Interactive Bayesian Identification of Kinematic Mechanisms

Patrick R. Barragán and Leslie Pack Kaelbling and Tomás Lozano-Pérez

***Abstract*— This paper addresses the problem of identifying mechanisms based on data gathered while interacting with them. We present a decision-theoretic formulation of this problem, using Bayesian filtering techniques to maintain a distributional estimate of the mechanism type and parameters. In order to reduce the amount of interaction required to arrive at a confident identification, we select actions explicitly to reduce entropy in the current estimate. We demonstrate the approach on a domain with four primitive and two composite mechanisms. The results show that this approach can correctly identify complex mechanisms including mechanisms which are difficult to model analytically. The results also show that entropy-based action selection can significantly decrease the number of actions required to gather the same information.**

I. INTRODUCTION

Consider a household robot that can move and grasp. It arrives in a new house and must quickly learn to interact with a variety of kinematic mechanisms: cupboard doors that rotate about hinges on the left or right or that slide sideways; drawers that pull out; security latches on the front door; faucet handles that rotate or slide along multiple axes.

We would expect the robot already to know about a general class of such mechanisms, possibly articulated in terms of one degree-of-freedom primitives and ways in which they can be combined. Then, faced with a new object, we would like it to be able to grasp and attempt to move it, possibly receiving information from several modalities, including joint torques and positions, tactile feedback, and visual tracking of parts of the object. In this process, the robot should quickly be able to discover the type of mechanism it is interacting with, as well as its parameters, such as hinge location, radius of rotation, etc.

In this paper, we present a decision-theoretic formulation of this problem, using Bayesian filtering techniques to maintain a distributional estimate of the mechanism type and parameters. In order to reduce the amount of interaction required to arrive at a confident identification, we select actions explicitly to reduce entropy in the current estimate.

If the ultimate goal of the robot is to open a cupboard door or to cause water to come out of faucet, then this problem is appropriately formulated as a partially-observable Markov decision process (POMDP) [1]. Such a formulation would support optimal action exploration in service of achieving the goal. Solution of POMDPs can be computationally very difficult, so in this work, we focus on the goal of identifying the mechanism and use a greedy action-selection method.

This work was supported in part by the NSF under Grant No. 1117325. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We also gratefully acknowledge support from ONR MURI grant N00014-09-1-1051, from AFOSR grant FA2386-?10-?1-?4135 and from the Singapore Ministry of Education under a grant to the Singapore-MIT International Design Center.

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA
barragan@mit.edu, tlp@mit.edu, lpk@mit.edu

Fig. 1. Willow Garage PR2 robot manipulating revolute model

We apply this framework in a simple experimental domain with four primitive and two composite mechanisms and demonstrate in simulation that it can use position information to perform effectively and that information-driven action selection offers significant advantages. We have conducted experiments on a PR2 robot, using active exploration and position information to discriminate among the mechanisms.

II. RELATED WORK

There has been substantial previous work on kinematic identification.

Katz et al. [2] showed accurate identification of kinematic joint types (e.g. revolute, prismatic) using vision-based tracking of features on a mechanism as it is actuated. After tracking the motion of features on the object, feature clusters are formed based on their relative motion. The relative motion between clusters indicates the type of joint connecting the links of the mechanism. They consider revolute and prismatic joints between each cluster by providing models of the transforms between features on separate bodies. Katz et al. [3] use action selection methods based on relational reinforcement learning. They show that using this action selection method can significantly reduce the number of actions required to correctly identify the kinematic relationships in the structure. Their results demonstrate robust joint identification using guided action selection.

Jain and Kemp [4] use Equilibrium Point Control (EPC) to actuate some simple mechanisms. A new equilibrium point at each step is calculated to keep a manipulator hook attached to the handle of the mechanism; the equilibrium point then becomes the next commanded position. The resulting end effector position is used to estimate the kinematics of the mechanism, which is assumed to be either a revolute or a prismatic joint, lying in a plane parallel to the floor. In later work, Jain and Kemp [5] show a data-driven approach to identifying specific doors, identifying door types, and detecting manipulation events such as locked doors or collisions between doors and other objects. By collecting training data, including forces and positions, while opening various types of doors, their system can correctly identify information of interest when encountering a new door instance.

Stürm et al. [6] also developed a sophisticated approach to kinematic identification from multiple information sources. They calculated a maximum-likelihood estimate of the parameters and probabilities over models for the joint. This work was restricted to four models for joints: rigid, revolute, prismatic, and Gaussian process, which is a data-driven model for any joint that could not be explained by the other three. The Bayesian Information Criterion was used to select the best model, trading off goodness of fit with model complexity. Three types of observations were used in the work. Much of the work used fiducial markers for tracking the individual links of the kinematic bodies. By analyzing the relative motion of the markers (much like Katz et al.), they were able to correctly identify joint types. They also used vision-based marker-less tracking in some of their experiments. Finally, in conjunction with Jain and Kemp [7], they integrated gripper positioning while operating a mechanism. The controller always attempted to make a useful move for the current most likely model. Their system allowed a robot to successfully operate multiple mechanism types. Their approach does not use the action to help predict the joint type but instead relies on the data measured as actions are taken. This useful approach is employed successfully in other works (Ruhr et al. [8] and Becker et al. [9]).

Our work differs in objective and approach from this earlier work. We use a Bayes filter to identify the type of the mechanism without segmenting out or identifying the individual joints. Also, our fundamental approach does not require visual information. It can use any combination of information sources but is demonstrated in this paper using position information. In our work, we expand the range of possible mechanisms that can be identified and manipulated with only gripper-position feedback. In particular, we also consider "serial" mechanisms, such as latches, that behave differently in different parts of their configuration spaces. As implied in the name, these latches have a fixed constraint that their handles can be placed in to restrict their motion. These mechanisms also include unilateral constraints which can be difficult to model analytically. We also choose actions (using entropy-based action selection) to gather information quickly about the type of mechanism instead of directly attempting to control the mechanism as if it were of the most likely type. This exploratory action selection also allows the system to start with no information about how to move, even initially, and to discover appropriate motions through trial and error.

III. PROBLEM FORMULATION

We formulate the mechanism as a discrete-state, Input-Output Hidden Markov Model (IOHMM), in which the hidden states are the states of the mechanisms, inputs are actions taken by the robot, and outputs are the sensed data.

A. Model specification

The *state* of a kinematic system is a tuple $S = \langle m, \theta, x \rangle$, where:

- m is the *type* of the mechanism. The type specifies a particular class of the object. These types need not simply be serial linkages. In general, a model class could describe a wide range of possible objects (e.g. soft bodies or systems of objects). In this work, we will focus on kinematic mechanisms. These objects can have any number of degrees of freedom, including "sequential" mechanisms, such as latches, in which some degrees of freedom may only be accessible when others are in a specific range of configurations. In this work, we will enumerate a finite set of types in advance.
- θ is a vector of *parameters* of the mechanism. Objects of a given type m will be characterized by a finite set of (typically continuous) parameters. Examples would be the position of the pivot and radius of a revolute joint or axis specifications of a prismatic joint.
- x is a vector of *variables* that specify a particular current configuration of the mechanism. Examples would be the current angle of a revolute joint or Cartesian position of a free mechanism.

For the purposes of identifying a mechanism, the state space will consist of a set of $\langle m, \theta, x \rangle$ tuples, with the dimensionalities of θ and x depending on the particular type m.

The action space A might include any set of primitive actions a robot could take to get information about or change the state of the mechanism it is interacting with, including looking at it from different viewpoints, pushing, pulling, etc. The only requirements on the space of actions are:

- Any action may be attempted on any mechanism type with any parameter and variable values.
- All actions always terminate in finite time.

The observation space O might include inputs from any of the robot's sensors, including vision, tactile, force, position, and vibration sensors.

Once the spaces are defined, we must characterize their relationships. The *transition model* describes the effects of actions on the state of the system as a conditional probability distribution over values of the state at time $t + 1$ given the state at time t and the action executed at time t. This is a discrete-time model that assumes each primitive action is run to completion. We can factor the transition model into two components: the probability that the mechanism type or parameters change (for example, because the robot breaks a constraint on the mechanism, or because the mechanism becomes jammed) and the probability that the variables change, given the mechanism type and parameters.

$$\begin{aligned}\Pr(s_{t+1} \mid s_t, a_t) &= \Pr(m_{t+1}, \theta_{t+1}, x_{t+1} \mid m_t, \theta_t, x_t, a_t) \\ &= \Pr(m_{t+1}, \theta_{t+1} \mid m_t, \theta_t, x_t, a_t) \\ &\quad \cdot \Pr(x_{t+1} \mid m_{t+1}, \theta_{t+1}, x_t, a_t)\end{aligned}$$

The characterization of type and parameter change,

$$\Pr(m_{t+1}, \theta_{t+1} \mid m_t, \theta_t, x_t, a_t) \ ,$$

might be written by hand or learned from very long-term interactions. We expect that, for now, it will be sufficient to use a distribution that keeps m and θ at their previous values with high probability and offers a near uniform change to other values. The normal operation of the mechanism is characterized by

$$\Pr(x_{t+1} \mid m_{t+1}, \theta_{t+1}, x_t, a_t) \ ,$$

which is a model of how the variables of a mechanism of type m with parameters θ change in response to actions of the robot. Such models may be described analytically, through the use of forward simulation or with data-driven approaches.

The *observation model* specifies a distribution over elements of O, given the current state and previous action:

$$\Pr(o_{t+1} \mid m_{t+1}, \theta_{t+1}, x_{t+1}, a_t) \ .$$

Exactly how this model is to be specified depends on the sensors involved. Note that o can be a vector of observations from different sensing modalities.

B. Bayesian inference

We will focus on the problem of identifying the type and parameters of the mechanism that the robot is interacting with. Having made such an identification, the robot would then be able to plan to manipulate the mechanism to achieve multiple goals within the operating space of the mechanism, such as opening or closing a door.

Given a sequence of actions, $a_{0:T-1} = a_0, \ldots, a_{T-1}$, and observations made as a result of those actions, $o_{1:T} = o_1, \ldots, o_T$, as well as a distribution characterizing the *a priori* belief in the different elements of the state space, $\Pr(s_0)$, we are interested in computing the posterior distribution over types and parameters, which is obtained by marginalizing out the variables at time T.

$$\begin{aligned}\Pr(m_T, \theta_T \mid a_{0:T-1}, o_{1:T}) &= \\ &\sum_{x_T} \Pr(m_T, \theta_T, x_T \mid a_{0:T-1}, o_{1:T}) \\ &= \sum_{x_T} \Pr(s_T \mid a_{0:T-1}, o_{1:T})\end{aligned}$$

This expression can be further seen as a marginalization over the states of the mechanism from times 0 through $T-1$:

$$\sum_{x_T} \sum_{s_{0:T-1}} \Pr(s_{0:T} \mid a_{0:T-1}, o_{1:T})$$

Then, we can use Bayes' rule and the conditional independence relationships in a IOHMM to write

$$\begin{aligned}&\Pr(m_T, \theta_T \mid a_{0:T-1}, o_{1:T}) \propto \\ &\sum_{x_T} \sum_{s_{0:T-1}} \Pr(s_0) \prod_{t=0}^{T-1} \Pr(o_{t+1} \mid s_{t+1}, a_t) \Pr(s_{t+1} \mid s_t, a_t)\end{aligned}$$

Depending on particular representational choices, this quantity can typically be computed efficiently using dynamic programming or recursive filtering.

IV. ACTIVE EXPLORATION

Random or arbitrary selection of actions is rarely efficient, in the sense that it may require a long sequence of actions to effectively determine the type and parameters of the mechanism the robot is interacting with. We can articulate the goal of identifying the mechanism as desiring that the conditional entropy,

$$H(m_T, \theta_T \mid a_{0:T-1}, o_{1:T}) \ ,$$

be below some desired value. The entropy of a random variable is a measure of its disorder or lack of information; in the discrete case, it is

$$H(X) = -\sum_x \Pr(x) \log_2 \Pr(x) \ .$$

We might have an additional implicit goal that the robot not change the type or parameters of the mechanism; this would prevent the robot from ripping the door off its hinges in order to reduce it to an easily identifiable free body.

Finding an optimal strategy for selecting observations to reduce overall entropy is as difficult as solving a POMDP. For efficiency reasons, we will pursue a myopic action-selection strategy. Given a belief state b, which is a probability distribution over the state of the mechanism at time T given all previous actions and observations, the objective is to select the action that, in expectation over observations it may generate, will result in the belief state b' with the lowest entropy. That is:

$$a_T^* = \min_a E_{o|b,a} H(s_{T+1} \mid a_{0:T-1}, o_{1:T}, a, o) \ .$$

Depending on the sizes of the spaces, this can be difficult to calculate analytically; we describe a sampling method in section V.

In problems that are *submodular*, this greedy strategy can be shown to be within a constant factor of optimal [10]. For a problem to be submodular, it must be that taking a particular action a_1 is never more informative after taking another action a_2 than it would have been before taking a_2. Unfortunately, that is not the case in our domain: attempting to rotate a latch may only be informative after it has been translated into a configuration in which the rotational freedom is available.

Even though it is not even boundedly suboptimal, we will find empirically that greedy action selection is a significant improvement over a random approach.

V. EXPERIMENTAL DOMAIN AND IMPLEMENTATION

We have applied this general approach to the relatively simple example problem of discriminating among 6 different types of mechanisms using position feedback resulting from manipulation of each mechanism at a designated handle. The mechanisms all operate in a plane, although this is not a fundamental limitation of the approach. The experimental results were generated using a simulated robot. Preliminary experiments using a Willow Garage PR2 robot have shown successful diagnosis of instances of several mechanism types.

a) States: The mechanism types include a completely free body, a rigidly fixed body, a 1-DOF revolute joint, a 1-DOF prismatic joint, and two sequential 2-DOF latch mechanisms. The first latch is a composite mechanism comprised of a revolute joint followed by a prismatic joint. However, as implied by the name, the latch is more than simply a composite mechanism because a fixed rigid body exists in which the end of the prismatic joint can be placed. This constraint causes this model to exhibit different behaviors in different parts of its configuration space. Similarly, the second latch is comprised of a prismatic joint followed by another prismatic joint perpendicular to the first. This mechanism also has a fixed rigid body to restrict its motion in certain parts of the mechanism's configuration space. Furthermore, the prismatic joints in these models have displacement limits along their axes. These components make these simulated models a somewhat realistic representation of a mechanism that may exist in the real world.

Table I shows the types, parameters, and variables that make up the state space. Fig. 2 shows the definition of each model, its parameters, and its variables. The rotated "U" shaped boxes around the handles (represented by the large dot in each diagram) of the latch mechanisms remain fixed in space even while the mechanism moves to different configurations. For the prismatic mechanism, the dashed line representing the axis is meant to indicate that the handle can move freely along this axis. This free motion along the axis of the prismatic model is distinctly different from the constrained motion that the prismatic joints exhibit in the two latch models. A dashed axis line is absent from these diagrams to indicate the existence of these limits.

b) Actions: In this work, we will restrict our attention to physical interactions in which we assume the robot is grasping the mechanism non-rigidly (using some sort of a caging grasp, for example). We assume that the robot will work within a finite workspace. Furthermore, we assume that the robot, while moving the handle of a particular mechanism, will have no probability of generating a state in which the handle exists outside of the workspace (e.g. the robot will not throw a free body out of its reach). Actions consist of commands to move the hand to target points in the horizontal plane. The commands are executed by a proportional-derivative (PD) controller for a fixed amount of time. The time-out is such that, if no obstacles are encountered, then any target point in the robot's workspace can be reached. However, the robot may not reach the target point if the mechanism in its current state constrains the motion. We have verified in our pilot tests with the PR2 that the compliance in the robot controller and the mechanisms are such that these incomplete motions do not generate large enough motions to damage the mechanisms. When the action terminates, the gripper is commanded to its current position with the effect of "relaxing" the robot. The possible actions allowed for the robot are a discrete set of target points within its workspace.

m	θ	x
Free		x, y location
Fixed	x_{pos}, y_{pos} location	
Revolute	x_{pivot}, y_{pivot} location, r radius	$\alpha \in [0, 2\pi]$ angle
Prismatic	$x_{axis}, y_{axis}, \alpha_{axis}$ axis definition	β displacement
Latch 1	$x_{pivot}, y_{pivot}, r, \alpha_{latch}, \beta_{latch}$	α, β
Latch 2	$x_{axis}, y_{axis}, \alpha_{axis}, \beta^1_{latch}, \beta^2_{latch}$	β^1, β^2

TABLE I

SPACE OF POSSIBLE MECHANISM STATES

c) Observations: In this simple domain, we use the robot's proprioception, in the form of the observed x, y position of the object's handle (the same position as the robot's end effector) as the observation. In simulation of the true mechanism, to generate an observation, we add Gaussian noise to the simulated transitions to mimic transition and observation noise of a real system. The relationship between the commanded position and the resulting position yields information about the underlying mechanism.

A. Transition and observation models

For many idealized situations, writing down transition and observation models analytically is possible by using ideal physics models to describe the nominal next state and observation and adding Gaussian noise. However, once collisions and friction are involved, generating analytical models becomes much more difficult. Our approach in this work is to use the Bullet Physics Library (`http://www.bulletphysics.org`) to construct approximate models of the mechanisms, and then use those models to simulate the effects of actions and to model the observations. The use of simulation allows significant flexibility in terms of the possible object classes that can be modeled.

The transition and observation models used in the estimator have Gaussian noise around nominal next states and observations. In order to compensate for possible modeling errors and to better demonstrate the effects of the action-selection strategies, we use a significantly higher variance in the observation model than would be expected in the real robot.

A consequence of using a simulation rather than an analytical model is that we do not have the ability to integrate effectively over states and observations; in our implementation we have to rely on samples drawn from the simulation.

B. Implementation

We use a discrete representation of the state space, which makes the belief state a multinomial distribution. The mecha-

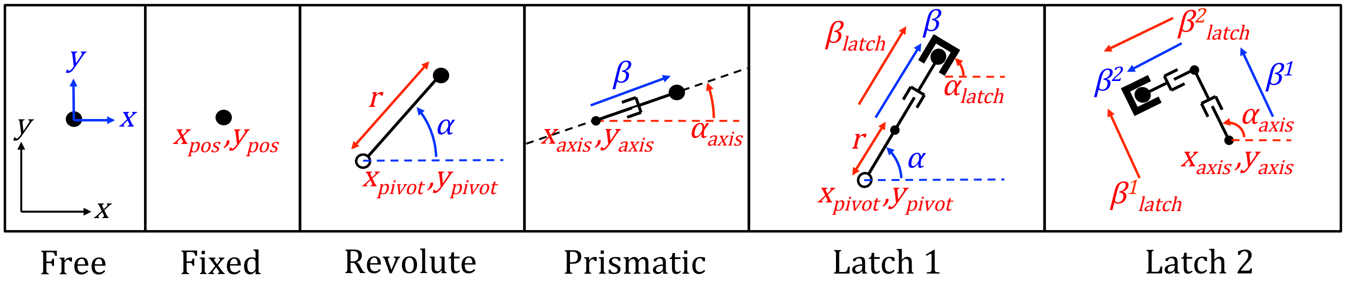


Fig. 2. Diagrams of the each of the 6 models considered. Fixed parameters are shown in red while variables are shown in blue. The large dot represents each mechanisms handle.

m	θ
Free	
Fixed	0.0, 0.0
Revolute	0.3, 0.3, 0.3$\sqrt{2}$
Prismatic	-0.16, -0.16, π/4
Latch 1	-0.2, 0.0, 0.1, 0.0, 0.1
Latch 2	-0.1, -0.1, 0.0, 0.1, 0.1

TABLE II

EXPERIMENTAL PARAMETER VALUES FOR MECHANISM TYPES

nism types are discrete, and each mechanism type has its own set of parameters and variables, the spaces of which are uniformly discretized, so that a "state" in the discrete space corresponds to a multi-dimensional "box" in parameter-variable space. An alternative strategy is to represent a continuous conditional distribution of the parameters and variables for each mechanism type. These conditional distributions are not uni-modal or otherwise easily characterized parametrically, and so the only plausible alternative representation would be as a set of samples. We let $\mathcal{S}$ be the set of discrete states;

$$s = \langle m, (\theta_{lo}, \theta_{hi}), (x_{lo}, x_{hi}) \rangle \in \mathcal{S} \enspace ,$$

and let

$$\hat{s} = \langle m, (\theta_{hi} - \theta_{lo})/2, (x_{hi} - x_{lo})/2 \rangle$$

be a canonical state value, with the values for parameters and variables chosen in the centers of their ranges.

For the following experiments, we restrict our state space to one parameter set for each of the 6 model types considered. However, adding another set of parameters for a model type is no different than adding a single parameter set for a new model type. Thus, the approach is capable of using a state space with many parameter sets for each model type, although the state-space size grows as the product of the number of parameter and variable values for each type. Each model-parameter pair has a discrete set of variable values which span the robot's workspace. The specific parameter values used for each model are given in Table II.

When the world is initialized, the robot assumes that its gripper's current position is (0,0) in its workspace which is box around its gripper in the plane in Cartesian space (x,y). The robot then performs actions relative to this position. Therefore, for any model-parameter pair to be considered with non-zero probability, the pair must have the Cartesian position (0,0) as an achievable pose relative to the robot's gripper. When simulating the "real" world, a true mechanism was chosen from among the possible mechanism types considered. Any observation from this true mechanism received added Gaussian noise. The initial variable values of each true mechanism were chosen such that the handle would begin at (0,0) as required.

The two critical computations in the system are the belief-state update and action selection.

The belief update takes a belief state from the previous time step, an action, and an observation, and computes an approximate new belief state.

BELIEFUPDATE(b, a, o):

1 $b' = \text{ZEROS}(b)$
2 **for** $s \in \mathcal{S}$
3 $\quad r = \text{SIMTRANS}(\hat{s}, a)$
4 $\quad$ **for** $s' \in \mathcal{S}$
5 $\quad\quad b'[s'] = b'[s'] + \mathcal{N}(\hat{s}', r, \sigma_{trans})$
6 **for** $s \in \mathcal{S}$
7 $\quad z = \text{SIMOBS}(\hat{s}, a)$
8 $\quad b'[s] = b'[s] \cdot \mathcal{N}(z, o, \sigma_{obs})$
9 **return** NORMALIZE(b')

Lines 2–5 compute the transition update. Given a discrete state s, we find the canonical continuous state, $\hat{s}$, and then find the nominal next continuous state, r by invoking the simulated transition model SIMTRANS. Now, we assign a transition probability from s to each discrete s' in proportion to a Gaussian density with variance σ^2_{obs}, evaluated on the distance between r and $\hat{s}'$, which is the canonical continuous value of state s'. At the end of this update, b' is an unnormalized multinomial distribution over the discrete states and represents the effects of taking action a. Lines 6–8 compute the observation update. Given a canonical continuous state $\hat{s}$ and action a, we invoke the simulator SIMOBS to find the nominal observation z. Then, we scale the posterior belief in s, $b'[s]$ by the Gaussian likelihood of making actual observation o when expecting observation z, with variance σ^2_{obs}. Finally, we divide through by the sum of the values in b' and return a multinomial distribution.

The action selection procedure uses sampling to estimate the expected entropy resulting from taking each possible action, and returns the minimizing action.

SELECTACTION(b, k):

1 $totalH[a] = 0$ for all $a \in A$
2 **for** $a \in A$
3   **for** $i \in 1..k$
4         $s \sim b$
5         $r \sim \text{GAUSSIAN}(simTrans(\hat{s}, a), \sigma_{trans})$
6         $o \sim \text{GAUSSIAN}(\text{SIMOBS}(r, a), \sigma_{obs})$
7         $b' = \text{BELIEFUPDATE}(b, a, o)$
8         $totalH[a] = totalH[a] + H(b')$
9 return $\text{ARGMIN}_a(totalH)$

For each action, we draw k sampled observations and compute the average entropy of the resulting belief states. Then we return the action that minimizes that average entropy. Lines 4–6 generate a sample observation by: sampling a discrete state s from b, then sampling a continuous resulting state r from a Gaussian centered on the nominal dynamics model applied to canonical continuous state $\hat{s}$ and action a, then sampling an observation o from a Gaussian centered on the nominal observation for continuous state r.

VI. RESULTS

Experimental results were obtained in simulation. The state space incorporated an instance of each mechanism type in many possible configurations. The filter utilized simulations in its transition step. A separate simulation with added Gaussian noise on the order of the noise levels seen from position measurements from the PR2 (1 [cm] standard deviation) was used to simulate the true mechanism. The observation covariance matrix used in the multivariate Gaussians in the observation model represented noise above this level.

Each of the 6 mechanism types was used as the true model. The latching mechanisms were initialized in their latched position. For a given experiment on a given mechanism, the robot took 10 actions, and the filter's belief state over the mechanism and parameters was recorded (after summing over all possible variable values for a given model-parameter pair). This experiment will be referred to as a trial. Each trial was repeated 10 times, and the results were averaged.

A. Single Trial

Considering a single trial on a particular mechanism can clearly illustrate the process of entropy-based action selection. Fig. 3 shows the results from a trial with the true mechanism of type Latch 2 with the parameters given in Table II for that mechanism. A model instance is a particular pair of model type and parameter set which completely define the behavior of the mechanism. In meters in Cartesian space in the plane denoted (x,y), the first three actions chosen during this trial were (0.12,-0.12), (0.12,0.12), and (-0.12,0.12).

The first action caused increased belief in the Latch 2 model instance (the same type and parameters as the true mechanism), the Revolute model instance, and the Free model instance. These three models are the only models which could come close to the position achieved by the true mechanism. Concretely, the true mechanism can almost reach the commanded position because the handle can slide out of the latching constraint and move within a short distance of the commanded point. The instances of the Free and Revolute models can also come near this position. However, the Fixed model instance cannot move. The Prismatic model instance moves along an axis that does not come near to the commanded point. Finally, the Latch 1 model instance is restricted by its latching constraint and also cannot move.

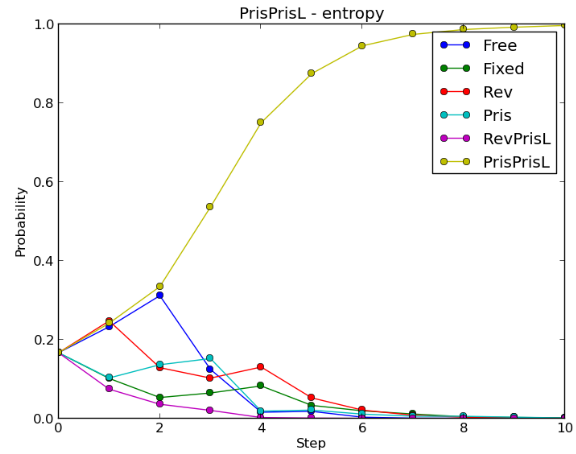
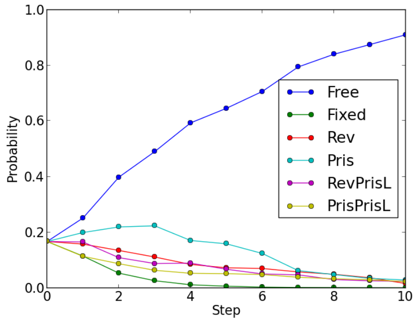


Fig. 3. Single trial for instance of the Latch 2 model type using entropy-based action selection.

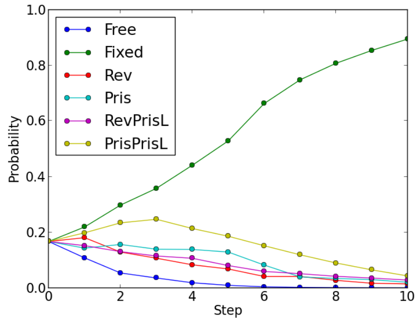
The second action chosen further increases the belief that the true mechanism is either the Free or Latch 2 model instances. However, the Revolute instance's probability decreases sharply because the commanded move takes the true mechanism's position within the radius of the Revolute instance.

The third action distinguishes between the last two instances that have high probabilities. Although the true mechanism can exist in a configuration state near the commanded target point, the mechanism begins in a position such that the action causes it to collide with the fixed latching constraint rigid body. If the true mechanism was the Free instance, the handle should move completely to the commanded point. Because the robot cannot reach the commanded point, the observation obtained drastically drops the belief in the Free instance leaving only the Latch 2 instance with high probability. The following actions further solidify the belief in the Latch 2 instance and thus correctly identifies the true mechanism.

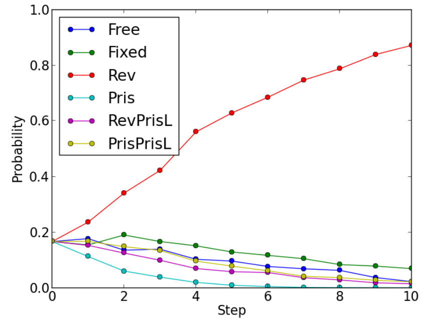
This example shows an important result. Considering the previous configuration of the mechanism, the system can choose an action for high information gain that may not have been a useful action in other possible configurations of the mechanisms considered. This is an illustration of the usefulness of belief state entropy-based action selection. By choosing actions carefully, the system is able to distinguish the true model from the other possibilities with relative few actions compared to random exploration.



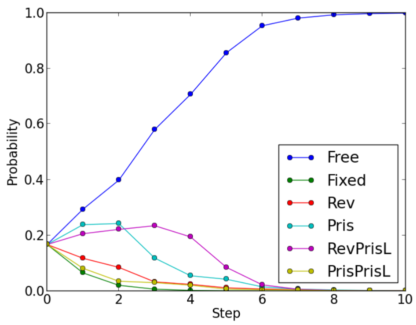
(a) Free - Random



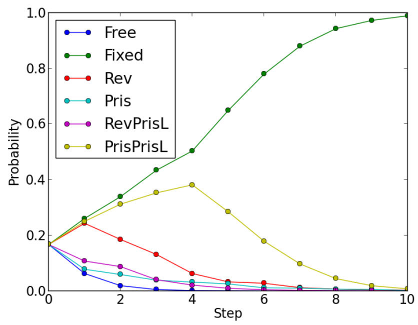
(b) Fixed - Random



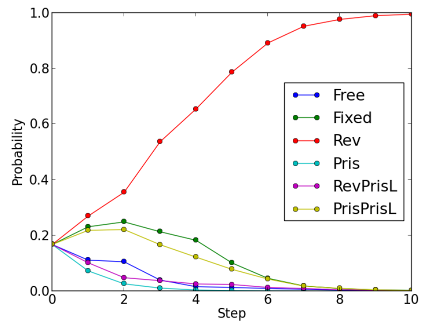
(c) Revolute - Random



(d) Free - Entropy

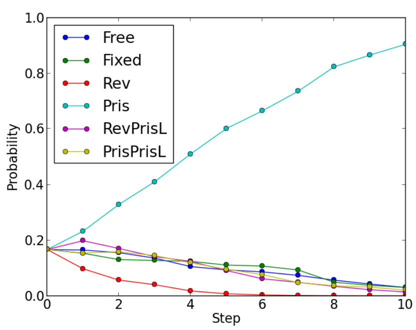


(e) Fixed - Entropy

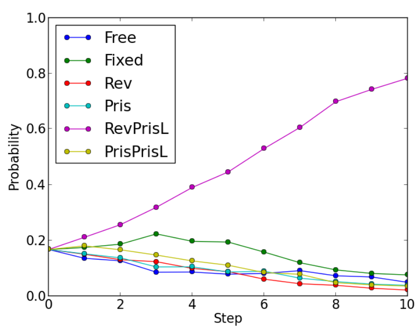


(f) Revolute - Entropy

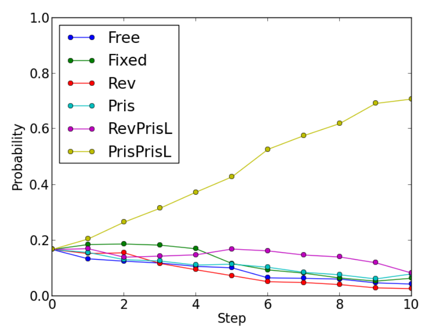
Fig. 4. Filter convergence and random vs. entropy-based action selection from Free, Fixed, and Revolute models.



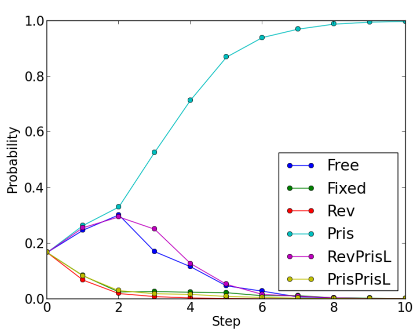
(a) Prismatic - Random



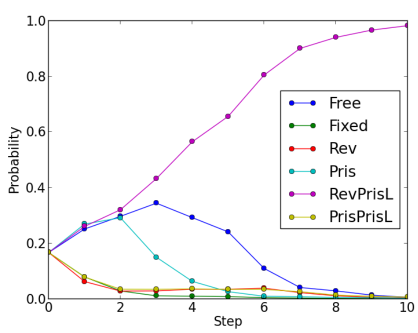
(b) Latch 1 - Random



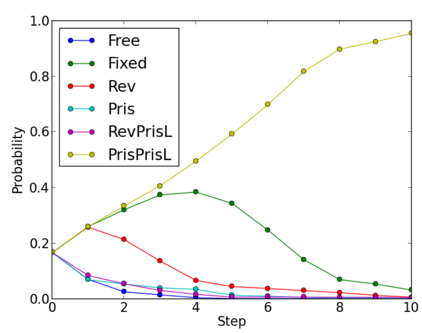
(c) Latch 2 - Random



(d) Prismatic - Entropy



(e) Latch 1 - Entropy



(f) Latch 2 - Entropy

Fig. 5. Filter convergence and random vs. entropy-based action selection from Prismatic, Latch 1, and Latch 2 models.

B. Full Trials

Fig. 4 and Fig. 5 show two main results. The top row of each figure shows the averaged experimental results for random action selection while the bottom row shows the results for entropy-based action selection. In nearly all cases (for each mechanism and each type of action selection), the true model-parameter set became much more likely than the other possibilities after less than 10 actions and, in most cases, after only a few. Moreover, the speed of convergence is significantly improved by the entropy-based action selection. In these experiments, entropy-based action selection requires between half and two-thirds of the number of actions required by random action selection to reach the same confidence level.

Using physics simulations, the filter was able to robustly predict the two different latch mechanisms. These mechanisms can experience collisions with the fixed part of the latch. Second, the latches themselves are built with limits to make them realistic models of latches found in the real world. For example, Latch 1 can only be extended to a certain distance even outside of the locking constraint. These limits are a type of unilateral constraint which can be hard to model analytically. However, the use of simulations can more easily model many complex real world systems.

Of the 120 experiments ran, a single experiment utilizing random action selection misclassified the mechanism. In this experiment, the true Latch 1 model was misclassified as the fixed mechanism after 10 actions because no selected action moved the handle out of the latch's constraint. After 10 actions, the models had nearly-identical probability, but the fixed mechanism was slightly favored due to discretization error. This result emphasizes the benefit of entropy-based action selection as it would have purposefully chosen actions to disambiguate between the two models in this scenario.

These simulation results do not test variations in the parameter values of the model types. Many different instances of each model type could be added to the state space allowing the filter to decide within a model type which parameter set is more likely. However, the state-space size scales as the product of the number of variable and parameter values, summed over types. The filter update time grows as the square of the state-space size. Action selection requires a filter update for each sample of the belief state for each action considered. Thus, with a sufficiently large state space, this method becomes computationally intractable.

Initial experiments have been conducted on the Willow Garage PR2 robot. The results suggest that the robot is able to distinguish between instances of several of the models. The corresponding video (`http://lis.csail.mit.edu/movies/ICRA14_1756_VI_fi.mp4`) shows examples of these successful trials. While the real world experiments are very promising, they can fail due to factors that are not present in the simulations. For example, out-of-plane displacements can cause the mechanism to bind. In the future, some combination of more realistic simulations and more robust execution is called for.

VII. CONCLUSION

Manipulation can be an extremely useful information-gathering tool. Robots attempting to act robustly in new environments can use many sensor modalities to identify the objects around them. This paper presents an approach to identifying mechanisms based on manipulation data. The approach uses a Bayesian filter to estimate the probability of different model types, their parameters, and their variables. The filter utilizes simulations in its transition model to allow for complex mechanisms which may be hard to model or could require large amounts of data to represent. Our approach correctly distinguishes the mechanism types used including two latching mechanisms which exhibit different constraints in different parts of their configuration spaces.

Moreover, action selection based on decreasing belief-state entropy is utilized and shown to significantly decrease the number of required actions to gather the same information. This approach attempts to choose actions which will most distinguish model-parameter pairs from others and thus explicitly attempts to gather useful information.

Our desire is to extend this work to include a larger variety of complex models and to integrate multiple sensor modalities. To effectively increase the model space, we may need to experiment with a more adaptive discretization method. The current approach becomes computationally intractable for large state spaces due to the exponential growth is number of discrete states. We plan to move to a hybrid state representation that leaves some dimensions continuous; this should drastically increase the speed of our filter update. Ultimately, we wish to learn transition and observation models from experience, so that novel mechanisms can be explored and understood by the robot.

REFERENCES

[1] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, vol. 21, pp. 1071–1088, 1973.

[2] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *ICRA*, 2008.

[3] D. Katz, Y. Pyuro, and O. Brock, "Learning to manipulate articulated objects in unstructured environments using a grounded relational representation," in *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008, pp. 254–261.

[4] A. Jain and C. C. Kemp, "Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with equilibrium point control," in *ICRA*, 2010.

[5] ——, "Improving robot manipulation with data-driven object-centric models of everyday forces," *Autonomous Robots*, pp. 1–17, 2012.

[6] J. Sturm, C. Stachniss, and W. Burgard, "A probabilistic framework for learning kinematic models of articulated objects," *J. Artif. Intell. Res. (JAIR)*, vol. 41, pp. 477–526, 2011.

[7] J. Sturm, A. Jain, C. Stachniss, C. C. Kemp, and W. Burgard, "Operating articulated objects based on experience," in *IROS*, 2010.

[8] T. Ruhr, J. Sturm, D. Pangercic, M. Beetz, and D. Cremers, "A generalized framework for opening doors and drawers in kitchen environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3852–3858.

[9] J. Becker, C. Bersch, D. Pangercic, B. Pitzer, T. Rühr, B. Sankaran, J. Sturm, C. Stachniss, M. Beetz, and W. Burgard, "The pr2 workshop-mobile manipulation of kitchen containers," in *IROS workshop on results, challenges and lessons learned in advancing robots with a common platform*, 2011.

[10] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *AAAI*, 2007, pp. 1650–1654.