

Online Bayesian Changepoint Detection for Articulated Motion Models

Scott Niekum^{1,2}

Sarah Osentoski³

Christopher G. Atkeson¹

Andrew G. Barto²

Abstract—We introduce CHAMP, an algorithm for online Bayesian changepoint detection in settings where it is difficult or undesirable to integrate over the parameters of candidate models. CHAMP is used in combination with several articulation models to detect changes in articulated motion of objects in the world, allowing a robot to infer physically-grounded task information. We focus on three settings where a changepoint model is appropriate: objects with intrinsic articulation relationships that can change over time, object-object contact that results in quasi-static articulated motion, and assembly tasks where each step changes articulation relationships. We experimentally demonstrate that this system can be used to infer various types of information from demonstration data including causal manipulation models, human-robot grasp correspondences, and skill verification tests.

I. INTRODUCTION

Many manipulation problems in robotics are characterized by articulation properties of objects such as rotating doors, sliding drawers, and rigidly connected parts. Several methods have recently been proposed to identify articulation relationships from visual observations [10], [17] but have assumed that these relationships remain static over time. However, much articulated motion is of a changing nature—a door can both rotate and then lock rigidly in place; a loose grasp allows for quasi-static revolute and prismatic motion of an object when external force is applied; assembly procedures aim to change articulation relationships between parts with each step. Learning generalizable policies for many tasks requires understanding the nature of these changing relationships and identifying what brings about these changes. To address this, we propose an online method to detect changes in the underlying articulation models that are generating the robot’s observations and infer the parameters of each of these models.

First, we introduce an algorithm for online Bayesian changepoint detection in settings where it is difficult to analytically or numerically integrate over the parameters of candidate models. Building on the work of Fearnhead and Liu [7], we show that with some modifications, approximate online Bayesian changepoint detection can be performed using estimates of the maximum likelihood parameters for each segment—for example, via regression or a sample consensus method. We call this new algorithm CHAMP (CHangepoint

detection using Approximate Model Parameters). This algorithm is a general contribution that is useful beyond finding changes in articulation models.

Next, we use CHAMP in conjunction with a set of articulation models (revolute, prismatic, and rigid) to detect changes in the articulated motion of objects in the world and to infer the parameterized models that describe those articulations. We examine three settings where a changepoint model is appropriate: objects with intrinsic articulation relationships that can change over time, object-object contact that results in quasi-static articulated motion, and assembly tasks where each step changes articulation relationships. Experiments are performed in a learning from demonstration setting and show how this system can be used to infer various types of information from demonstration data, including causal manipulation models, human-robot grasp correspondences, and skill verification tests.

II. RELATED WORK

A. Changepoint Detection

Hidden Markov Models (HMMs) are largely the *de facto* tool of choice when analyzing time series data, but the standard HMM formulation has several undesirable properties. The number of hidden states must be known ahead of time (or chosen using model selection), inference is often costly and subject to local minima when algorithms like Expectation-Maximization are used, and segment lengths are inherently geometrically distributed. Nonparametric Bayesian models like the HDP-HMM [8] relax some of these conditions, but incur a new set of challenges, including the need for MCMC-based inference. In settings where the primary objective is to identify model changes without considering shared hidden states across segments, changepoint detection methods can be a more appropriate algorithmic choice.

Frequentist approaches to changepoint detection and piecewise regression include methods such as PELT [11] that can perform exact inference in linear time over a wide range of cost functions. Alternately, Chopin [5] introduces a Bayesian changepoint detection algorithm that uses a recursive filtering approach, but requires MCMC steps for parameter inference. Building on this work, Fearnhead and Liu present an approximate Bayesian changepoint detection algorithm [7] that can perform online inference efficiently, finding the distribution of locations of the changepoints and the model parameters of each segment using computational time linear in the number of observations. However, this work requires that model parameters can be marginalized. This technique has been used in related work to detect

¹Scott Niekum and Christopher G. Atkeson are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA sniekum@cmu.edu

²Scott Niekum and Andrew G. Barto are with the School of Computer Science, University of Massachusetts Amherst, Amherst, MA 01020, USA

³Sarah Osentoski is with Robert Bosch Research and Technology Center, Palo Alto, CA 94304, USA

changes in the abstraction that underlies a value function [12]. Other approaches to multiple model fitting have been proposed, such as MultiRANSAC [19], but cannot take advantage of the time-series nature of our setting.

B. Articulation Models

Many previous methods have been proposed to identify static articulation relationships between objects from sensor observations. Two of most salient recent examples are the work of Katz and Brock [10] and Sturm et al. [17]. Katz and Brock use interactive perception to actively manipulate objects and visually identify object parts and articulation relationships. Visual features are identified and tracked in the scene to identify rigid bodies and a kinematic graph of connections between parts described by planar kinematic models. Sturm et al. infer kinematic graphs that describe fully 3-D articulation relationships between object parts by analyzing the movement of known objects or object parts. Rather than tracking visual features, they track either visual fiducials or use a markerless framework to detect rectangular objects. In addition to rigid, revolute, and prismatic models, they also allow for the discovery of more complex articulations (like the straight-and-curved track of a garage door opening) using a Gaussian process model. To the best of our knowledge, we present the first work that addresses articulation relationships that can change over time.

C. Learning from Demonstration

Learning from Demonstration (LfD) methods [2], [3] have emerged as a fast, effective way to provide robots with human insight into tasks. To date, the majority of these methods have focused on approaches to policy learning, including trajectory libraries [15], inverse reinforcement learning [1], and dynamic movement primitives [9]. By contrast, other recent work has used LfD to learn information about tasks and objects in the world that is not directly policy-related. Examples of this include incremental learning of object-centric skills and global task structure [14], inference of object affordances from demonstration data [13], and the discovery of articulated relationships between objects [17].

While the above types of information do not encode a policy directly, they provide an understanding of tasks and the world that can be used to generate generalized behavior. Direct policy learning can be highly effective for monolithic actions like a tennis swing [16], but our belief is that complex manipulation tasks are best approached by learning physically interpretable task-related qualities, which can be used to generate appropriate behavior in any given situation.

III. CHANGEPOINT DETECTION USING APPROXIMATE MODEL PARAMETERS

A. Online MAP Changepoint Detection

First, we describe the online MAP (maximum *a posteriori*) changepoint detection model of Fearnhead and Liu [7]. Assume we have time-series observations $\mathbf{y}_{1:n} = (y_1, y_2, \dots, y_n)$ and a set of candidate models Q . Our goal is to infer the MAP set of changepoints times $\tau_1, \tau_2, \dots, \tau_m$,

with $\tau_0 = 0$ and $\tau_{m+1} = n$, giving us $m+1$ segments. Thus, the i^{th} segment consists of observations $\mathbf{y}_{\tau_i+1:\tau_{i+1}}$ and has an associated model $q_i \in Q$ with parameters θ_i .

We assume that data after a changepoint is independent of data prior to that changepoint, and we model the changepoint positions as a Markov chain in which the transition probabilities are defined by the time since the last changepoint:

$$p(\tau_{i+1} = t | \tau_i = s) = g(t - s), \quad (1)$$

where $g(\cdot)$ is a probability distribution over time and $G(\cdot)$ is its cumulative distribution function.

Given a segment from time s to t and a model q , define the model evidence for that segment as:

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

It can be shown how the standard Bayesian filtering recursions and an online Viterbi algorithm can be used to efficiently estimate C_t , the distribution over the position of the first changepoint prior to time t [7]. Define \mathcal{E}_j as the event that given a changepoint at time j , the MAP choice of changepoints has occurred prior to time j and define:

$$P_t(j, q) = p(C_t = j, q, \mathcal{E}_j, \mathbf{y}_{1:t}) \quad (3)$$

$$P_t^{MAP} = p(\text{Changepoint at } t, \mathcal{E}_t, \mathbf{y}_{1:t}). \quad (4)$$

This results in the equations:

$$P_t(j, q) = (1 - G(t - j - 1)) L(j, t, q) p(q) P_j^{MAP} \quad (5)$$

$$P_t^{MAP} = \max_{j, q} \left[\frac{g(t - j)}{1 - G(t - j - 1)} P_t(j, q) \right]. \quad (6)$$

At any point, the Viterbi path can be recovered by finding the (j, q) values that maximize P_t^{MAP} . This process can then be repeated for the values that maximize P_j^{MAP} , until time zero is reached. A straightforward alternate formulation [7] allows for the simulation of the full posterior distribution of changepoint locations, though in this work, we focus only on the MAP changepoints.

The algorithm is fully online, but requires $O(n)$ computation at each time step, since $P_t(j, q)$ values must be calculated for all $j < t$. To reduce computation time to a constant, ideas from particle filtering can be leveraged to keep only a constant number of particles, M , at each time step, each of which represent a support point in the approximate density $p(C_t = j, \mathbf{y}_{1:t})$. At each time step, if the number of particles exceeds M , stratified optimal resampling [7] can be used to choose which particles to keep in a manner that minimizes the Kolmogorov-Smirnov distance from the true distribution in expectation.

B. CHAMP

The model evidence shown in Equation 2 requires that the parameters of the underlying model can be marginalized. This requires the use of either conjugate priors, allowing analytical integration, or a low dimensional parameter space that can be efficiently numerically integrated. However, many

models do not fit into either of these categories, requiring an alternate solution for when only point-estimates of model parameters are available.

We present CHAMP (**C**hangepoint detection using **A**pproximate **M**odel **P**arameters)—a modified version of Fearnhead and Liu’s changepoint algorithm that allows the use of models of any form, in which parameter estimates are available via means such as maximum likelihood fit, MCMC, or sample consensus methods. We propose three primary changes to best accommodate this new setting.

1) *Approximate model evidence*: The Bayesian Information Criterion (BIC) is a well-known approximation to integrated model evidence [4] that provides a principled penalty against more complex models by assuming a Gaussian posterior distribution of parameters around the estimated parameter value $\hat{\theta}$. Using the BIC, the model evidence can be approximated as:

$$\ln L(s, t, q) \approx \ln p(\mathbf{y}_{s+1:t} | q, \hat{\theta}) - \frac{1}{2} k_q \ln(t - s), \quad (7)$$

where k_q is the number of free parameters of model q . This approximation allows us to avoid directly evaluating the model evidence integral.

2) *Minimum segment length*: Since we are now assuming that parameter estimates come from some type of model fitting procedure, the quantity $L(s, t, q)$ is no longer well-defined for all $t > s$. Instead, each model q has a minimum value of $t - s$ for which the model is defined. For example, a line requires a minimum of two points to define, whereas a plane requires three. As a simplification, and to prevent overfitting, some sufficient minimum segment length α can be chosen for all models. This requires three changes: changepoints can only begin to be considered at time $t = 2\alpha$ (when a changepoint in the center would create two equal halves of length α), $P_t(j, q)$ must only be calculated for values of $t - j > \alpha$, and the choice of a segment length distribution $g(\cdot)$ must be reconsidered.

Fearnhead and Liu suggest the use of a geometric length distribution [7], as it arises naturally from a constant probability of seeing a changepoint at each time step. However, it is a monotonically decreasing distribution with a mode of 1 that favors shorter segments, which can lead to overfitting, especially in a setting with fitted model parameters. As an alternative, Chopin [5] suggests using a uniform prior over limited support to ensure it is well-defined. However, this artificially places a hard limit on segment lengths, regardless of the data. We propose the use of a truncated normal distribution, which enforces a minimum segment length naturally, has easily interpretable parameters, and is less prone to overfitting:

$$g(t) = \frac{\frac{1}{\sigma} \phi\left(\frac{t-\mu}{\sigma}\right)}{1 - \Phi\left(\frac{\alpha-\mu}{\sigma}\right)} \quad (8)$$

$$G(t) = \Phi\left(\frac{t-\mu}{\sigma}\right) - \Phi\left(\frac{\alpha-\mu}{\sigma}\right), \quad (9)$$

where ϕ is the standard normal PDF, Φ is its CDF, and α is the minimum segment length. Since the mode of the distribution is close to the mean (or identical if no truncation occurs), segment lengths are pushed toward the mean, instead of being pushed toward 1. By using a broad value of σ , we can support a wide range of segments lengths, while leaving μ as an adjustable parameter that can be tuned if over-segmentation or under-segmentation is an issue. Alternatively, if specific prior knowledge about segment length is known, μ can be set accordingly with a more narrow value of σ to restrict segment length appropriately.

3) *Particle definition*: Finally, since model fitting can be an expensive procedure, we suggest a slight revision of the definition of a particle from that of Fearnhead and Liu. Previously, each particle represented a support point to approximate the joint distribution $p(C_t = j, \mathbf{y}_{1:t})$, marginalizing over models q . To potentially save on the number of required model fits, we suggest each particle also include the model, so that our approximated distribution is $p(C_t = j, q, \mathbf{y}_{1:t})$, allowing particular models to be selectively discarded at each time step. This also prevents us from overlooking the possibility of a changepoint at a given time step when only one model is a reasonable fit and the others are very poor.

Figure 1 provides pseudocode for CHAMP. Additionally, an open-source implementation of CHAMP as a ROS service is available online ¹.

IV. ARTICULATION CHANGEPOINT DETECTION

In order to use CHAMP to detect changes in observed articulated motion between objects, we require an articulation likelihood model and a parameter estimation method for any particular data segment. In this work, we use the articulation models of Sturm et al. [17] and MLESAC [18] for parameter estimation. Define the relative transform between two objects with poses \mathbf{x}_i and $\mathbf{x}_j \in SE(3)$ as $\Delta_{ij} = \mathbf{x}_i \ominus \mathbf{x}_j$ ². Given covariance Σ_y , the observation model is defined as:

$$\mathbf{y} \sim \begin{cases} \Delta + \mathcal{N}(0, \Sigma_y) & \text{if } v = 1 \\ \mathcal{U} & \text{if } v = 0, \end{cases} \quad (10)$$

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

$$p(\mathbf{y} | \Delta) = p(\mathbf{y} | \Delta, \gamma) p(\gamma) \quad (11)$$

with:

$$p(\mathbf{y} | \Delta, \gamma) = (1 - \gamma) p(\mathbf{y} | v = 1) + \gamma p(\mathbf{y} | v = 0) \quad (12)$$

$$p(\gamma) \propto e^{-w\gamma}, \quad (13)$$

where w is a weighting constant. This formulation allows for the overall data likelihood to be only minimally affected by a small number of outliers.

¹<http://wiki.ros.org/changepoint>

²For example, if $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^{4 \times 4}$ are homogeneous matrices, then $\mathbf{x}_i \ominus \mathbf{x}_j = \mathbf{x}_i^{-1} \mathbf{x}_j$

Input: Observations $\mathbf{y}_{1:n}$, candidate models q_1, \dots, q_r , prior distribution $\pi(q)$, minimum segment length α , and maximum number of particles M .

Output: Viterbi path of changepoint times and models

```

// Initialize data structures
1: max_path, prev_queue, particles = {}
2: prev_queue.push(1/r)
3: for i = 1 : r do
4:   new_p = newParticle(pos = 0, model = q_i,
   prev_MAP = 1/r)
5:   particles.add(new_p)
6: end for

// Do for all incoming data, starting at time alpha
7: for t = alpha : n do

// Add new particles
8:   if t >= 2*alpha then
9:     pref = prev_queue.pop() // P_t^{MAP} alpha steps ago
10:    for i = 1 : r do
11:      new_p = newParticle(pos = t-alpha, model = q_i,
      prev_MAP = pref)
12:      particles.add(new_p)
13:    end for
14:  end if

// Compute fit probabilities for all particles
15:  for p in particles do
16:    p.tjq = L(p.pos, t, q) * pi(q) * p.prev_MAP
17:    p.MAP = g(t - p.pos) * p.tjq
18:  end for

// Find max particle and update Viterbi path
19:  max_p = max_p p.MAP
20:  prev_queue.push(max_p.MAP)
21:  max_path.add(j = max_p.pos, q = max_p.model)

// Resample if too many particles
22:  if particles.length > M then
23:    particles = stratOptResample(particles, M)
24:  end if

25: end for

// Recover the Viterbi path
26: v_path = {}
27: curr_cp = n
28: while curr_cp > 0 do
29:   <j, q> = max_path[curr_cp - alpha]
30:   v_path.add(start = j, end = curr_cp, model = q)
31:   curr_cp = j
32: end while
33: return v_path

```

Fig. 1. CHAMP

We then use three candidate articulation models $\mathcal{M}^{\text{rigid}}$, $\mathcal{M}^{\text{prismatic}}$, and $\mathcal{M}^{\text{revolute}}$ to describe the relationship between two objects. Define $p(\mathbf{y}|\mathcal{M}, \theta)$ as the likelihood function of the observed relative transform \mathbf{y} between two objects under model \mathcal{M} with parameters θ . The rigid model has 6 parameters that encode the rigid transform between the two objects. The prismatic model has 8 parameters which define the transform between the base object and the origin of the prismatic joint and its axis. The revolute model has 9 parameters that define the transform between the base object and the center of rotation, and a transform from the center of rotation to the moving point.

The full derivation of the forward and inverse kinematic functions defined by these articulation models is outside the scope of this work; for more details, see Sturm et al. [17]. However, each model defines functions of the form:

$$f_{\mathcal{M}, \theta}^{-1}(\mathbf{y}) = \mathbf{c} \quad (\text{inverse kinematics})$$

$$f_{\mathcal{M}, \theta}(\mathbf{c}) = \mathbf{\Delta} \quad (\text{forward kinematics}),$$

where \mathbf{c} is a configuration (ex. a scalar position on a prismatic axis, or an angle in radians on an axis of rotation). These can be used to project an observation \mathbf{y} onto a model, giving us a predicted relative pose $\hat{\mathbf{\Delta}}$, and allowing us to define a likelihood function, using the observation model introduced in Equation 11:

$$p(\mathbf{y}_{1:n}|\mathcal{M}, \hat{\theta}) = \prod_{t=1}^n p(\mathbf{y}_t|\hat{\mathbf{\Delta}}_t). \quad (14)$$

Finally, we use these models to define our BIC-penalized likelihood function for CHAMP:

$$\ln L(s, t, q) \approx \ln p(\mathbf{y}_{s+1:t}|\mathcal{M}_q, \hat{\theta}) - \frac{1}{2}k_q \ln(t - s), \quad (15)$$

where estimated parameters $\hat{\theta}$ are inferred using MLESAC (Maximum Likelihood Estimation Sample Consensus) [18]. MLESAC is a sample consensus method that generates a guess for the model parameters θ by randomly choosing a minimal set of points that can be used to define the model. After generating many guesses, the guess that maximizes that data likelihood is retained and can be improved iteratively with a nonlinear optimization method, BFGS, and by estimating the outlier ratio γ with Expectation-Maximization.

V. EXPERIMENTS

A. 1-D Gaussian Experiment

First, we present an experiment to demonstrate the ability of CHAMP to reliably detect changepoints using maximum likelihood parameter estimates. Five segments of data were generated (of lengths 40, 60, 30, 50, and 70) by making draws from a zero-mean Gaussian distribution with parameterized variance ($\sigma = 2.0, 1.0, 3.0, 1.5$, and 2.5), shown in the left panel of Figure 2. CHAMP was then used to try to recover the locations of the changepoints with the following parameters: a truncated Gaussian length distribution with $\mu = 50$ and $\sigma = 10$, a minimum segment length of 2, and 100 maximum particles.

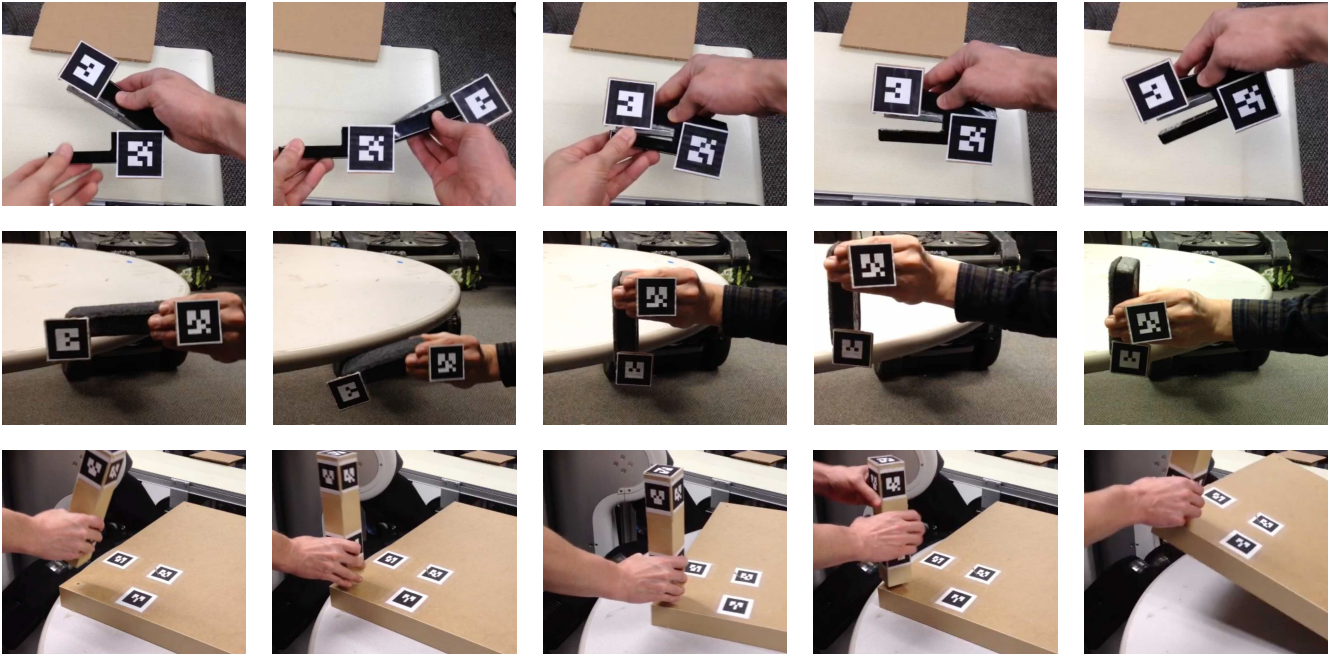


Fig. 3. Demonstrations of stapler manipulation (top), re-grasp of a whiteboard eraser (middle), and partial table assembly (bottom).

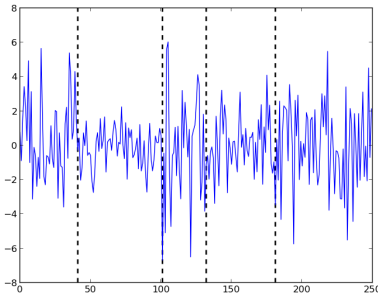


Fig. 2. Five segments of mean-zero Gaussian data with changing variance accurately segmented by CHAMP

Figure 2 shows a segmentation of the data by CHAMP that correctly divides the data into 5 segments. In 100 out of 100 trials, CHAMP discovered the correct number of changepoints and their locations were all found to be within 2 time steps of the true changepoint locations.

B. Articulation Experiments

We now present three scenarios in which CHAMP is used to detect changes in articulated motion between objects during task demonstrations. The timing of these changes and the parameters of each articulated motion are then used to infer physically-grounded information about the task and the associated objects. In each experiment, a human demonstration is observed via an RGB-D sensor mounted on a PR2 mobile manipulator. We have restricted each scene to contain only two relevant objects, or object parts, which are marked with AR-tags, a type of visual fiducial that allows for easy recovery of the 6-DOF object pose. Data is collected

at 10 Hz, recording the relative pose of the two objects $\mathbf{y} \in SE(3)$, and ignoring the data point if either object is not visible.

For all experiments, the following parameters were used for CHAMP: a truncated Gaussian length distribution with $\mu = 100$ and $\sigma = 5$, a minimum segment length of 10, and 10 maximum particles. Additionally, the following parameters we used for the articulation models: observation covariance $diag(\Sigma_y) = \{\sigma_{pos}, \sigma_{orient}\} = \{0.0075 \text{ m}, \pi/12 \text{ rad}\}$, 50 sample consensus iterations, and no parameter optimization, except for the final set of segment parameters, which go through 10 iterations of optimization.

In each experiment, only a single demonstration is given, emphasizing the possibility for one-shot learning when using our system, in contrast to the larger number of demonstrations often required by LfD methods that learn policies directly. In all experiments, segmentation was performed 10 times, always yielding highly similar results as in the Gaussian example. As a proof-of-concept, the changepoint times, models, and model parameters discovered by CHAMP are used to infer physically-interpretable task information in each case that could be used in the future to improve decision making, planning, and policy learning.

1) *Stapler Manipulation*: The first experiment focuses on an object that has multiple intrinsic articulation states that can change over time—a stapler that can act as a revolute joint or lock rigidly when rotated beyond a certain point. The robot is shown a demonstration of a human manipulating the top arm of the stapler, as shown in the top row of Figure 3, first showing it’s ability to rotate, then pushing it into a locked position and demonstrating rigid movement of the whole object.

The left panel of Figure 4 shows the segmentation pro-

duced by CHAMP, in which each data point is a relative pose between the two tags, and points are colored to indicate the segment they belong to. Two segments were found—first a rotational segment, denoted by the red circle drawn to show all possible relative poses allowed by the model, and a rigid segment, indicated by the green sphere with radius equal to $3\sigma_{pos}$. Thus, CHAMP was able to accurately detect the two separate articulation states.

Using this segmentation data, a simple proof-of-concept causal model can be learned to explain the cause of the articulation model change from revolute to rigid. First, all the data prior to the changepoint is projected onto the model using its inverse kinematic equation, providing rotational configurations c . We then examine the configurations that occur in the 2 second window prior to the changepoint, and check if those configurations are outside the range of configurations observed at any other prior time; if they are unique in that sense, then we take that as evidence that those configurations may have caused the model change.

The left column of Figure 5 shows a green cone of configurations (where a configuration angle corresponds to the location of the center of the tag on the top arm of the stapler) that appears to cause the model change. The bottom image shows the stapler resting at a position just short of becoming locked; it can be seen that the center of the tag is just above the beginning of the cone, showing that our system has accurately learned that if the arm were pushed any further, it would likely change to a rigid locked state. This is a simple example, but it illustrates the power of being able to identify changepoints, allowing relevant facts to be learned about objects in the world. It is easy to imagine extending this to mine a larger set of experiences for causal structure and to use additional types of data like force to understand, for example, how much force is required to unlock the stapler and make it rotate again.

2) *Eraser Regrasp*: The next experiment examines the case of frictional contact that can create quasi-static articulated motion—a loose grasp of an object can produce rigid motion with the hand, but can also afford revolute and prismatic motion when external forces are applied. We are motivated by the concept of *extrinsic dexterity* [6], in which in-hand regrasps of objects are accomplished by leveraging forces like gravity and contact with external objects, rather than relying on complex, high-DOF manipulation with the hand. Thus far, such re-grasps have been hand-coded for particular objects [6], but we provide a proof-of-concept that it is possible to learn these re-grasps from demonstration by identifying changes in articulation models.

The robot is shown a demonstration of a re-grasp of a whiteboard eraser, as shown in the center row of Figure 3: it begins with a side-grasp from a table top, and then uses the table to rotate the eraser 90 degrees, followed by a repositioning, and another assist from the table to prismatically slide the grasp to the center of mass so that the grasp is suitable for using the eraser. The center panel of Figure 4 shows the segmentation produced by CHAMP, in which 3 segments are found—a rotation, a rigid segment

(moving into position for the next table contact), and a prismatic shift.

It would be desirable for the robot to be able to reproduce this behavior, possibly in a new environment, after seeing the demonstration, but there is the *correspondence problem*, in which the robot does not know how its body corresponds to that of the demonstrator. For example, from the robot’s point of view, it only saw two tags moving around, and has no concept of hands, fingers, or the grasp locations utilized by the demonstrator. However, using the segmentation data, we can work backwards and infer the human’s initial grasp location, without resorting to computer vision or other more complicated means.

Examining the first rotational segment, we simply identify the perpendicular axis that goes through the center of rotation, giving us the initial grasp location. The center column of Figure 5 shows a green arrow representing this axis, which can serve as an approximately correct grasp location at the edge of the eraser. From this initial location, it would then be trivial to calculate where to apply forces on the eraser to produce the desired motion from the demonstration, and also how the expected grasp location would change, giving us a full map of the re-grasp to follow in the future.

3) *Table Assembly*: The final experiment involves an assembly task, in which each step of the task intentionally changes the articulation relationships between objects. Our goal is to analyze demonstrations to discover the underlying number of steps and the associated articulation changes. These expected changes in articulation relationships can then be used to automatically design kinematic tests that can be used to verify if an assembly step has been successfully accomplished.

The robot is shown an example of a partial table assembly task, as shown in the bottom row Figure 3. First, the table leg and screw are inserted into the main table surface. The demonstrator then moves the leg in the plane, causing the table to move as well, showing the solid connection that has been made. The demonstrator then screws the leg in until it is rigid and again moves the leg, this time showing rigidity out of the plane as well. We preprocess the data, identifying segments of free-body movement if the leg and table are too far apart to be touching. The right panel of Figure 4 shows the segmentation produced by CHAMP, in which 6 segments are found—a free-body segment (the insertion approach), a rigid segment (the movement in the plane), 3 rotational segments during the screwing-in process (as the screw went further in, the center of rotation got lower, and CHAMP correctly detected that as a change in parameters), and a final rigid segment (the movement in and out of the plane).

In sequential tasks that require precision, it is often important to verify that each step has been completed correctly before moving on to the next step. In this case, the rigid data segments can be leveraged by using them as templates for verification tests. Each rigid segment is analyzed to determine in which axis-aligned directions rigid movement between the objects was actually observed, and in which directions no movement was observed, using a threshold to

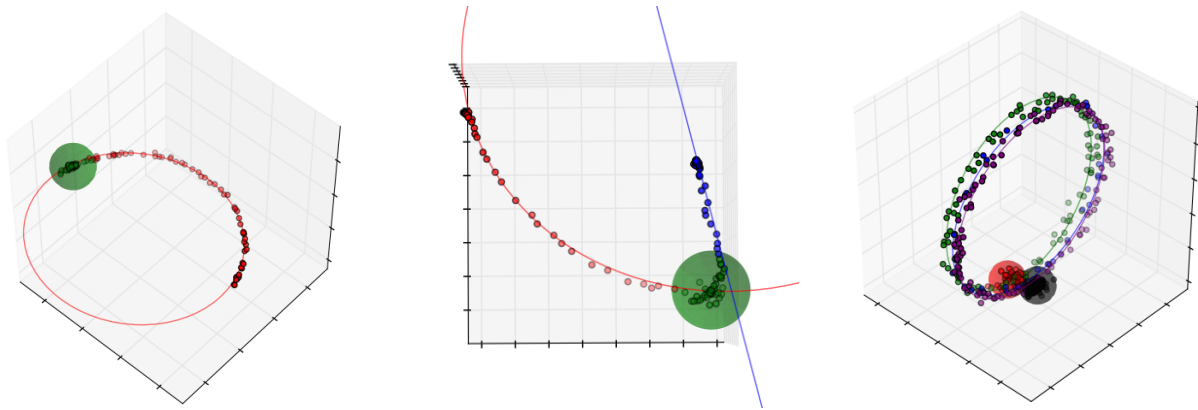


Fig. 4. Segmentation via CHAMP of the stapler data (left), eraser data (center), and table data (right). Data points (relative poses), along with their corresponding articulation, are colored by segment. A colored sphere indicates a rigid model, where the radius corresponds to $3\sigma_{pos}$ in the observation model. A line and circle illustrate a fitted revolute and prismatic model, respectively.

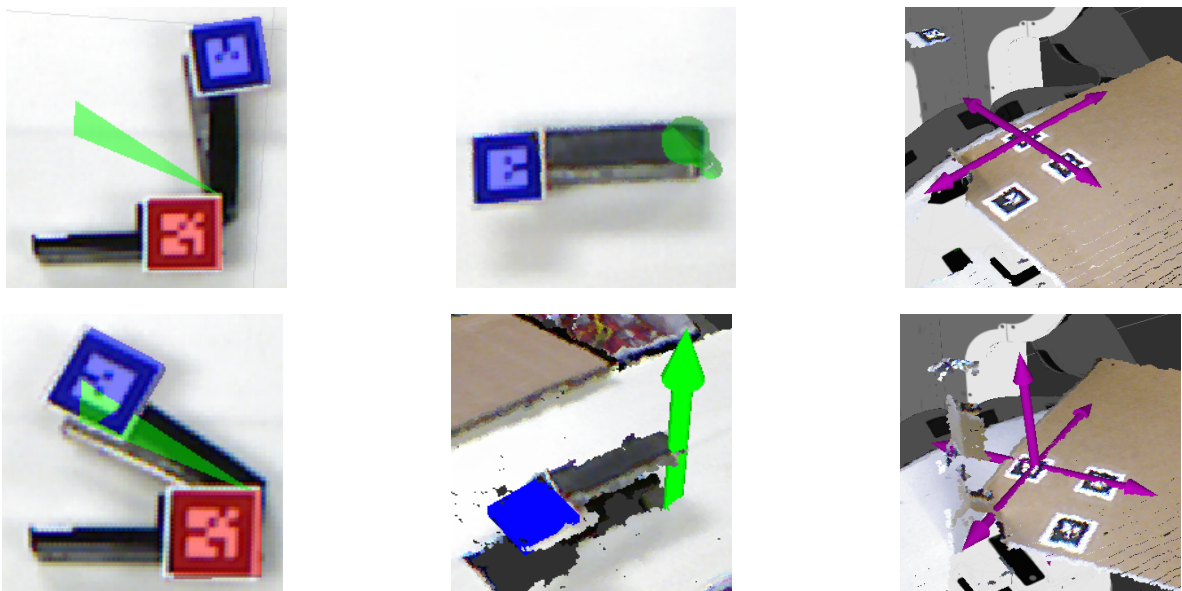


Fig. 5. (Left) The green cone shows the range of configurations that are associated with a change from a revolute to a rigid model. (Center) The green arrow indicates the estimated initial grasp position used by the human demonstrator. (Right) The purple arrows indicate the directions (relative to a fixed table coordinate frame) in which rigidity between the parts has been observed at various times during the task. When the leg/screw is inserted, rigidity in the plane is observed (top), but after the leg is screwed in, there is also rigidity out of the plane (bottom).

account for noise.

The right column of Figure 5 shows purple arrows that have been drawn relative to the table to illustrate the directions rigid motion was observed after the insertion (top image, arrows only in the x-y plane) and after fully screwing in the leg (bottom image, arrows in the x-y plane, and in the z-direction as well). We see that our approach is able to infer that after the insertion, rigidity is expected in the plane of the table, but not out of it (since that would pull out the screw), and that after the screw has been tightened, rigidity is expected in all directions. In the future, such information could be used to automatically design skill verification tests—for example, applying force in each direction that rigidity is expected and checking visually or through force sensing for the appropriate response.

VI. DISCUSSION AND CONCLUSION

Many everyday tasks in robotics depend on understanding the changing physical relationships between objects in the world. We introduced a general-purpose changepoint detection algorithm, CHAMP, that extends Bayesian changepoint detection to settings in which it is difficult or undesirable to integrate out the parameters of candidate models. We then used CHAMP in conjunction with several models of articulated movement to identify changes in articulation relationships between observed objects in the world. This changepoint information was then used to infer physically grounded task-relevant information that could be used to assist in future robotic decision making and policy generation. Specifically, we inferred a simple causal manipulation model, solved a human-robot grasp correspondence problem, and

automatically generated skill verification tests for a task.

More generally, we view this work as one particular example of how robots can learn about the physical world from demonstration data to inform robust decision making, rather than learning potentially brittle policies directly from demonstrations. When using structured models that reflect prior knowledge about the world (in this case, a limited number of basic articulation types), we show that this kind of inference can be performed robustly with very little data.

Our application of CHAMP advances the state-of-the-art in both changepoint detection and articulation identification, but leaves several interesting issues to be addressed in future work. First, we could consider additional articulation models (such as a helical screw), compositions of models, or even use CHAMP on entirely different types of data like force measurements. Second, the MLESAC method of fitting articulation models is entirely passive and can be tricked easily. For example, if two unconnected objects move only along a line with respect to each other, our method will assume that a prismatic joint connects them, despite having no evidence that they are connected rigidly in other directions. To remedy this, future work could include a model of uncertainty over various hypotheses, and additionally could use active learning to allow the robot to actively reduce uncertainty and learn about articulation relationships autonomously.

ACKNOWLEDGMENT

Scott Niekum, Sarah Osentoski, and Andrew G. Barto were funded in part by the National Science Foundation under grant IIS-1208497. This material is also based upon work supported in part by the National Science Foundation (IIS-0964581) and the DARPA Robotics Challenge programs.

REFERENCES

- [1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty First International Conference on Machine Learning*, pages 1–8, 2004.
- [2] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.
- [3] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. *Handbook of Robotics*, chapter Robot programming by demonstration. Springer, Secaucus, NJ, USA, 2008.
- [4] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.
- [5] N. Chopin. Dynamic detection of change points in long time series. *Annals of the Institute of Statistical Mathematics*, 59(2):349–366, 2007.
- [6] N. C. Daffe, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrügge. Extrinsic dexterity: In-hand manipulation with external forces. In *International Conference on Robotics and Automation*, 2014.
- [7] P. Fearnhead and Z. Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007.
- [8] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. An HDP-HMM for systems with state persistence. In *Proceedings of the 25th International Conference on Machine Learning*, pages 312–319. ACM, 2008.
- [9] A. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. *Advances in Neural Information Processing Systems 16*, pages 1547–1554, 2003.
- [10] D. Katz and O. Brock. Extracting planar kinematic models using interactive perception. In *Unifying Perspectives in Computational and Robot Vision*, pages 11–23. 2008.
- [11] R. Killick, P. Fearnhead, and I. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.
- [12] G. Konidaris, S. Kuindersma, R. Grunpen, and A. Barto. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31(3):360–375, 2012.
- [13] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor. Learning object affordances: From sensory-motor coordination to imitation. *Robotics, IEEE Transactions on*, 24(1):15–26, 2008.
- [14] S. Niekum, S. Chitta, B. Marthi, S. Osentoski, and A. G. Barto. Incremental semantically grounded learning from demonstration. In *Robotics: Science and Systems*, 2013.
- [15] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal. Towards associative skill memories. *IEEE-RAS International Conference on Humanoid Robots*, 2012.
- [16] S. Schaal. Dynamic movement primitives—a framework for motor control in humans and humanoid robotics. *The International Symposium on Adaptive Motion of Animals and Machines*, 2003.
- [17] J. Sturm, C. Stachniss, and W. Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41(2):477–526, 2011.
- [18] P. H. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.
- [19] M. Zuliani, C. S. Kenney, and B. Manjunath. The multiRANSAC algorithm and its application to detect planar homographies. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–153. IEEE, 2005.