

Learning Hybrid Object Kinematics for Efficient Hierarchical Planning Under Uncertainty

Ajinkya Jain¹ and Scott Niekum¹

Abstract—Sudden changes in the dynamics of robotic tasks, such as contact with an object or the latching of a door, are often viewed as inconvenient discontinuities that make manipulation difficult. However, when these transitions are well-understood, they can be leveraged to reduce uncertainty or aid manipulation—for example, wiggling a screw to determine if it is fully inserted or not. Current model-free reinforcement learning approaches require large amounts of data to learn to leverage such dynamics, scale poorly as problem complexity grows, and do not transfer well to significantly different problems. By contrast, hierarchical POMDP planning-based methods scale well via plan decomposition, work well on novel problems, and directly consider uncertainty, but often rely on precise hand-specified models and task decompositions. To combine the advantages of these opposing paradigms, we propose a new method, MICAH, which given unsegmented data of an object’s motion under applied actions, (1) detects changepoints in the object motion model using action-conditional inference, (2) estimates the individual local motion models with their parameters, and (3) converts them into a hybrid automaton that is compatible with hierarchical POMDP planning. We show that model learning under MICAH is more accurate and robust to noise than prior approaches. Further, we combine MICAH with a hierarchical POMDP planner to demonstrate that the learned models are rich enough to be used for performing manipulation tasks under uncertainty that require the objects to be used in novel ways not encountered during training.

I. INTRODUCTION

Robots working in human environments need to perform dexterous manipulation on a wide variety of objects. Such tasks typically involve making or breaking contacts with other objects, leading to sudden discontinuities in the task dynamics. Furthermore, many objects exhibit configuration-dependent dynamics, such as a refrigerator door that stays closed magnetically. While the presence of such nonlinearities in task dynamics can make it challenging to represent good manipulation policies and models, if well-understood, these nonlinearities can also be leveraged to improve task performance and reduce uncertainty. For example, when inserting a screw into the underside of a table, if direct visual feedback is not available, indirect feedback from wiggling the screw (a semi-rigid connection between the screw and the table) can be leveraged to ascertain whether the screw is inserted or not. In other words, the sensed change in dynamics (from free-body motion to rigid contact) serves as a landmark, partially informing the robot about the state of the system and reducing uncertainty. Such dynamics can be naturally represented as hybrid dynamics models or *hybrid*

automata [1], in which a discrete state represents which continuous dynamics model is active at any given time.

Current model-free reinforcement learning approaches [2]–[5] can learn to cope with hybrid dynamics implicitly, but require large amounts of data to do so, scale poorly as the problem complexity grows, face representational issues near discontinuities, and do not transfer well to significantly different problems. Conversely, hierarchical POMDP planning-based methods [5]–[8] can represent and reason about hybrid dynamics directly, scale well via plan decomposition, work well on novel problems, and reason about uncertainty, but typically rely on precise hand-specified models and task decompositions. We propose a new method, Model Inference Conditioned on Actions for Hierarchical Planning (MICAH), that bridges this gap and enables hierarchical POMDP planning-based methods to perform novel manipulation tasks given noisy observations. MICAH infers hybrid automata for objects with configuration-dependent dynamics from unsegmented sequences of observed poses of object parts. These automata can then be used to perform motion planning under uncertainty for novel manipulation tasks involving these objects.

MICAH consists of two parts, corresponding to our two main contributions: (1) a novel action-conditional inference algorithm called Act-CHAMP for kinematic model estimation and changepoint detection from unsegmented data, and (2) an algorithm to construct hybrid automata for objects using the detected changepoints and estimated local models from Act-CHAMP. Due to action-conditional inference, MICAH is more robust to noise and less vulnerable to several modes of failure than existing model inference approaches [9]–[11]. These prior approaches assume that the visual pose observations alone provide sufficient information for model estimation, which does not hold for many scenarios and can lead to poor performance. For example, an observation-only approach cannot distinguish between observations obtained by applying force against a rigid object and taking no action at all on a free body, estimating that the model is rigid in both the cases.

To evaluate the proposed method, we first show that for articulated objects, MICAH can correctly infer changepoints and the associated local models with higher fidelity and less data than a state-of-the-art observation-only algorithm, CHAMP [9]. We also consider four classes of noisy data to demonstrate its robustness to noise. Next, to test the planning-compatibility of the learned models, we learn hybrid automata for a microwave and a drawer from human demonstrations and use them with a recently proposed hier-

¹Personal Autonomous Robotics Lab (PeARL), The University of Texas at Austin. Contact: ajinkya@utexas.edu

archical POMDP Planner, POMDP-HD [6], to successfully manipulate them in new situations. Finally, we show that the learned models through MICAH are rich-enough to be leveraged creatively by a hierarchical planner for completing novel tasks efficiently—we learn a hybrid automaton for a stapler and use it to dexterously place the stapler at a target point that is reachable only through a narrow corridor in the configuration space.

II. RELATED WORKS

Learning kinematic models for articulated objects directly from visual data has been studied via different approaches in the literature [9]–[21]. Sturm et al. [12] proposed a probabilistic framework to learn motion models of articulation bodies from human demonstrations. However, the framework assumes that the objects are governed by a single articulation model, which may not hold true for all objects. For example, a stapler intrinsically changes its articulation state (e.g. rigid vs. rotational) based on the relative angle between its arms. To address this, Niekum et al. [9] proposed an online changepoint detection algorithm, *CHAMP*, to detect both the governing articulation model and the temporal changepoints in the articulation relationships of objects. However, all these approaches are observation-only and may fail to correctly infer the object motion model under noisy demonstrations or in cases, when actions are critical for inference.

In other closely related works, *interactive perception* approaches aim at leveraging the robot’s actions to better perceive objects and build accurate kinematic models [11], [17]–[19]. Katz et al. first used this approach to learn articulated motion models for planar objects [17], and later extended it to use RGB-D data to learn 3D kinematics of articulated objects [18]. Though these approaches use a robot’s actions to generate perceptual signals for model estimation, they require the robot’s interaction behavior to be pre-scripted by an expert, unlike MICAH, that can estimate models even from noisy demonstrations given by non-expert humans.

An alternative method for learning object motion models is to learn them directly from raw visual data [20], [21]. While deep neural network-based approaches have shown much potential, the biggest hurdle in using such approaches on a wide variety of real-world robotics tasks is the need for a vast amount of training data, which is often not readily available. Also, these approaches tend to transfer poorly to new tasks. In this work, we combine model learning with generalizable planning under uncertainty to address these challenges, though deep learning methods may be useful in future work, in place of our more traditional perception pipeline.

III. BACKGROUND

A. Kinematic Graphs

We represent the kinematic structure for articulated objects using *kinematic graphs* [12]. A kinematic graph $G = (V_G, E_G)$ consists of a set of vertices $V_G = 1, \dots, p$, corresponding to the p parts of the articulated object, and a set of undirected edges $E_G \subset V_G \times V_G$, each describing

the kinematic link between two object parts. An example kinematic graph for a microwave is shown in Figure 1. Sturm et al. [12] proposed to associate a single kinematic link model \mathcal{M}_{ij} with model parameter vector θ_{ij} with each edge. However, there are many articulated objects with links that are not governed by a single kinematic link model. For example, in most configurations, a microwave door is a revolute joint with respect to the microwave; however, due to the presence of a latch, this relationship changes to a rigid one when the door is closed. In this work, we extend kinematic graphs so that they can represent the hybrid kinematic structure of such objects (see Figure 2).

B. Changepoint Detection

Given a time series of observations $\mathbf{y}_{1:n}$, a changepoint model introduces a number of temporal *changepoints* τ_1, \dots, τ_m that split the data into a set of disjoint segments, with each segment assumed to be governed by a single model (though different models can govern different segments). We build on the online MAP (maximum *a posteriori*) changepoint detection model proposed by Fearnhead and Liu [22], which was specialized for detecting motion models for articulated objects by Niekum et al. [9]. Given a time series of observations $\mathbf{y}_{1:n}$ and a set of parametric candidate models M , the changepoint model infers the MAP set of changepoint times $\tau = \{\tau_0, \tau_1, \dots, \tau_m, \tau_{m+1}\}$ where $\tau_0 = 0$ and $\tau_{m+1} = n$, giving us $m + 1$ segments. Thus, the k^{th} segment consists of observations $\mathbf{y}_{\tau_k+1:\tau_{k+1}}$, and has an associated model $\mathcal{M}^k \in \mathbb{M}$ with parameters θ^k .

Assuming that the data after a changepoint is independent of the data prior to that changepoint, we model the position of changepoints in the time series as a Markov chain in which the transition probabilities are defined by the time since the last changepoint,

$$p(\tau_{i+1} = t | \tau_i = j) = \beta(t - j) \quad (1)$$

where $\beta(\cdot)$ is a probability distribution over time. For a segment from time s to t , the model evidence for the governing model being \mathcal{M} , is defined as:

$$L(s, t, \mathcal{M}) = p(\mathbf{y}_{s+1:t} | \mathcal{M}) = \int p(\mathbf{y}_{s+1:t} | \mathcal{M}, \theta) p(\theta) d\theta \quad (2)$$

The distribution over the position of the most recent changepoint prior to time t , C_t , can be efficiently estimated using the standard Bayesian filtering recursions and an online Viterbi algorithm [22]. We define E_s as the event that given a changepoint at time s , the MAP choice of changepoints has occurred prior to time s . Then, the probability of having a changepoint at time t , P_t , is defined as:

$$P_t = p(C_t = s, \mathcal{M}, E_s, \mathbf{y}_{1:t}) \quad (3)$$

$$P_t^{MAP} = p(\text{Changepoint at } t, E_s, \mathbf{y}_{1:t})$$

which results in

$$P_t(s, \mathcal{M}) = (1 - B(t - s - 1)) L(s, t, \mathcal{M}) p(\mathcal{M}) P_s^{MAP}$$

$$P_t^{MAP} = \max_{s, \mathcal{M}} \left[\frac{\beta(t - s)}{1 - B(t - s - 1)} P_t(s, \mathcal{M}) \right] \quad (4)$$

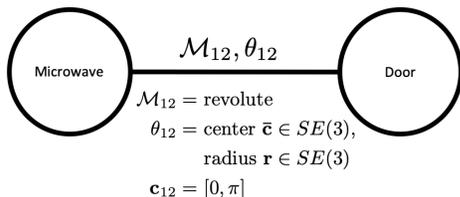


Fig. 1: Kinematic graph for microwave which considers the kinematic model as only revolute.

where $B(\cdot)$ is the cumulative distribution function of $\beta(\cdot)$. By finding the values of (s, \mathcal{M}) that maximize P_t^{MAP} , the Viterbi path can be recovered at any point. This process can be repeated until the time $t = 0$ is reached to estimate all changepoints that occurred in the given time series $\mathbf{y}_{1:T}$.

The algorithm is fully online, but requires $O(n)$ computations at each time step, since $P_t(s, \mathcal{M})$ values must be calculated for all $s < t$. The computation time is reduced to a constant by using a particle filter that keeps a constant number of particles, M , at each time step, each of which represents a support point in the approximate density $p(C_t = s, \mathbf{y}_{1:t})$. If at any time step, the number of particles exceeds M , stratified optimal resampling [22] is used to choose which particles to keep such that the Kolmogorov-Smirnov distance from the true distribution is minimized in expectation.

C. Hybrid Automaton

A hybrid automaton describes a dynamical system which evolves both in the continuous space and over a finite set of discrete states with time [1]. Formally, a hybrid automaton is a collection $\mathbf{H} = (Q, X, U, Init, f, I, \mathcal{E}, \mathcal{G}, R, \phi)$, where each discrete state $q \in Q$ of the system can be interpreted as representing a separate local dynamics model $f : \mathbf{Q} \times \mathbf{X} \times \mathbf{U} \rightarrow \mathbb{R}^N$ that governs the evolution of the continuous states $x \in X = \mathbb{R}^N$ under applied actions $u \in U$. $Init \subseteq \mathbf{Q} \times \mathbf{X}$ represents the set of initial states. The discrete state transitions can be represented as a directed graph with each possible discrete state q corresponding to a node and edges ($e \in \mathcal{E} \subseteq \mathbf{Q} \times \mathbf{Q}$) marking possible transitions between the nodes. These transitions are conditioned on the continuous states through guards $\mathcal{G} : \mathcal{E} \rightarrow 2^{\mathbf{X} \times \mathbf{U}}$. A transition from the discrete state q to another state q' happens if the continuous states \mathbf{x} are in the $\mathcal{G}(e)$ of the edge $e = (q, q') \in \mathcal{E}$. $I : \mathbf{Q} \rightarrow 2^{\mathbf{X} \times \mathbf{U}}$ assigns to each $q \in \mathbf{Q}$ an input dependent invariant set $I(q)$, such that $\forall x_0 \in I(q) \subseteq X$ there exists a control law $u_t = h(x_t)$ such that $x_t \in I(q)$ for all $t \geq 0$ and $u_t = h(x_t) \in U$. Reset map $R : \mathcal{E} \times \mathbf{X} \times \mathbf{U} \rightarrow 2^{\mathbf{X}}$ assigns to each $e = (q, q') \in \mathcal{E}$, $x \in \mathbf{X}$, and $u \in \mathbf{U}$ a map that relates the values of the continuous states before and after the discrete state transition through the edge e . The set of admissible inputs for each state $q \in \mathbf{Q}$ is defined using $\phi : \mathbf{Q} \times \mathbf{X} \rightarrow 2^{\mathbf{U}}$.

IV. APPROACH

Given a sequence of object part pose observations \mathcal{D}_y (e.g. from a visual tracking algorithm) and a sequence of applied actions \mathcal{D}_a on an articulated object, MICAH creates a planning-compatible hybrid automaton for the object. It

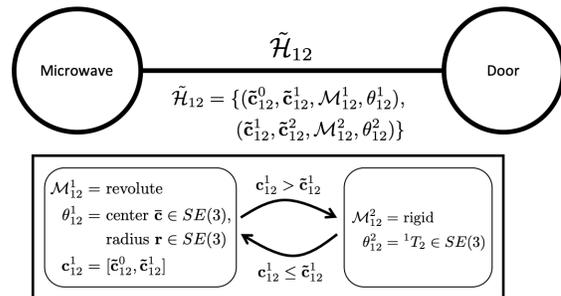


Fig. 2: Extended kinematic graph for microwave which considers a hybrid model that can be revolute or rigid, depending on the configuration.

does so in two steps: (1) it estimates the kinematic graph \hat{G} representing the kinematic structure of the object given the sequence of pose observations \mathcal{D}_y and the applied actions \mathcal{D}_a , and then (2) constructs a hybrid automaton \mathbf{H} representing the motion model for the object given \hat{G} .

For the first step, we extend the framework proposed by Sturm et al. [12] in two important ways to better learn the kinematic structure of articulated objects. First, we include reasoning about the applied actions along with the observed motion of the object while estimating its kinematic structure. Second, we extend the framework to be able to learn the kinematic structure of more complex articulated objects that may exhibit configuration-dependent kinematics, e.g., a microwave. The original framework [12] assumes that each link of an articulated body is governed by a single kinematic model. For complex articulated objects that exhibit configuration-dependent kinematics, the transition points in the kinematic model along with the set of governing local models and their parameters need to be estimated to learn the complete kinematic structure of the object.

To facilitate these extensions, we introduce a novel action-conditional changepoint detection algorithm, **Action** conditional **Changepoint** detection using **Approximate Model Parameters** (Act-CHAMP), that can detect the changepoints in the relative motion between two rigid objects (or two object parts), given a time series of observations of the relative motion between the objects and the corresponding applied actions. The algorithm is described in section IV-A.

Kinematic trees have the property that their edges are independent of each other. As a result, when learning the kinematic relationship between object parts i and j of an articulated object, only their relative transformations are relevant for estimating the edge model. MICAH first uses the Act-CHAMP algorithm to learn the kinematic relationships between different parts of the articulated object separately, and then combines them to estimate the complete kinematic graph \hat{G} for the object. Once the kinematic graph \hat{G} for an articulated object is known, MICAH constructs a hybrid automaton \mathbf{H} to represent its motion model. We choose hybrid automata as they present a natural choice to model the motion of objects that may exhibit different motion models based on their configuration. Steps to construct a hybrid

automation from the learned kinematic graph \hat{G} is described in section IV-B.

A. Action-conditional Model Inference

Following Sturm et al. [12], we define the relative transform between two objects with poses x_i and $x_j \in SE(3)$ at time t as: $\Delta_{ij,t} = \mathbf{x}_{i,t} \ominus \mathbf{x}_{j,t}$ ¹. Additionally, we define an action \mathbf{a}_t taken by the demonstrator at a time t as the intended displacement to be applied to the relative transform between two objects from time t to $t+1$ as: $\mathbf{a}_t = \Delta_{ij,t} \ominus \Delta_{ij,t+1}$. Given the time-series of observations $\mathcal{D}_{\mathbf{y}_{ij}} = \mathbf{y}_{1:T}$ of relative motion between the two object parts i and j of an articulated object and the corresponding applied actions $\mathcal{D}_{\mathbf{a}_{ij}} = \mathbf{a}_{1:T}$, we wish to find the set \mathcal{H}_{ij} defining the kinematic relationship between the two object parts. The set $\tilde{\mathcal{H}}_{ij}$ consists of tuples $(\tilde{\mathbf{c}}_{ij}^{k-1}, \tilde{\mathbf{c}}_{ij}^k, \mathcal{M}_{ij}^k, \theta_{ij}^k)$, where $\tilde{\mathbf{c}}_{ij}^{k-1}$ and $\tilde{\mathbf{c}}_{ij}^k$ denote the starting and the end configurations for the model \mathcal{M}_{ij}^k to be the governing local model with parameters θ_{ij}^k . For the sake of clarity, we drop the subscript $\{ij\}$ in the following discussion in this section.

We propose a novel algorithm, Act-CHAMP, that performs action-conditional changepoint detection to estimate the set $\tilde{\mathcal{H}}_{ij}$ given input time series of observations $\mathbf{y}_{1:T}$ and the corresponding applied actions $\mathbf{a}_{1:T}$. Act-CHAMP builds upon the CHAMP algorithm proposed by Niekum et al. [9]. The CHAMP algorithm reasons only about the observed relative motion between the objects for estimating the kinematic relationship between the objects. However, an observation-only approach can easily lead to false detection of changepoints and result in an inaccurate system model. Consider an example case of deducing the motion model for a drawer from a noisy demonstration in which the majority of applied actions are orthogonal to the axis of motion of the drawer. Due to intermittent displacements, an observation-only approach might model the motion of the drawer to be comprised of a sequence of multiple rigid joints. On the other hand, an action-conditional inference can maintain an equal likelihood of observing either a rigid or a prismatic model under off-axis actions, leading to a more accurate model.

Given the two time series inputs $\mathbf{y}_{1:T}$ and $\mathbf{a}_{1:T}$, we define the model evidence for model \mathcal{M} being the governing model for the time segment between times s and t as:

$$L(s, t, \mathcal{M}) = p(\mathbf{y}_{s+1:t} | \mathcal{M}, \mathbf{a}_{s:t-1}) = \int p(\mathbf{y}_{s+1:t} | \mathcal{M}, \theta, \mathbf{a}_{s:t-1}) p(\theta) d\theta \quad (5)$$

Each model \mathcal{M} admits two functions: a forward kinematics function, $f_{\mathcal{M},\theta}$, and an inverse kinematics function, $f_{\mathcal{M},\theta}^{-1}$, which maps the relative pose between the objects Δ_{ij} to a unique configuration \mathbf{c} for the model (e.g. a position along the prismatic axis, or an angle with respect to the axis of

¹The operators \oplus and \ominus represent motion composition operations. For example, if poses $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}^{4 \times 4}$ are represented as homogeneous matrices, then these operators correspond to matrix multiplications $\mathbf{x}_i \oplus \mathbf{x}_j = \mathbf{x}_i \mathbf{x}_j$ and its inverse multiplication, $\mathbf{x}_i \ominus \mathbf{x}_j = \mathbf{x}_i^{-1} \mathbf{x}_j$, respectively.

rotation) as:

$$\begin{aligned} f_{\mathcal{M},\theta}(\mathbf{c}_{\mathcal{M}}) &= \Delta && \text{(forward kinematics)} \\ f_{\mathcal{M},\theta}^{-1}(\Delta) &= \mathbf{c}_{\mathcal{M}} && \text{(inverse kinematics)} \end{aligned}$$

We consider three candidate models $\mathcal{M}^{\text{rigid}}$, $\mathcal{M}^{\text{revolute}}$, and $\mathcal{M}^{\text{prismatic}}$ to define the kinematic relationship between two objects. Complete definitions of forward and inverse kinematics models for these models are beyond the scope of this work; for more details, see Sturm et al. [12].

Additionally, we define the Jacobian and inverse Jacobian functions for the model \mathcal{M} as

$$\begin{aligned} J_{\mathcal{M},\theta}(\delta \mathbf{c}_{\mathcal{M}}) &= \delta \Delta && \text{(Jacobian)} \\ J_{\mathcal{M},\theta}^{-1}(\delta \Delta) &= \delta \mathbf{c}_{\mathcal{M}} && \text{(inverse Jacobian)} \end{aligned}$$

where $\delta \Delta$ and $\delta \mathbf{c}_{\mathcal{M}}$ represent small perturbations applied to the relative pose and the configuration, respectively.

Using these functions, we can define the likelihood of obtaining observations $\mathbf{y}_{1:T}$ upon applying action $\mathbf{a}_{1:T}$ under model \mathcal{M} as:

$$p(\mathbf{y}_{2:T} | \mathcal{M}, \theta, \mathbf{a}_{1:T-1}) = \prod_{t=2}^T p(\mathbf{y}_t | \hat{\Delta}_t) \quad (6)$$

where $\hat{\Delta}_t$ is the predicted relative pose under the model \mathcal{M} at time t , and can be calculated using the observation \mathbf{y}_{t-1} and applied action \mathbf{a}_{t-1} at time $t-1$ as:

$$\hat{\Delta}_t = f_{\mathcal{M},\theta}(f_{\mathcal{M},\theta}^{-1}(\mathbf{y}_{t-1}) + J_{\mathcal{M},\theta}^{-1} \mathbf{a}_{t-1}) \quad (7)$$

The probability $p(\mathbf{y}_t | \hat{\Delta}_t)$ can be calculated by defining an observation model, given an observation error covariance Σ_y for the perception system as:

$$\mathbf{y}_t \sim \begin{cases} \Delta_t + \mathcal{N}(0, \Sigma_y) & \text{if } \nu = 1 \\ \mathcal{U} & \text{if } \nu = 0 \end{cases} \quad (8)$$

where the probability of observation being an outlier is $p(\nu = 0) = \gamma$, in which case it is drawn from a uniform distribution \mathcal{U} . The data likelihood is then defined as:

$$p(\mathbf{y}_t | \Delta_t) = p(\mathbf{y}_t | \Delta_t, \gamma) p(\gamma), \quad \text{where,} \quad (9)$$

$$p(\mathbf{y}_t | \Delta_t, \gamma) = (1 - \gamma) p(\mathbf{y}_t | \nu = 1) + \gamma p(\mathbf{y}_t | \nu = 0), \quad (10)$$

$$p(\gamma) \propto e^{-w\gamma}, \quad (11)$$

and w is a weighting constant.

Finally, similar to Niekum et al. [9], we can define our BIC-penalized likelihood function as:

$$\ln L(s, t, \mathcal{M}) \approx \ln p(\mathbf{y}_{s+1:t} | \mathcal{M}, \hat{\theta}, \mathbf{a}_{s:t-1}) - \frac{1}{2} k_q \ln(t - s) \quad (12)$$

where estimated parameters $\hat{\theta}$ are inferred using MLESAC (Maximum Likelihood Estimation Sample Consensus) [23]. This likelihood function can be used in conjunction with the changepoint detection algorithm described in section III-B to infer the MAP set of changepoint times τ along with the associated local models $\mathcal{M}^k \in \mathbb{M}$ with parameters θ^k . The detected changepoints τ and the local models can be later combined appropriately to obtain a set \mathcal{H}_{ij} consisting of tuples $(\tau_{ij}^{k-1}, \tau_{ij}^k, \mathcal{M}_{ij}^k, \theta_{ij}^k)$, where τ_{ij}^{k-1} and τ_{ij}^k denote

the starting and the end changepoints for the time segment k in the input time series $\mathbf{y}_{1:T}$.

The transition conditions between the local models can be made independent of the changepoint times, τ , by making use of the observations corresponding to the changepoint times $\mathbf{y}_\tau \subseteq \mathbf{y}_{1:T}$. If an observation \mathbf{y}_{τ^k} corresponds to the changepoint τ^k denoting the transition from local model \mathcal{M}^k to the model \mathcal{M}^{k+1} , then the inverse kinematics function $f_{\mathcal{M}^k, \theta^k}^{-1}$ can be used to find an equivalent configurational changepoint $\tilde{\mathbf{c}}^k$, a fixed configuration for model \mathcal{M}^k , that marks the transition from model \mathcal{M}^k to the next model \mathcal{M}^{k+1} . We can thus convert the set \mathcal{H}_{ij} to the set $\tilde{\mathcal{H}}_{ij}$, consisting of tuples $(\tilde{\mathbf{c}}_{ij}^{k-1}, \tilde{\mathbf{c}}_{ij}^k, \mathcal{M}_{ij}^k, \theta_{ij}^k)$, that is independent of the input time series.

The complete kinematic structure of the articulated object can then be estimated by finding the set of edges E_G , denoting the kinematic connections between its parts, that maximizes the posterior probability of observing \mathcal{D}_z under applied actions \mathcal{D}_a [12]. However, to account for complex articulated objects that exhibit configuration-dependent kinematics, now each edge $\tilde{e}_{ij} \in E_G$ of the kinematic graph G can correspond to multiple kinematic link models \mathcal{M}_{ij}^k , unlike the original framework [12], in which each edge corresponds to only one kinematic link model $\mathcal{M}_{ij} \in \mathbb{M}_{ij}$. To denote the change, we call such kinematic graphs, *extended kinematic graphs*. An example extended kinematic graph for a microwave is shown in Figure 2.

B. Hybrid Automaton Construction

Hybrid automata present a natural choice for representing an articulated object that can have a discrete number of configuration-dependent kinematics models. A hybrid automaton can model a system that evolves over both discrete and continuous states with time effectively, which facilitates robot manipulation planning for tasks involving that object. We define the hybrid automaton $\mathbf{H} = (Q, X, U, \text{Init}, f, I, \mathcal{E}, \mathcal{G}, R, \phi)$ for the articulated object as:

- $Q = \prod_{\tilde{e}_{ij} \in E_G} \mathbb{M}_{ij}$, i.e. the Cartesian product of the sets of local models defining kinematic relationship between two object parts;
- $X = \prod_{\tilde{e}_{ij} \in E_G} \tilde{c}_{ij}$, where we use a single variable $\tilde{c}_{ij} \in \mathbb{R}$ to represent the configuration value $\mathbf{c}_{\mathcal{M}^k}$ under all models $\mathcal{M}_{ij}^k \in \mathbb{M}_{ij}$, as each of the candidate articulation models admits a single-dimensional configuration variable $\mathbf{c}_{\mathcal{M}} \in \mathbb{R}$;
- $U = U_C = \prod_{\tilde{e}_{ij} \in E_G} u_{ij}$, where $u_{ij} \in \mathbb{R}$ is the input delta to be applied to the continuous state \tilde{c}_{ij} and the set of discrete input variables is the null set $U_D = \emptyset$ as we cannot control the discrete states directly;
- *Init* is defined as per the task definition;
- The vector field f governing the evolution of the continuous state vector x with time is defined as $f(q, x, u) = (\mathbf{x}_t - \mathbf{x}^q) + \mathbf{u}_t$, where $q \in Q$, $\mathbf{x}_t, \mathbf{x}^q \in X$, and $\mathbf{u} \in U$. The vector $\mathbf{x}^q \in X$ is so defined that its l -th element $\mathbf{x}^q[l] = \sum_{r=0}^{k-1} \tilde{\mathbf{c}}_{ij}^r$, where l -th dimension of X corresponds to the kinematic relationship between object parts i and j with $\tilde{e}_{ij} \in E_G$, and $q[l] = \mathcal{M}_{ij}^k$;

- For each discrete state $q \in \mathbf{Q}$, an invariant set $I(q)$ is defined such that within it the time evolution of the continuous states is governed by the vector field $f(q, x, u) \forall x \in I(q) \subseteq X, u \in U$. We define I as $I = \prod_{\tilde{e}_{ij} \in E_G} \text{Dom}(\mathbb{M}_{ij})$, where $\text{Dom}(\mathbb{M}_{ij}) = \{\text{Dom}(\mathcal{M}_{ij}^k) \forall k \in |\mathbb{M}|\}$ with $\text{Dom}(\mathcal{M}_{ij}^k)$ defined as $\text{Dom}(\mathcal{M}_{ij}^k) = [0, \tilde{\mathbf{c}}_{ij}^{k+1})$;
- The set of edges defines the set of feasible transitions between the discrete states, $\mathcal{E} = \{(q, q') \mid q[l] = \mathcal{M}_{ij}^k \Rightarrow q'[l] = \mathcal{M}_{ij}^r, r \in \{k-1, k, k+1\}\}$;
- Guards \mathcal{G} can be constructed using the configurational changepoints estimated for the object. If an edge $e = (q, q') \in \mathcal{E}$ corresponds to a transition from a local model \mathcal{M}_{ij}^k to model \mathcal{M}_{ij}^{k+1} , then the guard for the edge e can be defined as $\mathcal{G}(e) = \{\tilde{c}_{ij} \geq \tilde{\mathbf{c}}_{ij}^{k+1}\}$. Analogously, the guard for the reverse transition $\mathcal{G}(e' = (q', q)) = \{\tilde{c}_{ij} < 0\}$. To handle the corner cases when $\tilde{c}_{ij} < 0$ for \mathcal{M}_{ij}^1 or $\tilde{c}_{ij} > \tilde{\mathbf{c}}_{ij}^{m+1}$ for model \mathcal{M}_{ij}^m (assuming $|\mathbb{M}_{ij}| = m$), we define two additional edges e_{ij}^0 and e_{ij}^{-1} which corresponds to the self transitions to the same discrete states such that \tilde{c}_{ij} is lower-bounded at 0 for \mathcal{M}_{ij}^1 and upper-bounded at $\tilde{\mathbf{c}}_{ij}^{m+1}$ for model \mathcal{M}_{ij}^m ;
- The reset map R is an identity map;
- The set of admissible inputs $\phi(q, \mathbf{x}) = \mathbf{U}$.

V. EXPERIMENTS AND DISCUSSIONS

In the first set of experiments, we compare the performance of Act-CHAMP with the CHAMP algorithm [9] to estimate changepoints and local motion models for a microwave and a drawer. Next, we test the complete method, MICAH, to construct planning-compatible hybrid automata for the microwave and drawer and discuss the results of manipulation experiments to open and close the microwave door and the drawer using the learned models. Finally, we show that MICAH can be combined with a recent hierarchical POMDP planner, POMDP-HD [6], to develop a complete pipeline that can learn a hybrid automaton from demonstrations and leverage it to perform a novel manipulation task—in this case, with a stapler. A video showcasing the experiments is available at: <https://youtu.be/f35gMoOoOy8>.

A. Learning Kinematics Models for Objects

We collected six sets of demonstrations to estimate motion models for the microwave and the drawer. We provided kinesthetic demonstrations to a two-armed robot, in which the human expert physically moved the right arm of the robot, while the left arm shadowed the motion of the right arm to interact with objects while collecting unobstructed visual data. The first two sets provide low-noise data, by manipulating the door handle or drawer knob via a solid grasp. The next two sets provide data in which random periods of no actions on the objects were deliberately included while giving demonstrations. The last two sets consist of high-noise cases, in which the actions were applied by pushing with the end-effector without a grasp. Relative poses of object parts were recorded as time-series observations with an RGB-D sensor using the SimTrack object tracker [24]. For each

time step t , the demonstrator’s action \mathbf{a}_t on the object was defined as the difference between the position of the right end-effector at times t and $t + 1$.

With grasp: Both algorithms (CHAMP and Act-CHAMP) detected a single changepoint in the articulated motion of the microwave door and determined the trajectory to be composed of two motion models, namely rigid and revolute. For the drawer, both algorithms were able to successfully determine its motion to be composed of a single prismatic motion model(see Table I). This demonstrates that for clean, information-rich demonstrations, Act-CHAMP can perform on par with the baseline.

No-Actions: When no action is applied to an object, due to the lack of motion, an observation-only model inference algorithm can infer the object motion model to be rigid. Moreover, if the agent stops applying actions after interacting with the object for some time, an observation-only approach can falsely detect a changepoint in the motion model. We hypothesize that an action-conditional inference algorithm such as Act-CHAMP won’t suffer from these shortcomings as it can reason that no motion is expected if no actions are applied. To test it, we conducted experiments in which the demonstrator stopped applying actions on the object midway during a demonstration for an extended time randomly at two distinct locations. As expected, the observation-only CHAMP algorithm falsely detected changepoints in the object motion model and performed poorly (see Table I). However, as Act-CHAMP reasons about the applied actions as well, it performed much better (see Table I).

Without grasp: When actions are applied directly on the object (microwave door and the drawer, respectively), the majority of the applied actions are orthogonal to the axis of motion leading to low-information demonstrations. In such a case, while CHAMP almost completely failed to detect correct motion models for the microwave (5% success), Act-CHAMP was able to correctly detect models in almost one-third of the trials (see Table I). For the drawer, CHAMP falsely detected a changepoint and determined that the articulation motion model is composed of two separate prismatic articulation models with different model parameters (Figure 4). However, due to action-conditional inference,

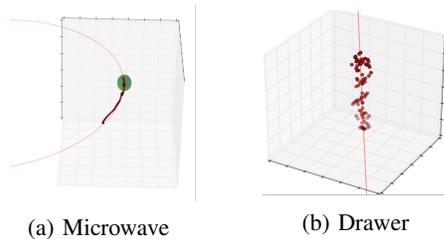


Fig. 3: Inferred motion models for the microwave and the drawer using Act-CHAMP. Points denote the recorded relative poses of object parts from one demonstration. The green circle represents the detected rigid model, the circular arc represents the detected revolute model, and the line represents the detected prismatic model.

Act-CHAMP correctly classified the motion to be composed of only one articulation model (Figure 4, see Table I).

B. Object Manipulation Using Learned Models

To test the effectiveness of the learned hybrid automata using MICAH, we used them to perform the tasks of opening and closing a microwave door and a drawer using a robot manipulator. We use the POMDP-HD planner [6] to develop manipulation plans. Figure 5 shows the belief space and actual trajectories for the microwave and drawer manipulation tasks. For both the objects, low final errors were reported: 0.05 ± 0.01 rad for the microwave and 0.005 ± 0.003 m for the drawer (average of 5 different tasks), validating the effectiveness of the learned automata.

C. Leveraging Learned Models for Novel Manipulations

Finally, we show that our learned models and planner are rich enough to be used to complete novel tasks under uncertainty that require intelligent use of object kinematics. To do so, we combine MICAH with the POMDP-HD planner for performing a manipulation task of placing a desk stapler at a target point on top of a tall stack of books. Due to the height of the stack, it is challenging to plan a collision-free path to deliver the stapler to the target location through a narrow corridor in the free configuration space of the robot; if the robot attempts to place the stapler at the target point while its governing kinematic model is revolute, the lower arm of the stapler will swing freely and collide with the obstacle. However, a feasible collision-free motion plan can be obtained if the robot first closes and locks the stapler (i.e. rigid articulation), and then proceeds towards the goal. To change the state of the stapler from revolute to rigid, the robot can plan to make contact with the table surface to press down and lock the stapler in a non-prehensile fashion.

As the task involves making and breaking contacts with the environment, we need to extend the learned hybrid motion model of the stapler to include local models due to contacts. We approximately define the contact state between the stapler and the table as to be either a line contact (an edge of the lower arm of the stapler in contact with the table), a surface contact (the lower arm lying flat on the table) or no contact. The set of possible local models for the hybrid task kinematics can be obtained by taking a Cartesian product of the set of possible discrete states for the stapler’s hybrid

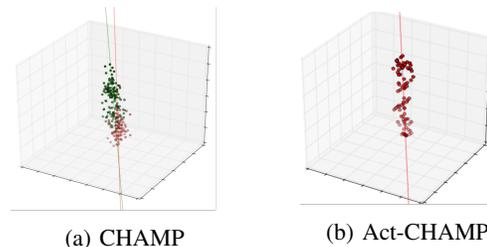


Fig. 4: Act-CHAMP correctly infers the drawer motion model, while CHAMP (baseline) falsely detects a changepoint under noisy demonstrations.

Case	Object	Algorithms	Changepoint Detection		Error in Model Parameters
			Correct No.	Position Error	
With grasp	Microwave	CHAMP	20/20 (100%)	0.001 ± 0.001	Center: 0.046 ± 0.024 m Axis: 0.02 ± 0.01 rad Radius: 0.027 ± 0.024 m
		ActCHAMP	20/20 (100%)	0.001 ± 0.001	Center: 0.066 ± 0.034 m Axis: 0.03 ± 0.02 rad Radius: 0.013 ± 0.059 m
	Drawer	CHAMP	20/20 (100%)	—	Axis: 0.04 ± 0.01 rad
		ActCHAMP	20/20 (100%)	—	Axis: 0.04 ± 0.01 rad
No Actions	Microwave	CHAMP	11/20 (55%)	0.001 ± 0.001	Center: 0.015 ± 0.006 m Axis: 0.01 ± 0.01 rad Radius: 0.011 ± 0.006 m
		ActCHAMP	14/20 (70%)	0.001 ± 0.001	Center: 0.019 ± 0.009 m Axis: 0.01 ± 0.01 rad Radius: 0.002 ± 0.014 m
	Drawer	CHAMP	4/20 (20%)	—	Axis: 0.03 ± 0.01 rad
		ActCHAMP	12/20 (60%)	—	Axis: 0.03 ± 0.01 rad
Without grasp	Microwave	CHAMP	1/20 (5%)	0.001	Center: 0.51 m Axis: 0.42 rad Radius: 0.056 m
		ActCHAMP	6/20 (30%)	0.001 ± 0.001	Center: 0.328 ± 0.125 m Axis: 0.58 ± 0.20 rad Radius: 0.164 ± 0.103 m
	Drawer	CHAMP	9/20 (45%)	—	Axis: 0.18 ± 0.01 rad
		ActCHAMP	15/20 (75%)	—	Axis: 0.14 ± 0.04 rad

TABLE I: Model detection comparison

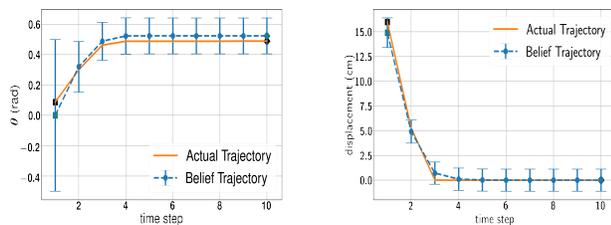
automaton and the set of possible contact states between the stapler and the table. However, if the stapler is in the rigid mode, its motion would be the same under all contact states. Hence, a compact task kinematics model would consist of four local models—the stapler in revolute mode with no contact with the table, the stapler in revolute mode with a line contact with the table, the stapler in revolute mode with a surface contact with the table, and the stapler in rigid mode.

Given a human demonstration of robot’s interaction with the stapler as input, MICAH first learns a hybrid automaton for the stapler and then extends it to the hybrid task model using the provided task-specific parameters. Next, the POMDP-HD planner uses the learned task model to develop motion plans to complete the task with minimum final state uncertainty. Note that only the final Cartesian position for the stapler was specified as the target for the task and not the articulation state of the stapler (rigid/revolute). Motion plans generated by the planner are shown in Figure 6. As can be seen from the plots, the planner plans to make contacts with the table to reduce the relative angle between the stapler arms

and change the articulation model of the stapler. The plan drags the stapler along the surface of the table, indicating that it waits until it is highly confident that the stapler has become rigid before breaking contact. Making contacts with the table along the path also helps in funneling down the uncertainty in the stapler’s location relative to the table in a direction parallel to the table plane normal, thereby increasing the probability of reaching the goal successfully. Figure 7 shows snapshots of the motion plan and actual execution of the robot performing the task.

VI. CONCLUSION

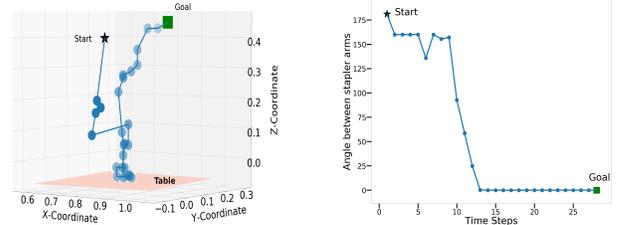
Robots working in human environments require a fast and data-efficient way to learn motion models of objects around them to interact with them dexterously. We present a novel method MICAH, that performs action-conditional model inference from unsegmented human demonstrations via a novel algorithm, Act-CHAMP, and then uses the resulting models to construct hybrid automata for articulated objects. Action-conditional inference enables articulation motion models to



(a) Microwave

(b) Drawer

Fig. 5: Plots showing belief space [blue] and actual trajectories [orange] for microwave and drawer manipulation tasks using learned models. Error bars represent belief uncertainty.



(a) Stapler: $\{x, y, z\}$

(b) Stapler: θ vs time

Fig. 6: Planned trajectories for the stapler placement experiment. (Left) in $\{x, y, z\}$ (Right) Relative angle of the stapler arms over time.



Fig. 7: Snapshots showing the executed trajectory for the stapler placement task. The red region denotes the unreachable workspace for the robot’s end-effector.

be learned with higher accuracy than the prior methods in the presence of noise and leads to the development of models that can be used directly for manipulation planning. Furthermore, we demonstrate that the learned models are rich enough to be used for performing novel tasks with such objects in a manner that has not been previously observed. One advantage of using an action-conditional model inference approach over observation-only approaches is that it can enable robots to take informative exploratory actions for learning object motion models autonomously. Hence, future work may include the development of an active learning framework that can be used by a robot to autonomously learn the motion models of objects in a small number of trials. Another promising extension to the method can be to extend it to learn hybrid automata for articulated objects with multiple joints autonomously through noisy human demonstrations.

ACKNOWLEDGEMENT

This work has taken place in the Personal Autonomous Robotics Lab (PeARL) at The University of Texas at Austin. PeARL research is supported in part by the NSF (IIS-1724157, IIS-1638107, IIS-1617639, IIS-1749204) and ONR(N00014-18-2243).

REFERENCES

- [1] J. Lygeros, S. Sastry, and C. Tomlin, “Hybrid systems: Foundations, advanced topics and applications,” *under copyright to be published by Springer Verlag*, 2012.
- [2] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, “Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning,” in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 100. PMLR, 2020, pp. 1025–1037.
- [3] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, “Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7559–7566.
- [4] D. Xu, M. Agarwal, F. Fekri, and R. Sivakumar, “Accelerating reinforcement learning agent with eeg-based implicit human feedback,” *arXiv preprint arXiv:2006.16498*, 2020.
- [5] O. Kroemer, S. Niekum, and G. Konidaris, “A review of robot learning for manipulation: Challenges, representations, and algorithms,” *arXiv preprint arXiv:1907.03146*, 2019.
- [6] A. Jain and S. Niekum, “Efficient hierarchical robot motion planning under uncertainty and hybrid dynamics,” in *Conference on Robot Learning*, 2018, pp. 757–766.
- [7] E. Brunskill, L. Kaelbling, T. Lozano-Perez, and N. Roy, “Continuous-State POMDPs with Hybrid Dynamics,” *Symposium on Artificial Intelligence and Mathematics*, pp. 13–18, 2008.

- [8] M. Toussaint, L. Charlin, and P. Poupart, “Hierarchical pomdp controller optimization by likelihood maximization,” in *UAI*, vol. 24, 2008, pp. 562–570.
- [9] S. Niekum, S. Osentoski, C. G. Atkeson, and A. G. Barto, “Online bayesian changepoint detection for articulated motion models,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1468–1475.
- [10] S. Pillai, M. R. Walter, and S. Teller, “Learning articulated motions from visual demonstration,” *arXiv preprint arXiv:1502.01659*, 2015.
- [11] R. Martín-Martín and O. Brock, “Coupled recursive estimation for online interactive perception of articulated objects,” *The International Journal of Robotics Research*, p. 0278364919848850, 2019.
- [12] J. Sturm, C. Stachniss, and W. Burgard, “A probabilistic framework for learning kinematic models of articulated objects,” *Journal of Artificial Intelligence Research*, vol. 41, pp. 477–526, 2011.
- [13] P. R. Barragán, L. P. Kaelbling, and T. Lozano-Pérez, “Interactive bayesian identification of kinematic mechanisms,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2013–2020.
- [14] C. Pérez-D’Arpino and J. A. Shah, “C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4058–4065.
- [15] G. Subramani, M. Gleicher, and M. Zinn, “Recognizing geometric constraints in human demonstrations using force and position signals,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [16] K. Hausman, S. Niekum, S. Osentoski, and G. S. Sukhatme, “Active articulation model estimation through interactive perception,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3305–3312.
- [17] D. Katz and O. Brock, “Manipulating articulated objects with interactive perception,” in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 272–277.
- [18] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz, “Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects,” in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5003–5010.
- [19] R. M. Martin and O. Brock, “Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2494–2501.
- [20] B. Abbatematteo, S. Tellex, and G. Konidaris, “Learning to generalize kinematic models to novel objects,” in *Proceedings of the Third Conference on Robot Learning*, 2019.
- [21] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song, “Category-level articulated object pose estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3706–3715.
- [22] P. Fearnhead and Z. Liu, “On-line inference for multiple changepoint problems,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 589–605, 2007.
- [23] P. H. Torr and A. Zisserman, “Mlesac: A new robust estimator with application to estimating image geometry,” *Computer vision and image understanding*, vol. 78, no. 1, pp. 138–156, 2000.
- [24] K. Pauwels and D. Kragic, “Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany, 2015.