif. Intell.* 32(1-4):393–431.

McCarthy, J., and Hayes, P. J. 1969. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence*, 463–502.

McIlraith, S.; Biswas, G.; Clancy, D.; and Gupta, V. 2000. Hybrid systems diagnosis. In *Proc. of Workshop on Hybrid Systems: Computation and Control (HSCC 2000)*, LNCS, 282–295.

Ng, R., and Subrahmanian, V. 1992. Probabilistic logic programming. *Information and Computation* 101(2):150–201.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Poole, D. 1998. Decision theory, the situation calculus and conditional plans. *Electron. Trans. Artif. Intell.* 2:105–158.

Reiter, R. 2001. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT Press.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine learning* 62(1):107–136.

Sanner, S.; Delgado, K. V.; and de Barros, L. N. 2011. Symbolic dynamic programming for discrete and continuous state MDPs. In *Proc. UAI*, 643–652.

Sanner, S. 2011. Relational dynamic influence diagram language (rddl): Language description. Technical report, Australian National University.

Scherl, R. B., and Levesque, H. J. 2003. Knowledge, action, and the frame problem. *Artificial Intelligence* 144(1-2):1–39.

Shapiro, S. 2005. Belief change with noisy sensing and introspection. In *NRAC Workshop*, 84–89.

Thielscher, M. 2001. Planning with noisy actions (preliminary report). In *Proc. Australian Joint Conference on Artificial Intelligence*, 27–45.

Thrun, S.; Burgard, W.; and Fox, D. 2005. *Probabilistic Robotics*. MIT Press.

Van Benthem, J.; Gerbrandy, J.; and Kooi, B. 2009. Dynamic update with probabilities. *Studia Logica* 93(1):67–96.

Younes, H., and Littman, M. 2004. PPDDL 1. 0: An extension to pddl for expressing planning domains with probabilistic effects. Technical report, Carnegie Mellon University.