Learning and Generalization of Complex Tasks from Unstructured Demonstrations

Scott Niekum[1] Sarah Osentoski[2] George Konidaris[3] Andrew G. Barto[1]

***Abstract*— We present a novel method for segmenting demonstrations, recognizing repeated skills, and generalizing complex tasks from unstructured demonstrations. This method combines many of the advantages of recent automatic segmentation methods for learning from demonstration into a single principled, integrated framework. Specifically, we use the Beta Process Autoregressive Hidden Markov Model and Dynamic Movement Primitives to learn and generalize a multi-step task on the PR2 mobile manipulator and to demonstrate the potential of our framework to learn a large library of skills over time.**

I. INTRODUCTION

A simple system that allows end-users to intuitively program robots is a key step in getting robots out of the laboratory and into the real world. Although in many cases it is possible for an expert to successfully program a robot to perform complex tasks, such programming requires a great deal of knowledge, is time-consuming, and is often task-specific. In response to this, much recent work has focused on robot learning from demonstration (LfD) [1], where non-expert users can teach a robot how to perform a task by example. Such demonstrations eliminate the need for knowledge of the robotic system, and in many cases require only a fraction of the time that it would take an expert to design a controller by hand.

Ideally, an LfD system can learn to perform and generalize complex tasks given a minimal number of demonstrations without requiring knowledge about the robot. Much LfD research has focused on the case in which the robot learns a monolithic policy from a demonstration of a simple task with a well-defined beginning and end. This approach often fails for complex tasks that are difficult to model with a single policy. Thus, structured demonstrations are often provided for a sequence of subtasks, or *skills*, that are easier to learn and generalize than the task as a whole, and which may be reusable in other tasks.

However, a number of problems are associated with segmenting tasks by hand and providing individual skill demonstrations. Since the most natural way to demonstrate a task is by performing it continuously from start to finish, dividing a task into component skills is not only time-consuming, but often difficult—an effective segmentation can require knowledge of the robot's kinematic properties, internal representations, and existing skill competencies. Since skills may be repeated within and across tasks, defining skills also requires qualitative judgements about when two segments can be considered a single skill, or in deciding the appropriate level of granularity at which to perform segmentation. Users cannot be expected to manually manage this collection of skills as it grows over time.

For this reason, recent work has aimed at automating the segmentation process. Collectively, this body of work has addressed four key issues that are critical to any system that aims to learn increasingly complex tasks from unstructured demonstrations. (By *unstructured*, we refer to demonstrations that are unsegmented, possibly incomplete, and may come from multiple tasks or skills.) First, the robot must be able to recognize repeated instances of skills and generalize them to new settings. Second, segmentation should be able to be performed without the need for *a priori* knowledge about the number or structure of skills involved in a task. Third, the robot should be able to identify a broad, general class of skills, including object manipulation skills, gestures, and goal-based actions. Fourth, the representation of skill policies should be such that they can be improved through practice.

Although many of these issues have already been addressed individually, no system that we are aware of has jointly addressed them all in a principled manner. Our contribution is a framework that addresses all of these issues by integrating a principled Bayesian nonparametric approach to segmentation with state-of-the-art LfD techniques as a first step towards a natural, scalable system that will be practical for deployment to end users. Segmentation and recognition are achieved using a Beta-Process Autoregressive HMM [2], while Dynamic Movement Primitives [3] are used to address LfD, policy representation, and generalization. We apply our framework to acquire skills from demonstration in simulation, and on the PR2 mobile manipulator.

II. BACKGROUND

A. Bayesian Nonparametric Time Series Analysis

Hidden Markov models (HMMs) are generative Bayesian models that have long been used to make inferences about time series data. An HMM models a Markov process with discrete, unobservable hidden states, or modes[1], which generate observations through mode-specific emission distributions. A transition function describes the probability of

[1]S. Niekum and A.G. Barto are with the Department of Computer Science, University of Massachusetts Amherst, Amherst, MA 01003, USA. {sniekum,barto}@cs.umass.edu

[2]Sarah Osentoski is with the Bosch Research and Technology Center, Palo Alto, CA 94304, USA. Sarah.Osentoski@us.bosch.com

[3]G. Konidaris is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. gdk@csail.mit.edu

[1]We refer to hidden states as *modes*, as to not confuse them with the RL concept of states.

each mode at time $t+1$ given the mode at time t, but observations are limited to being conditionally independent given the generating modes. Given a set of observations, the forward-backward and Viterbi algorithms can be used to efficiently infer parameters for the model and determine the most likely sequence of modes that generated the data. Unfortunately, the number of modes must be specified *a priori* or chosen via model selection, which is prone to overfitting. This severely limits the usefulness of HMM inference when dealing with unstructured data. However, recent work in Bayesian nonparametrics offers a principled way to overcome these limitations.

The Beta Process Autoregressive HMM (BP-AR-HMM) [2] fixes two major problems with the HMM model. First, it uses a beta process prior that leverages an infinite feature-based representation, in which each time series can exhibit a subset of the total number of discovered modes and switch between them in a unique manner. Thus, a potentially infinite library of modes can be constructed in a fully Bayesian way, in which modes are flexibly shared between time series, and an appropriate number of modes is inferred directly from the data, without the need for model selection. Second, the BP-AR-HMM is *autoregressive* and can describe temporal dependencies between continuous observations as a Vector Autoregressive (VAR) process, a special case of a linear dynamical system (LDS). The generative model for the BP-AR-HMM can be summarized as follows [4]:

$$
\begin{aligned}
&B|B_0 \sim \text{BP}(1, B_0) \\
&X_i|B \sim \text{BeP}(B) \\
&\pi_j^{(i)}|\boldsymbol{f_i}, \gamma, \kappa \sim \text{Dir}([\gamma, ..., \gamma+\kappa, \gamma, ...] \otimes \boldsymbol{f_i}) \\
&z_t^{(i)} \sim \pi^{(i)}_{z_{t-1}^{(i)}} \\
&\mathbf{y}_t^{(i)} = \sum_{j=1}^{r} A_{j, z_t^{(i)}} \mathbf{y}_{t-j}^{(i)} + \mathbf{e}_t^{(i)}(z_t^{(i)})
\end{aligned}
$$

First, a draw B from a Beta Process (BP) provides a set of global weights for the potentially infinite number of modes. Then, for each time series, an X_i is drawn from a Bernoulli Process (BeP) parameterized by B. Each X_i can be used to construct a binary vector $\boldsymbol{f_i}$ indicating which of the global features, or modes, are present in the i^{th} time series. Thus, B encourages sharing of features amongst multiple time series, while the X_i leave room for variability. Next, given the features that are present in each time series, for all modes j, the transition probability vector $\pi_j^{(i)}$ is drawn from a Dirichlet distribution with self transition bias κ. A mode $z_t^{(i)}$ is then drawn for each time step t from the transition distribution of the mode at the previous time step. Finally, given the mode at each time step and the *order* of the model, r, the observation is computed as a sum of mode-dependent linear transformations of the previous r observations, plus mode-dependent noise.

B. Dynamic Movement Primitives

Dynamic Movement Primitives (DMPs) [3] provide a framework in which dynamical systems can be described as a set of nonlinear differential equations in which a linear point attractive system or limit cycle oscillator is modulated by a nonlinear function. Stability and convergence are guaranteed by introducing an additional canonical system, governed by linear equations that control a 0 to 1 phase variable that attenuates the influence of the nonlinear function over time. DMPs provide simple mechanisms for LfD, RL policy improvement, and execution, which scale easily in time and space and can support discrete or oscillatory movements [5]. In this paper, we focus on the use of point attractive systems for implementing discrete movements with DMPs.

A discrete movement DMP can be described by the transformation system,

$$\tau \dot{v} = K(g - x) - Dv - K(g - x_0)s + Kf(s) \quad (1)$$

$$\tau \dot{x} = v, \quad (2)$$

and the canonical system,

$$\tau \dot{s} = -\alpha s, \quad (3)$$

for spring constant K, damping constant D, position x, velocity v, goal g, phase s, temporal scaling factor τ, and constant α [6]. The nonlinear function f can be represented as a linear combination of basis functions $\psi_i(s)$, scaled by the phase variable, s: $f(s) = \sum_{i=1}^{N} w_i \psi_i(s) s$. We use the univariate Fourier basis [7] for our function approximator, though others have commonly used normalized radial basis functions [6]. The spring and damping constants can be set to ensure critical damping, but we still must find appropriate weights w_i for the nonlinear function f.

Given a demonstration trajectory $x(t)$, $\dot{x}(t)$, $\ddot{x}(t)$ with duration T, we can use LfD to learn a set of values for these weights [5]. Rearranging equation 1, integrating equation 3 to convert time to phase, and substituting in the demonstration for the appropriate variables, we get:

$$f_{\text{target}}(s) = \frac{-K(g - x(s)) + D\dot{x}(s) + \tau\, \ddot{x}(s)}{g - x_0}. \quad (4)$$

Setting the goal to $g = x(T)$, and choosing τ such that the DMP reaches 95% convergence at time $t = T$, we obtain a simple supervised learning problem to find the weights w_i for the basis functions. We use standard linear regression for this task. This LfD procedure provides us with weights for a baseline controller that can be further improved through practice using RL [5], though we do not do so in this paper.

III. LEARNING FROM UNSTRUCTURED DEMONSTRATIONS

We now introduce a framework which integrates four major capabilities critical for the robust learning of complex tasks from unstructured demonstrations. First, the robot must be able to recognize repeated instances of skills and generalize them to new settings. Given a set of demonstrations for a task, we use the BP-AR-HMM to parse the demonstrations into segments that can be explained by a set of latent skills, represented as VAR processes. The BP-AR-HMM enables these skills to be shared across demonstrations and tasks by employing a feature-based representation in which each skill

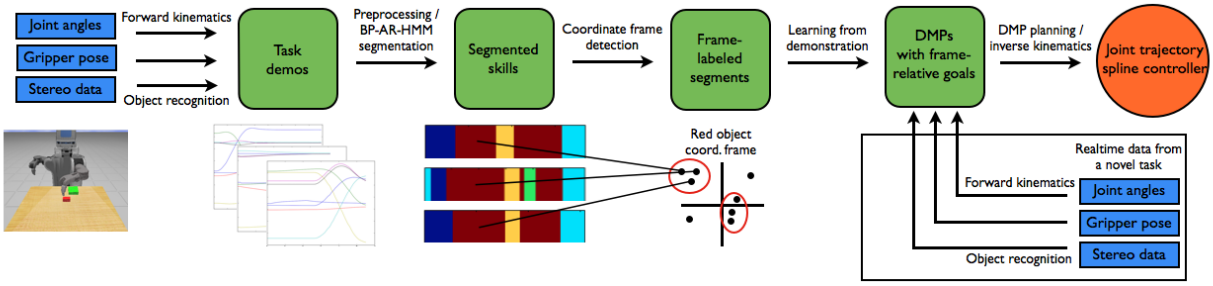


Fig. 1. Overview of the framework used in the experiments, as described in section IV

corresponds to a feature that may or may not be present in a particular trajectory. Furthermore, this representation allows each trajectory to transition between skills in a unique manner, so that skills can be identified flexibly in a variety of situations, while still retaining globally shared properties.

Segmentation of trajectories into VAR models allows for tractable inference over the time-series dependencies of observations and provides a parameterization of each skill so that repeat instances can be recognized. This representation models how state changes over time, based on previous state values, potentially allowing instances of the same underlying skill to be recognized, even when performed with respect to different coordinate frames. The BP-AR-HMM also models skill-dependent noise characteristics to improve the identification of repeated skills. By recognizing repeated skills, a skill library can be incrementally constructed over time to assist in segmenting new demonstrations. Additionally, skill controllers that have been previously learned and improved through practice can be reused on new tasks. Thus, recognition of repeated skills can reduce the amount of demonstration data required to successfully segment and learn complex tasks. Similarly, if we have multiple examples of a skill, we can discover invariants that allow us to generalize the skill to new situations robustly. In this paper, we use this data to identify the coordinate frames that each skill takes place in, as described in detail in the next section.

Second, segmentation must be able to be performed without the need for *a priori* knowledge about the number or structure of skills involved in a task. The BP-AR-HMM places a beta process prior over the matrix of trajectory-feature assignments, so that a potentially infinite number of skills can be represented; the actual finite number of represented skills is decided upon in a principled, fully Bayesian way. Skill durations are modeled indirectly through a learned self-transition bias, preventing skills from being over-segmented into many small components unnecessarily. The BP-AR-HMM also provides reliable inference, having only a few free parameters that are robust to a wide range of initial settings and hyperparameters that conform to the data as inference progresses. Thus, little tuning should be necessary for varying tasks for a given robotic platform.

Third, our system must be able to identify a broad, general class of skills. Since our segmentation method is based upon state changes, rather than absolute state values, we are able to identify a wide array of movement types ranging from object manipulation skills to gestures and goal-based actions. Furthermore, by identifying the relevant coordinate frame of repeated skills, we can discover specific objects and goals in the world that skills are associated with.

Fourth, the representation of skill policies should be such that they can be easily generalized and improved through practice. To accomplish this, we represent skill controllers in the DMP framework. The spring-damper mechanics of a DMP allow for easy generalization, since the start and goal set-points can be moved, while still guaranteeing convergence and maintaining the "spirit" of the demonstration through the output of the nonlinear function.

IV. METHODOLOGY

A. Demonstrations

For the first two experiments in this paper, we use a simulated Willow Garage PR2 mobile manipulator and the ROS framework; the final experiment uses a real PR2. We used hand-coded controllers to provide task demonstrations to the simulated robot. The robot is placed in a fixed position in front of a table, as shown in Figure 2. At the beginning of each demonstration, the robot looks downward and captures a stereo image of the table. It then removes the flat table top and obtains a point cloud for each of the objects on the table, recording their positions and dimensions. On the real robot, object positions are determined by a visual fiducial placed on each object of interest. Once the demonstration begins, data are collected by recording the 7 joint angles in the left arm and the gripper state (a scalar indicating its degree of closedness). Offline, the joint angles are converted to a series of 3D Cartesian positions and 4D quaternion orientations, which are subsampled down to 10 Hz and smoothed, along with the gripper positions.

B. Segmentation

We build on a BP-AR-HMM implementation made available by Emily Fox[2] to segment sets of demonstration trajectories. We preprocess the demonstrations so that the variance of the first differences of each dimension of the data is 1, as in Fox et al. [4], and adjust it to be mean zero. We choose an autoregressive order of 1 and use identical

[2] http://stat.wharton.upenn.edu/~ebfox/software

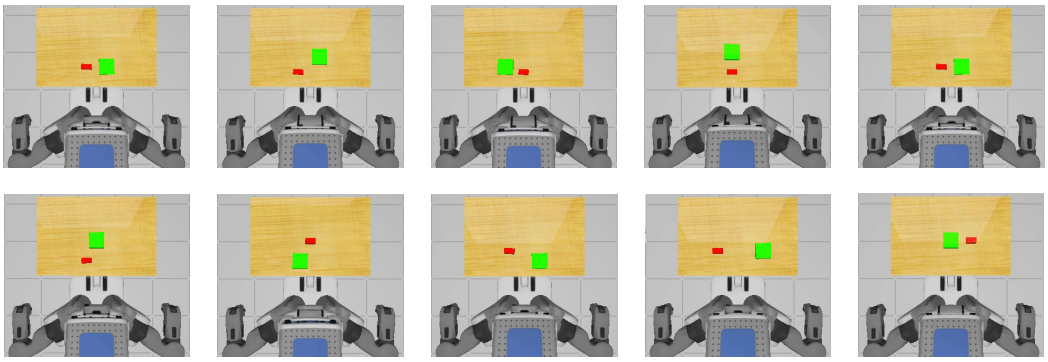


Fig. 2. 5 task demonstration configurations (top) and 5 novel test configurations (bottom).

parameters as those used by Fox on a human exercise motion capture dataset [4], with one exception—in the simulated experiments, we adjust the matrix-normal inverse-Wishart prior on the dynamic parameters, since the simulated data has significantly different statistical properties from that in Fox et al. [4]. To segment the demonstrations, we run the combined Metropolis-Hastings and Gibbs sampler 10 times for 1000 iterations each, producing 10 segmentations. Qualitatively, the segmentations across runs were very consistent, but to ensure good results, the segmentation from the 10 runs with the highest log likelihood of the feature settings is selected.

C. Coordinate Frame Detection

After the demonstrations are segmented, each segment is examined to infer the coordinate frame that it is occurring in. Even though segments assigned to the same skill correspond to similar movements, they may be happening in different frames of reference. For example, a repeated reaching motion may be classified as being generated by the same skill, but be reaching toward several different objects. In order to robustly replay tasks in novel configurations, it is desirable to determine which coordinate frame each segment is associated with, so that DMP goals can be generalized correctly.

We define a coordinate frame centered on each known object, along with one centered at the torso of the robot. Other frames could be used as well if desired, such as a frame relative to the gripper, or a world frame. Then, the final point of each segment is plotted separately in each of the coordinate frames, and clusters are found in each frame by identifying points within a Euclidean distance threshold of each other. The reasoning is that clusters of points indicate that multiple segments have similar endpoints in a particular coordinate frame, suggesting that the skill often occurs in that frame of reference.

After the points are clustered in each frame, all the singleton clusters are discarded. If any remaining segment endpoint belongs only to a cluster in a single coordinate frame, then the evidence is unambiguous, and that segment is assigned to that coordinate frame. Otherwise, if a segment endpoint belongs to clusters in multiple frames, it is simply assigned to the frame corresponding to the largest cluster. It should be emphasized that the any coordinate frame inference method could be used in place of ours, and that there are many other skill invariants that could be exploited. The purpose of this method is primarily to demonstrate the utility of being able to segment and recognize repeated skills.

D. Task Replay

To perform a task in a novel configuration, we first determine the poses and identities of objects in the scene, using either stereo data (simulated experiment) or visual fiducials (real robot). The position of each object is then examined to find the demonstration that begins with the objects in a configuration that is closest to the current one in a Euclidean sense. We only consider demonstrations that have an identified coordinate frame for every segment, so that the task will generalize properly. A DMP is then created and trained using the LfD algorithm from section II-B for each segment in the demonstration. However, rather than using the final point of a segment as the goal of a DMP, each goal is adjusted based on the coordinate frame that the segment takes place in. If the segment is associated with the torso frame, it requires no adjustment. Otherwise, if it is associated with an object frame, the goal is adjusted by the difference between the object's current position and its position in the demonstration. Finally, the DMPs are executed in the sequence specified by the demonstration. A plan is generated by each of the DMPs until the predicted state is within a small threshold of the goal. Each plan is a Cartesian trajectory (plus a synchronized gripper state) that is converted into smooth joint commands using inverse kinematics and spline interpolation. A graphical overview of our method is shown in Figure 1.

V. EXPERIMENTS

A. Experiment 1: Pick and Place (Simulated)

The first experiment demonstrates the ability of our framework to learn and generalize a complex task by segmenting multiple task demonstrations, identifying repeated skills, and discovering appropriate segment reference frames. Each instance of the task begins with two blocks on the table—a smaller red block and a larger green block. The robot always starts in a "home" configuration, with its arms at its sides so that its field of view is unobstructed. We provide 5 task demonstrations for 5 different configurations of the blocks,

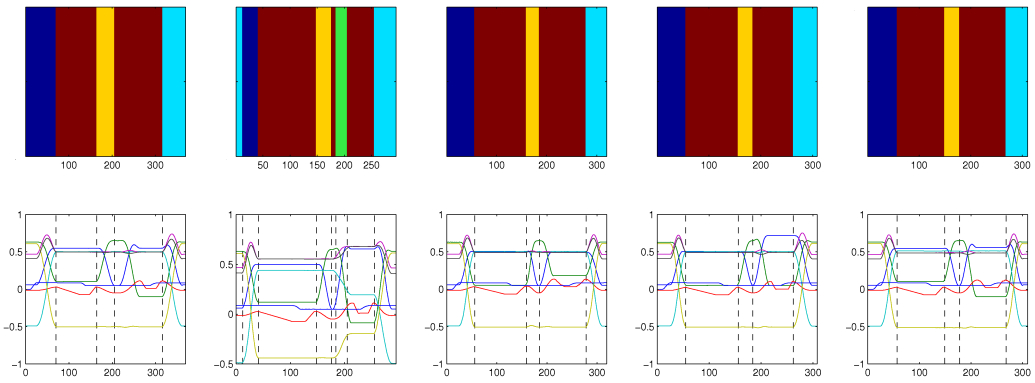
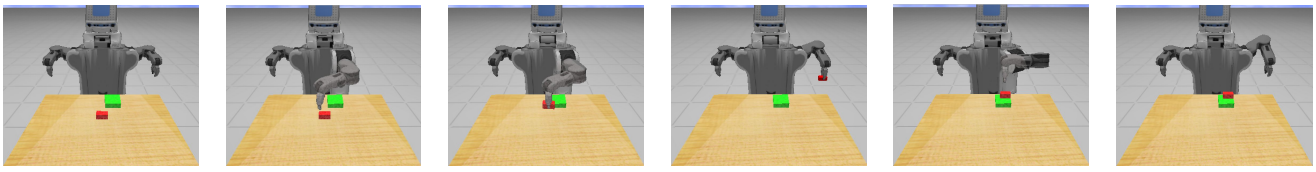


Fig. 3. Top: BP-AR-HMM segmentations of the 5 demonstration trajectories for the pick and place task. Time (in tenths of a second) is shown on the x-axis. Skill labels at each time step are indicated by unique colors. Bottom: Segmentation points overlaid on the demonstrated 8D movement data.



(a) Starting pose (b) Reaches toward red block (red block frame) (c) Picks up red block (red block frame) (d) Returns to home position (torso frame) (e) Places red block on green block (green block frame) (f) Returns to home position (torso frame)

Fig. 4. Successful task replay on a novel test configuration for the pick and place task, demonstrating generalization. From left to right: the starting pose and the final point of each executed DMP. Automatically detected coordinate frames used for each segment are listed in parentheses.

as shown in the first row of Figure 2 (configurations 1 and 5 are identical, but the demonstration is performed at a higher speed in the latter configuration). In each demonstration, the robot first picks up the red block, returns to the home position, places the red block on the green block, and returns to the home position once more.[3]

Figure 3 shows the results of segmentation. The top row shows one colored bar per skill, while the bottom row displays the skill divisions overlaid on a plot of each of the 8 dimensions of the demonstration data. The BP-AR-HMM consistently recognizes repeated skills across demonstrations, even though they occur at differing speeds and with different goals. The segmentations are highly similar, with the exception of the second demonstration, which identifies one additional skill that the others do not have. It is worth noting that despite the extra skill being inserted in the segmentation, the rest of the segmentation is essentially the same as the others. This is a direct benefit of the BP-AR-HMM allowing each trajectory to have its own switching dynamics, while sharing global features.

Next, we examine task generalization to 5 novel test configurations, shown in the bottom row of Figure 2, to determine whether our segmentation produced semantically meaningful results. Our method was able to successfully identify a coordinate frame for every segment except the extra segment in demonstration two (which is impossible to infer, since there is only one example of it). Using this information, the robot performed task replay as described in section IV-D. In all 5 novel configurations, the robot was able to successfully generalize and place the red block on the green block.[4]

Figure 4 shows the starting state of the robot and the resulting state after each DMP is executed in a novel test configuration. Here, it becomes clear that the results of both the segmentation and coordinate frame detection are semantically intelligible. The first skill is a reaching skill to right above the red block. The second skill moves down, grasps the red block, and moves back upward. The third skill goes back to the home position. The fourth skill reaches toward the green block, moves downward, releases the red block and moves back upward. Finally, the fifth skill goes back to the home position. Notice that the second and fourth segments are identified by the BP-AR-HMM as being the same skill, despite having different relevant coordinate frames. However, in both skills, the arm moves down toward an object, changes the gripper pose, and moves back upward; the reach from the home position toward the green block gets rolled into this skill, rather than getting its own, seemingly because it is a smoother, more integrated motion than the reach and grasp associated with the red block.

Given the commonality of pick and place tasks in robotics,

[3]Due to the planning delay in the hand written controllers there are some pauses between segments which we remove to avoid giving the segmentation algorithm an unfair advantage.

[4]The green block in novel configuration 4 was partially out of the robot's visual range, causing part of it to be cut off. Thus, it placed the red block too close to the edge of the green block, causing it to tumble off. However, given the available information, it acted correctly.

success in this domain may seem trivial. However, it is important to keep in mind that the robot is given only demonstrations in joint space and absolutely no other *a priori* knowledge about the nature of the task. It does not know that it is being shown a pick and place task (or doing grasping at all). It is unaware of the number of subtasks that comprise the task and whether the subtasks will be object-related, gestural, or have other sorts of objectives. Beginning with only motion data and a simple assumption about the types of coordinate frames that are relevant to inspect, the robot is able to automatically segment and generalize a task with multiple parts, each having its own relevant coordinate frame.

B. Experiment 2: Using a Skill Library (Simulated)

The first experiment demonstrated that our method can learn and generalize a complex task when given a sufficient number of demonstrations. However, this type of learning will not scale up to more complex tasks easily unless the robot can incrementally build a library of skills over time that allows it to quickly recognize previously seen skill / coordinate frame combinations and reuse complex skill controllers that have been improved through practice. To demonstrate our system's capability to recognize skills in this manner, we simulate a previously existing library of skills by providing the robot with a pre-segmented demonstration of the previous experiment. We then give it a single demonstration of the task to see if it can segment it using the "library" of skills.

The BP-AR-HMM correctly recognized each of the skills in the task as being a skill from the pre-existing library. Thus, assuming the robot already had learned about these skills from previous experiences, it would allow a user to provide only a single demonstration of this task and have the robot correctly segment and generalize the task to new configurations. This serves as a proof-of-concept that our proposed framework has the right basic properties to serve as a building block for future models that will scale up LfD to more complex tasks than have previously been possible. It also emphasizes that our method can learn tasks from unstructured demonstrations, as the majority of demonstrations were not even of the task in question, but of a sub-component, unbeknownst to the robot.

C. Experiment 3: The Whiteboard Survey (Physical PR2)

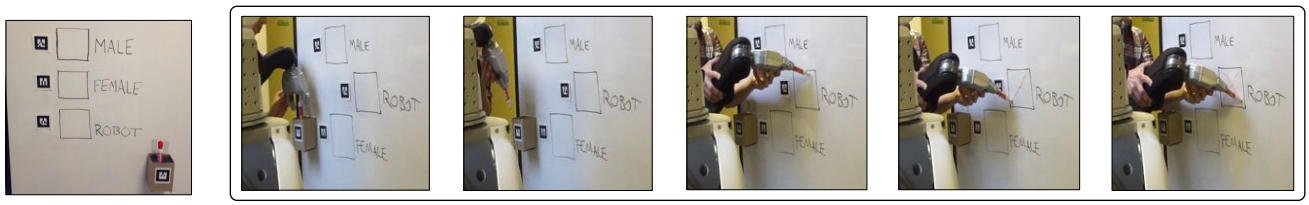
Finally, we demonstrate that our method is scalable to a real robot system, using a physical PR2. Figure 5(a) shows one configuration of a task in which the PR2 must fill out a survey on a whiteboard by picking up a red marker and drawing an 'X' in the checkbox corresponding to "robot" while ignoring the checkboxes for "male" and "female". Each checkbox has its own unique fiducial placed one inch to the left of it, while the container that holds the marker has a fiducial directly on its front. The positions of the checkboxes and the marker container on the whiteboard, as well as the position of the whiteboard itself, change between task configurations. Two kinesthetic demonstrations in each of three task configurations were provided, along with one additional demonstration in which the marker is picked up and then lifted above the robot's head. An example demonstration is shown in Figure 5(b).

Figure 6 shows that the BP-AR-HMM generally parses the demonstrations into three main segments, corresponding to reaching for the marker, grasping and lifting the marker, and drawing an 'X' in the checkbox. However, the reaching and drawing segments are considered to be the same skill. This appears to happen because both motions are statistically similar, not in terms of absolute position, but in the way that the positions evolve over time as a VAR system. Our coordinate frame detection successfully disambiguates these skills and splits them into two separate skill/coordinate frame combinations. Demonstrations 1, 2, and 5 contain a small additional skill near the beginning that corresponds to a significant twitch in the shoulder joint before any other movement starts, which appears to correspond to the teacher's first contact with the arm, prior to the demonstration. Finally, although the last demonstration is of a different task, the reaching and grasping/lifting skills are still successfully recognized, while the final motion of lifting the marker over the robot's head is given a unique skill of its own. Despite having only a single example of the over-head skill, the BP-AR-HMM robustly identified it as being unique in 50 out of 50 trial segmentations, while also recognizing other skills from the main task. After the learning phase, the robot was able to successfully replay the task in three novel configurations, an example of which is shown in Figure 7.

VI. RELATED WORK

A variety of approaches have been proposed for LfD, including supervised learning [8], [9], [10], [11], reinforcement learning [12], [13], [14], and behavior based approaches [15]. However, this work has generally been limited to single tasks with a well-defined beginning and end. In a recent example, Pastor et al. [16] use DMPs to acquire single motor skills from structured demonstrations of a complex billiards shot. In their framework, multiple imperfect demonstrations of a skill are used to learn an initial DMP controller, which is then improved using RL.

While many approaches enable the learning of a single policy from data, some approaches perform automatic segmentation of the demonstrations into skills. Jenkins and Matarić introduced Spatio-Temporal Isomap in order to find the underlying low-dimensional manifolds within a set of demonstrated data [17], [18]. This work extends the dimensionality reduction technique Isomap to include temporal information and allows the discovery of repeated motion primitives. However, segmentation is performed with a heuristic and the motion primitives cannot be improved through techniques like RL. Dixon and Khosla [19] demonstrate that generalizable motions can be parameterized as linear dynamical systems. This algorithm also uses heuristic segmentation and cannot recognize repeated instances of skills. Gienger et al. [20] segment skills based on co-movement between the demonstrator's hand and objects in the world and automatically find appropriate task-space



(a) Task example (b) A kinesthetic demonstration

Fig. 5. The whiteboard survey task.

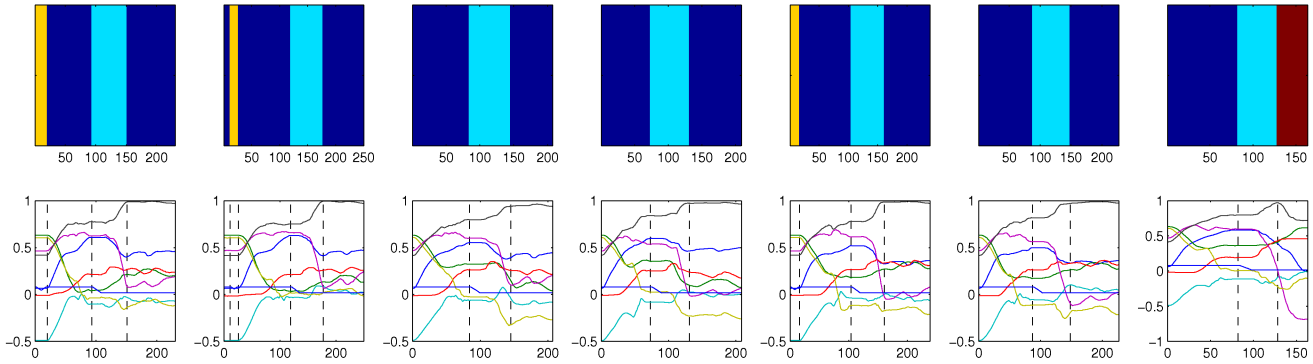


Fig. 6. Top: BP-AR-HMM segmentations of the 7 demonstration trajectories for the whiteboard survey task. Time (in tenths of a second) is shown on the x-axis. Skill labels at each time step are indicated by unique colors. Bottom: Segmentation points overlaid on the demonstrated 8D movement data.

abstractions for each skill. Their method can generalize skills by identifying task frames of reference, but cannot describe skills like gestures or actions in which the relevant object does not move with the hand.

More recent work has examined using principled statistical techniques to segment example trajectories into multiple skills. Grollman and Jenkins [21] introduce the Realtime Overlapping Gaussian Expert Regression (ROGER) model to estimate the number of subtasks and their policies in a way that avoids perceptual aliasing, in which perceptual information alone is not sufficient to choose the correct next action. Butterfield et al. [22] extend Hierarchical Dirichlet Processes Hidden Markov Models (HDP-HMM) to handle perceptual aliasing and automatically discover an appropriate number of skills. Although we use a Bayesian mechanism to parse demonstration trajectories, rather than inferring policies, we discover repeated dynamical systems which are considerably simpler to model than policies.

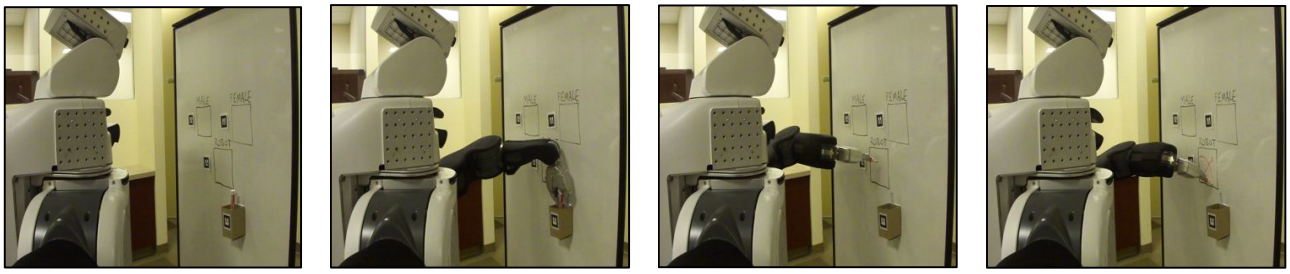
CST [23] uses an online changepoint detection method to segment example trajectories and then merges the resulting chains of skills into a skill tree. This approach simultaneously segments the trajectories and discovers abstractions, but cannot recognize repeated skills to assist with the segmentation process. Kulic et al. [24] demonstrate an online method that can recognize repeated motion primitives to improve segmentation as additional data is collected, by assume that data points from the same primitive are generated by the same underlying distribution. Ciappa and Peters [25] model repeated skills as being generated by one of a set of possible hidden trajectories, which is rescaled and noisy. To guide segmentation, they define an upper bound on the number of possible skills and explicitly constrain segment lengths.

VII. DISCUSSION

We presented a novel method for segmenting demonstrations, recognizing repeated skills, and generalizing complex tasks from unstructured demonstrations. Though previous research has addressed many of these issues individually, our method aims to address them all in a single integrated and principled framework. By using the BP-AR-HMM and DMPs, we are able to experimentally learn and generalize a multiple step task on the PR2 mobile manipulator and to demonstrate the potential of our framework to learn a large library of skills over time.

Our framework demonstrates several of the critical components of an LfD system that can incrementally expand a robot's competency and scale to more complex tasks over time. However, this work is only a first step toward such a system, and leaves a great number of directions open for future research. A more nuanced method must be developed for managing the growing library of skills over time, so that inference in our model does not become prohibitively expensive as the size of the library grows. While our model allows for DMP policy improvement through RL, we did not address such improvement experimentally in this paper. Future work may use techniques such as inverse RL [13] to derive an appropriate reward function for each skill so that policy improvement can be effectively applied.

There are also many more opportunities to take advantage of abstractions and invariants in the data; searching for skill coordinate frames is a very simple example of a much richer



(a) Starting pose (b) Reaches for marker (marker frame) (c) Grasps marker and lifts toward checkbox (robot checkbox frame) (d) Draws 'X' (robot checkbox frame)

Fig. 7. Successful task replay on a novel test configuration for the whiteboard survey task, demonstrating generalization. From left to right: the starting pose and the final point of each executed DMP. Automatically detected coordinate frames used for each segment are listed in parentheses.

class of generalization techniques. It is also desirable to take a more principled approach to coordinate frame detection by integrating it directly into the Bayesian segmentation model, so that reference frames are inferred jointly along with the skills. Finally, more intelligent methods can be applied to make better use of the demonstration data that we have available. In this work, DMPs are constructed from single segments that came from the task configuration most similar to the current one that the robot faces. However, there exist more sophisticated methods involving dynamic time warping [20] and Bayesian techniques [26] to perform LfD with many demonstration segments. Using such techniques, it may be possible to create more robust skill models that can be used in an ever-increasing number of complex situations, allowing end-users to program robots with ease.

REFERENCES

[1] B. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "Sharing features among dynamical systems with beta processes," *Advances in Neural Information Processing Systems 22*, pp. 549–557, 2009.

[3] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," *Advances in Neural Information Processing Systems 16*, pp. 1547–1554, 2003.

[4] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "Joint modeling of multiple related time series via the beta process," *arXiv:1111.4226*, November 2011.

[5] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," *International Symposium on Robotics Research*, pp. 561–572, 2004.

[6] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 763–768.

[7] G. Konidaris, S. Osentoski, and P. Thomas, "Value function approximation in reinforcement learning using the Fourier basis," in *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence*, 2011.

[8] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997, pp. 12–20.

[9] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the Second Conference on Human-Robot Interaction*, 2007.

[10] S. Chernova and M. Veloso, "Confidence-based policy learning from demonstration using gaussian mixture models," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, May 2007.

[11] D. H. Grollman and O. C. Jenkins, "Sparse incremental learning for interactive robot control policy estimation," in *Proceedings of the International Conference on Robotics and Automation*, 2008.

[12] W. D. Smart and L. P. Kaelbling, "Effective reinforcement learning for mobile robots," in *2002 IEEE International Conference on Robotics and Automation*, 2002, pp. 3404–3410.

[13] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the Twenty First International Conference on Machine Learning*, 2004, pp. 1–8.

[14] B. D. Ziebart, A. Maas, J. D. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, July 2008.

[15] M. Nicolescu and M. J. Matarić, "Natural methods for robot task learning: Instructive demonstration, generalization and practice," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003, pp. 241–248.

[16] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, "Skill learning and task outcome prediction for manipulation," in *Proceedings of the 2011 IEEE International Conference on Robotics & Automation*, 2011.

[17] O. C. Jenkins and M. Matarić, "A spatio-temporal extension to Isomap nonlinear dimension reduction," in *Proceedings of the Twenty-First International Conference on Machine Learning*, Jul 2004, pp. 441–448.

[18] O. C. Jenkins and M. J. Matarić, "Performance-derived behavior vocabularies: Data-driven acquisition of skills from motion," *International Journal of Humanoid Robotics*, vol. 1, no. 2, pp. 237–288, Jun 2004.

[19] K. Dixon and P. Khosla, "Trajectory representation using sequenced linear dynamical systems," in *IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 2004, pp. 3925–3930.

[20] M. Gienger, M. Muhlig, and J. Steil, "Imitating object movement skills with robots: A task-level approach exploiting generalization and invariance," in *International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 1262–1269.

[21] D. Grollman and O. Jenkins, "Incremental learning of subtasks from unsegmented demonstration," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 261–266.

[22] J. Butterfield, S. Osentoski, G. Jay, and O. Jenkins, "Learning from demonstration using a multi-valued function regressor for time-series data," in *Proceedings of the Tenth IEEE-RAS International Conference on Humanoid Robots*, 2010.

[23] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 360–375, 2012.

[24] D. Kulic, W. Takano, and Y. Nakamura, "Online segmentation and clustering from continuous observation of whole body motions," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1158–1166, 2009.

[25] S. Chiappa and J. Peters, "Movement extraction by detecting dynamics switches and repetitions," *Advances in Neural Information Processing Systems*, vol. 23, pp. 388–396, 2010.

[26] A. Coates, P. Abbeel, and A. Ng, "Learning for control from multiple demonstrations," in *Proceedings of the 25th International Conference on Machine Learning*. ACM, 2008, pp. 144–151.