

# Probabilistic Segmentation Applied to an Assembly Task

Rudolf Lioutikov<sup>1</sup>, Gerhard Neumann<sup>1</sup>, Guilherme Maeda<sup>1</sup>, Jan Peters<sup>1,2</sup>

**Abstract**—Movement primitives are a well established approach for encoding and executing robot movements. While the primitives themselves have been extensively researched, the concept of movement primitive libraries has not received as much attention. Libraries of movement primitives represent the skill set of an agent and can be queried and sequenced in order to solve specific tasks. The goal of this work is to segment unlabeled demonstrations into an optimal set of skills. Our novel approach segments the demonstrations while learning a probabilistic representation of movement primitives. The method differs from current approaches by taking advantage of the often neglected, mutual dependencies between the segments contained in the demonstrations and the primitives to be encoded. Therefore, improving the combined quality of both segmentation and skill learning. Furthermore, our method allows incorporating domain specific insights using heuristics, which are subsequently evaluated and assessed through probabilistic inference methods. We demonstrate our method on a real robot application, where the robot segments demonstrations of a chair assembly task into a skill library. The library is subsequently used to assemble the chair in an order not present in the demonstrations.

## I. INTRODUCTION

A key goal of modern robotics is to provide robots with the ability to learn new tasks. A commonly followed concept to achieve such behavior is imitation-learning. The robot is provided with one or more demonstrations of a task, which the robot subsequently applies and improves. Usually an entire task consists of a single motion, encoded as a movement primitive. This concept has been applied in a variety of tasks, including hitting movements in table tennis [1] and locomotion [2].

Solving more complex, non-monolithic tasks with a single movement primitive implies, however, a great loss of generality. Considering complex tasks as a sequence of primitives offers multiple advantages. For example, primitives can be easily generalized between the points where they connect. Also, the same set of primitives can be reused to execute different tasks, and the movement plan can be adapted by replacing one primitive within the sequence by a different one. The fundamental problems of such approaches are related to the autonomous acquisition of these primitives without relying on hand labeled demonstrations, and to the subsequent sequencing of the learned skills. In this paper we address the former problem by proposing a framework for segmenting unlabeled demonstrations into a library of movement primitives.

<sup>1</sup>Intelligent Autonomous Systems, TU Darmstadt {lioutikov, neumann, maeda, peters}@ias.tu-darmstadt.de

<sup>2</sup>Max Planck Institute Tuebingen

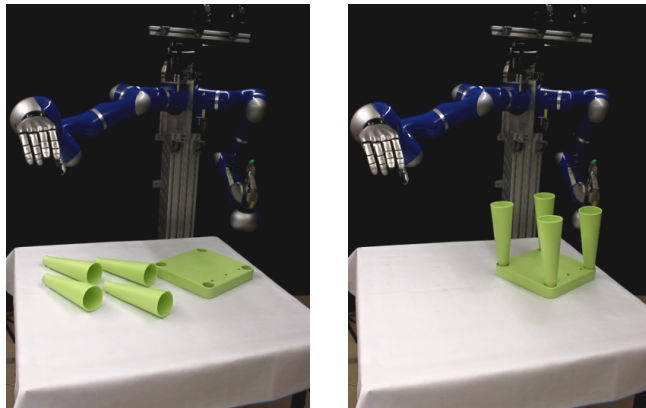


Fig. 1: The robot platform used for a chair assembly experiment. We used a seven DoF KUKA lightweight arm equipped with a five finger DLR HIT Hand II as end effector. The executed movement primitives were learned by segmenting human demonstrations.

Essentially, there are two problems we tackle in this work which will be solved in conjunction. First, the segmentation of demonstrated trajectories. Second, the learning of a movement primitive library given these demonstrations. Each demonstrated trajectory can be considered a multidimensional time series. A common way to segment time series data is to apply heuristics. However, the quality of such heuristic and therefore, the corresponding segmentation is highly task depended. For instance, while an assembly task consisting of point to point motions might be well segmented at zero crossing velocities, the same heuristic applied on continuously written words might achieve poor, meaningless results. Furthermore, different parts of the data could be best explained by different heuristics, which raises the problem of identifying at what point to apply which heuristic.

Our approach starts from the premise that a task-specific heuristic can only segment a given trajectory sub-optimally, therefore, leading to a low-quality library. As a consequence, some skills may not be meaningful while others will suitably describe the data. Therefore, our method applies probabilistic inference to reason iteratively over all possible segmentations, by learning a probabilistic representation of movement primitives from a weighted set of segments. In return, the learned primitives are used to improve the set of segments by down-weighting segments that are less plausible given the current skill library. We provide the mathematical formulation for the solution of this problem as an iterative Expectation-Maximization (EM) algorithm and show that our algorithm converges to a compact set of skills given over-segmented

demonstrations.

In summary, the main contribution of this work is the Probabilistic Segmentation (ProbS) algorithm that concurrently improves a given segmentation and the library of skills. Additionally, the method is validated on a real robot platform, by segmenting the human motion of a chair assembly and subsequently sequencing skills from the learned library to assemble the chair with the robot.

### A. Problem Statement and Notation

Given a set of observed trajectories  $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_{|\mathcal{T}|}\}$ , the goal of this work is to learn a set of underlying skills  $\mathcal{M} = \{m_1, m_2, \dots, m_{|\mathcal{M}|}\}$  which explains  $\mathcal{T}$ , i.e., the movement primitive library which produced  $\mathcal{T}$ . Since we are interested in the underlying library  $\mathcal{M}$  of a task domain, the trajectories in  $\mathcal{T}$  can describe the same or multiple tasks, as long as they belong to the same domain, and, therefore, can be explained by the same library. The duration of each individual demonstration  $\tau$  might be different. An example of a one dimensional trajectory is illustrated in Figure 2. For each trajectory  $\tau \in \mathcal{T}$ , a set of possible cutting points is defined  $\mathcal{C}^{[\tau]}$ . Each cutting point  $c_i$  is a time step at which one segment ends  $s_{h,i}$  and another one starts  $s_{i,j}$ , with  $c_h, c_i, c_j \in \mathcal{C}^{[\tau]}$ . We assume that  $\mathcal{C}^{[\tau]}$  over-segments  $\tau$ , i.e, there are more cutting points in  $\mathcal{C}^{[\tau]}$  than transitions between the underlying segments. Unfortunately, it is unknown which of the cuts are true positives and which are false positives. Therefore, each possible subset  $d^{[\tau]} \subseteq \mathcal{C}^{[\tau]}$  has to be considered as a possible segmentation. The set of all possible segmentations will be denoted as  $\mathcal{D}^{[\tau]}$ . Furthermore, the set of segments  $\mathcal{S}^{[\tau]}$  does not only contain segments between two consecutive cuts, but all segments defined by each possible combination of two cuts in  $\mathcal{C}^{[\tau]}$ . The presented method tackles two challenges simultaneously: Determine the correct segmentation  $d^{[\tau],*} \subseteq \mathcal{C}^{[\tau]}$  and learning a MP library from the chosen segments.

The segments  $s_{i,j}^{[\tau]} \in \mathcal{S}^{[\tau]}$ , the segmentations  $d^{[\tau]} \in \mathcal{D}^{[\tau]}$  and the cuts  $c_i^{[\tau]} \in \mathcal{C}^{[\tau]}$  are always defined with respect to a trajectory  $\tau$ . For simplicity we will drop the superscript from now on.

The remainder of this paper is organized as follows. In Section II related work is presented and discussed followed by Section III where the Probabilistic Segmentation approach is introduced. In Section IV a chair assembly task is used to compare the proposed method to a baseline method and a state-of-the-art segmentation method. The experiment was executed and validated on a real robot platform.

## II. RELATED WORK

Algorithms for automatic segmentation have been investigated extensively, not only for the purposes of generation of robot skills but mainly as a general tool for movement analysis and classification. Hidden Markov Models (HMMs) have been widely adopted in this context. For example, in [3], video images are analyzed to train a HMM to

classify if a person is walking, running or crouching. In [4] automatic segmentation of motion patterns based on HMMs have been used to group segments hierarchically, where higher level representations of symbols can then be used to orchestrate and generate low level robot movements. More recently, in [5], the authors propose on-line segmentation method based on HMMs that creates a tree of primitives; the lower nodes representing detailed movements with generality increasing towards the root. HMMs have also been used in conjunction with the superposition of movement primitives for the specific case of handwriting analysis [6]. In general, HMMs and methods that explicitly address temporal sequences (e.g. [7]) have been generally accepted for segmentation. In this paper, however, we opt for a shape-based clustering approach on the basis that our desired library must be invariant to the possible combinations of skills transitions. The encoding of trajectories that do have a sequential pattern are naturally addressed by our method as it maintains only the most probable combinations of segments.

Our work takes advantage of movement primitive representations that are time invariant, such as Dynamical Movement Primitives (DMPs) [8] or Probabilistic Movement Primitives (ProMPs) [9]. Segmentation with movement primitives is particularly suited for library construction as segments with the same profile, but with different time scales are treated as the same skill. In [10], for example, the expected time scales of possible segments had to be taken into account with the introduction of a heuristic about the minimum and maximum duration of the skills; in our approach such user-defined inputs are not necessary.

From the movement primitive perspective, our algorithm relates to the work of [11], and [12] where DMPs have been used in different ways. In the first approach, a library of primitives is assumed given, while in our work we design our algorithm to start from an empty set. Compared to the work of [12], the authors treat segmentation as an independent initial step which therefore, later affects the reconstruction of a task, in this case using finite state automaton. As a consequence, interactive corrections given by a human demonstrator are introduced. In contrast, our approach treats segmentation and primitive learning as an iterative optimization process where both are intrinsically connected.

The concept of hierarchical skills has been explored in [4], [5], [13], and [14], and can be very efficient for on-line applications or to represent different granularities in the task. The philosophy of our method differs in the sense that we do not enrich a model by adding branches, but instead prune unnecessary segments given by a possibly erroneous initial heuristic. We leverage on batch, off-line learning to essentially reconstruct the segmentation-library, iteratively. This leads to a single level representation which decreases the number of segments as the library is improved after each EM iteration.

A general problem in movement segmentation and library generation is the trade off between the generality of the method and its tractability, the latter usually achieved by the introduction of heuristics. For example, Zero Crossing

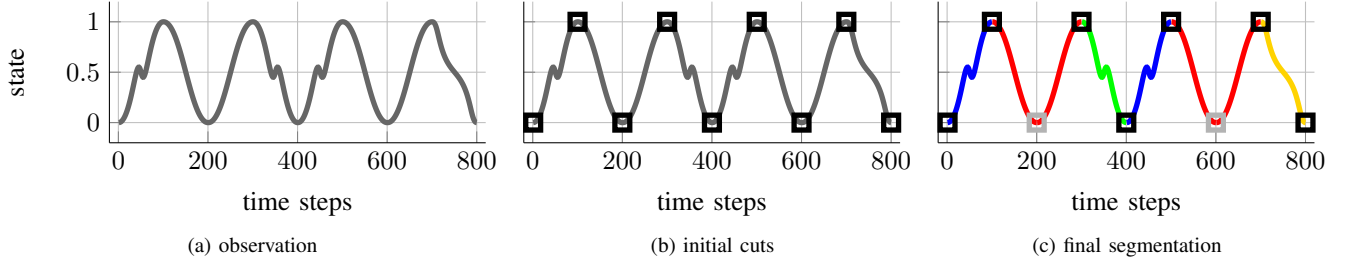


Fig. 2: An illustration of a possible segmentation. (a) shows a one dimensional, continuous observation. (b) shows the initially suggested cuts, illustrated as black squares. (c) shows a possible segmentation. Grey squares are eliminated false positive cuts.

Velocity (ZCV) has usually been used as an intuitive criterion to obtain the initial segmentation of trajectories [15], [16], [17]. In the context of movement primitives, however, ZCV usually leads to over-segmentation, especially when the robot moves at low speeds. In [18] the authors propose segmenting demonstrations based on geometric similarities. Other heuristics applied to segmentation include the two-thirds power law [19], velocity profiles and minimum jerk [20], and changes of the system dynamics [21]. Our work differs by being insensitive to the particular choice of the heuristic. Our assumption is that a given heuristic will lead to an initial number of excessive segments, which will be then optimized by decreasing the occurrence of cuts among them when necessary.

### III. LEARNING SKILL LIBRARIES USING PROBABILISTIC SEGMENTATION

We assume that each observation  $\tau \in \mathcal{T}$  consists of a single or multiple segments  $s \in \mathcal{S}$ , where  $\mathcal{S}$  denotes the set of all possible segments for trajectory  $\tau$ . Each segment  $s_{i,j}$  is defined by the cutting points at which the segment starts and ends  $c_i, c_j \in \mathcal{C}$ .

The method is initialized with a set of possible cutting points  $\mathcal{C}$ , which divides each trajectory into multiple segments as shown in Figure 2. Each demonstration  $\tau$  can consist of a single or multiple segments

$s_{i,j} \in \mathcal{S}$ , where  $s_{h,i}, s_{i,j}$ , with  $0 \leq h < i < j \leq |\mathcal{C}^{\tau}| + 1$

describe two consecutive segments, separated by a discrete cutting point  $c_i \in \mathcal{C}$ . Every possible combination of active cutting points within  $\mathcal{C}$  can be considered one possible segmentation  $d$ .

Our goal is to determine the correct segmentation  $d^*$  while simultaneously learning the underlying library  $\mathcal{M}$ . Since  $d^*$  is not known, we treat  $d^*$  as a latent variable. Unfortunately, the number of possible segmentations  $|\mathcal{D}| = 2^{|\mathcal{C}|}$  is exponential in the number of cuts  $|\mathcal{C}|$ , which makes a direct evaluation of  $\mathcal{D}$  unfeasible. However, we can equivalently evaluate the set of all possible segments  $\mathcal{S}$ , which is only quadratic  $|\mathcal{S}| = 0.5(|\mathcal{C}| + 1)(|\mathcal{C}| + 2)$  in the number of cuts. We therefore directly learn the optimal set of segments  $\mathcal{S}^* \subseteq \mathcal{S}$  instead of  $d^*$ . Therefore, we define a set of complete sets of segments  $\mathcal{R} = \{\mathcal{R} | \mathcal{R} \subset \mathcal{S}, \text{concat}(s \in \mathcal{R}) = \tau\}$ .

Each set of segments  $\mathcal{R} \in \mathcal{R}$  is complete in  $\tau$ , such that the concatenation of all segments  $s \in \mathcal{R}$  is equivalent to  $\tau$ . The proposed approach applies probabilistic inference methods to learn locally optimal  $\mathcal{M}$  and  $d^*$  in conjunction.

#### A. Defining the Cutting Points $\mathcal{C}$

Our proposed method depends on the set of possible cuts  $\mathcal{C}$ . Considering every time step of the observation as a cutting point is unfeasible. A possibility to restrict  $\mathcal{C}$  to a manageable size is to initially use heuristics to determine  $\mathcal{C}$ . These heuristics can be chosen task specifically and different heuristics can also be combined seamlessly. Alternatively, any kind of initial segmentation can be used to determine  $\mathcal{C}$ . However, the method only considers the cuts contained in  $\mathcal{C}$ , i.e., it is restricted to eliminate false positive cuts. Therefore,  $\mathcal{C}$  has to provide a weak over-segmentation, i.e.,  $\mathcal{C}^* \subseteq \mathcal{C}$ , where  $\mathcal{C}^*$  denotes the set of true cuts.

#### B. Generative Skill Model

Each skill  $m \in \mathcal{M}$  is represented by a parameterized, generative model. Therefore, a segment  $s$  can be considered as a sample drawn from a distribution representing the skill

$$s \sim p(s|\theta_k), \quad (1)$$

with  $\theta_k$  denoting the parameters of the  $k^{\text{th}}$  skill  $m_k$ . The skill library is defined as a mixture of skills, which allows computing the probability of a segment given the entire library

$$p(s|\Theta) = \sum_{k=1}^{|\mathcal{M}|} \lambda_k p(s|\theta_k), \quad \text{with} \quad (2)$$

$$\Theta = \{(\lambda_1, \theta_1), \dots, (\lambda_{|\mathcal{M}|}, \theta_{|\mathcal{M}|})\},$$

where  $\lambda_k$  denotes the mixture coefficient for skill  $m_k$ . Furthermore, we assume that every segment can be projected into a lower dimensional space describing the shape of the movement

$$v : s \rightarrow w, \quad s \in \mathcal{S} \text{ and } w \text{ being a real vector.} \quad (3)$$

Such projections are very common in movement primitives, e.g, Dynamic Movement Primitives [8] or Probabilistic Movement Primitives (ProMPs) [9]. In this work, we will use ProMPs as generative skill model.

ProMPs project trajectories into a weight space using a ridge regression and subsequently define a distribution

over the projected segments  $w$  corresponding to the given trajectories.

The projected segment  $w$  for each segment  $s$  are computed as

$$w = v(s) = \left( \Phi \Phi^T + \epsilon I \right)^{-1} \Phi s, \quad (4)$$

where  $\Phi$  denotes the feature matrices as defined in [9]. The features  $\Phi$  are usually represented as radial basis functions and depend on the duration of  $s$  and therefore render the weight  $w$  time invariant. Furthermore, if the segment  $s_{i,j}$  is a valid segment, there exists an underlying skill  $m_k$  with parameters  $\theta_k$  which produced the corresponding weights

$$w \sim \mathcal{N}(w|\theta_k), \theta_k = \{\mu_k, \Sigma_k\}.$$

We only consider correlations between the dimensions and not between the single time steps. Therefore, the probability of a segment  $s$  given a skill  $m_k$  is defined as

$$p(s|\theta_k) = \prod_{t=1}^{|s|} p(s_t|\theta_k),$$

where  $|s|$  denotes the number of time steps in  $s$ , hence it's duration, and  $s_t$  describes the segment  $s$  at the time step  $t$ .

The probability for a single time step  $t$  given the skill  $m_k$  is

$$p(s_t|\theta_k) = \mathcal{N}\left(s_t \mid \phi_t^T \mu_k, \phi_t^T \Sigma_k \phi_t\right),$$

with  $\phi_t$ ,  $\mu_k$  and  $\Sigma_k$  being the feature vectors and the skill parameters at time step  $t$ .

### C. Probabilistic Inference on Segmentations

We assume that every observation  $\tau \in \mathcal{T}$  was drawn from a parameterized generative model

$$\tau \sim p(\tau|\Theta, \mathcal{S}^*) = \prod_{s \in \mathcal{S}^*} p(s), [\Theta] \quad (5)$$

with  $\mathcal{S}^*$  being the set of correct segments. Since  $\mathcal{S}^*$  is unknown, we can treat it as a latent variable, and integrate it out. Such formulation leads to

$$p(\tau|\Theta) = \sum_{\mathcal{R} \in \mathcal{R}} p(\mathcal{R}) \prod_{s \in \mathcal{R}} p(s|\Theta),$$

where  $\mathcal{R}$  is the set of complete sets of segments. The most likely model  $\Theta$  is now determined by maximizing the likelihood

$$\Theta^* = \arg \max_{\Theta} \sum_{\tau \in \mathcal{T}} p(\tau|\Theta)$$

We optimize this likelihood by applying the EM algorithm [22] to the auxiliary function

$$Q(\Theta, \Theta') = \sum_{\tau \in \mathcal{T}} \sum_{s \in \mathcal{S}} \alpha_s \log p_s p(s|\Theta), \quad (6)$$

$$\text{with } \alpha_s = \sum_{\mathcal{R} \in \mathcal{R}} \mathbf{I}_{s \in \mathcal{R}} p(\mathcal{R}|\tau, \Theta'), \quad (7)$$

where the weighting is given by  $\alpha_s$ . The marginal  $\alpha_s$  weights each segment, depending on its probability to be part of the correct Segmentation  $s \in \mathcal{S}^*$  or not. The prior

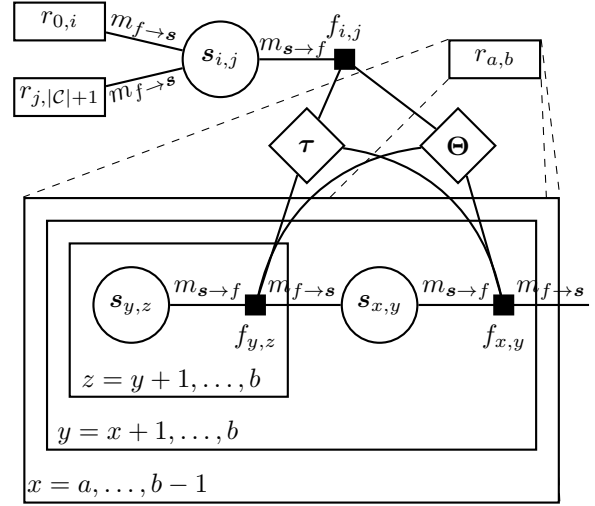


Fig. 3: The factor graph describing the relation between a specific segment  $s_{i,j}$  and all possible preceding and succeeding segments. The blocks  $r_{0,i}$  and  $r_{j,|C|+1}$  contain all possible preceding respectively succeeding segments. The messages,  $m_{f \to s}$  and  $m_{s \to f}$  allow an efficient computation of the marginal  $\alpha_{s_{i,j}}$ .

coefficient  $p_s$  emerges from defining the prior  $p(\mathcal{R})$  as a product of priors over segments

$$p(\mathcal{R}) = p_c \prod_{s \in \mathcal{R}} p_s \text{ with } p_s = (1 - p_c)^{j-i-1} p_c.$$

The constant  $0 < p_c < 1$  defines how probable it is that a cut is a true positive cut and  $i, j$  are indices of the two cuts limiting  $s$ . For  $p_c > 0.5$  segments which span over multiple cutting points are preferred, whereas  $p_c < 0.5$  indicates that longer segments are preferable.

Given the function defined in Equation 6, the EM algorithm iteratively computes the weighting  $\alpha_s$  in E-Step and updates the current model estimate  $\Theta'$  in the M-Step.

### D. Expectation Step: Computing the Probability of the Segments

In the E-Step, the marginal as described in Equation 6 is updated, and therefore the segments  $s \in \mathcal{S}$  are re-weighted. In order to efficiently compute  $\alpha_s$ , Equation 7 is reformulated by applying Bayes Theorem on Equation 5

$$\alpha_s = \frac{1}{Z} \sum_{\mathcal{R} \in \mathcal{R}} \mathbf{I}_{s \in \mathcal{R}} \prod_{s \in \mathcal{R}} p_s p(s|\Theta'), \quad (8)$$

$$Z = \sum_{\mathcal{R} \in \mathcal{R}} \prod_{s \in \mathcal{R}} p_s p(s|\Theta'),$$

where  $Z$  denotes the normalizing constant. Such a formulation is well studied in Graphical Models and can be solved efficiently using message passing algorithms. Figure 3 describes the factor graph corresponding to this particular case. The factors and messages are given as

$$f_{i,j} = p_{s_{i,j}} p(s_{i,j}|\tau, \Theta').$$

For each segment  $s_{i,j} \in \mathcal{S}$  the for the remaining nodes can be computed as

$$m_{s \rightarrow f}(s_{x,y}) = \begin{cases} 1, & \text{if } y = b \\ \sum_{k=j+1}^b m_{f \rightarrow s}(s_{y,z}), & \text{otherwise} \end{cases}$$

$$m_{f \rightarrow s}(s_{i,j}) = m_{s \rightarrow f}(s_{x,y}) f_{x,y},$$

with  $b \in i, |C| + 1$  depending if we compute the messages for all possible preceding or succeeding segments. Since message re-occur for different  $s_{i,j}$ , these messages can be computed once and reused for different segments. The complexity to compute  $\alpha_s$  is quadratic in the number of cuts and the marginal can be computed as

$$\alpha_s = \frac{1}{Z} m_{s \rightarrow f}(s) f_{i,j}, \quad (9)$$

$$\text{with } Z = m_{s \rightarrow f}(s_{0,|C|+1}). \quad (10)$$

The message formulation in Equation 10 grows quadratically with the number of cuts, and therefore allows the algorithm to consider significantly more cuts than a naive application of Equation 6.

#### E. Maximization Step: Learning the Skill Library

In the maximization step, the model parameters  $\Theta$  are updated by maximizing  $Q(\Theta, \Theta')$ . We assume that all observed demonstrations were generated by the same underlying model, implying that the model update considers all possible segments independently of their corresponding demonstration. Following [9], we update the model based on the projected segments rather than the trajectories directly. Therefore, the maximization step is defined over the set of projected segments

$$\mathcal{W} = \left\{ \mathbf{w} \mid \mathbf{w} = v(s), \forall s \in \mathcal{S}^{[\tau]}, \forall \tau \in \mathcal{T} \right\}, \quad (11)$$

where  $v$  is the projection function defined in Equation 3. In addition, we also define  $\alpha_s$  over the projected vectors

$$\alpha_w = \alpha_s \iff \mathbf{w} = v(s).$$

Given these definitions, the reformulated auxiliary function

$$Q(\Theta, \Theta') = \sum_{\mathbf{w} \in \mathcal{W}} \alpha_w \log p(\mathbf{w} \mid \Theta),$$

takes the form of a weighted log likelihood. Maximizing  $Q(\Theta, \Theta')$  directly with respect to  $\Theta$  is unfeasible as  $p(\mathbf{w} \mid \Theta)$  is a mixture model containing latent variables as we don't know which projected segment belongs to which skill. It is, however, possible to estimate  $\Theta$  by maximizing the weighted maximum likelihood. Given the nature of our model we can again apply a weighted EM algorithm on Gaussian Mixture Models (GMM), with the auxiliary function

$$Q_2(\Theta, \Theta') = \sum_{k=1}^{|\mathcal{M}|} \sum_{\mathbf{w} \in \mathcal{W}} \alpha_w \beta_{k,\mathbf{w}} \log \lambda_k p(\mathbf{w} \mid \theta_k)$$

$$\text{s.t. } \sum_{k=1}^{|\mathcal{M}|} \lambda_k = 1,$$

where  $\beta_{k,\mathbf{w}} = p(k \mid \mathbf{w}, \Theta')$  are typically referred to as the responsibilities. The EM algorithm for GMMs is widely applied and well known.

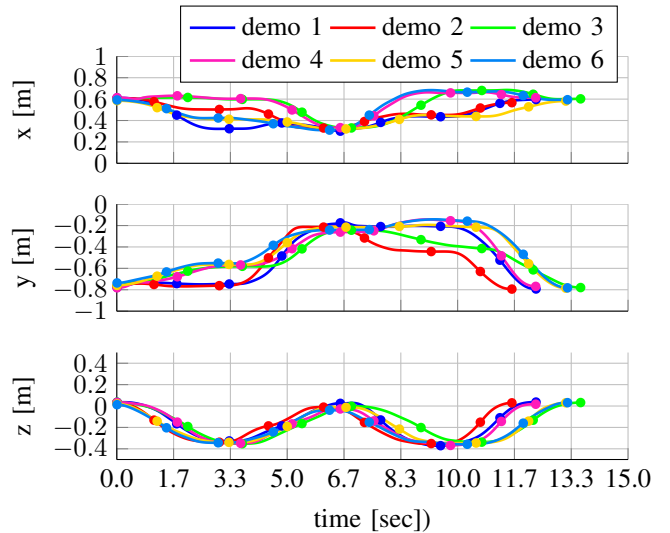


Fig. 4: The first three dimensions, corresponding to the Cartesian position, of the six observations. Each observation demonstrates the insertion of a chair leg into a hole in the seat. The cuts determined by the initial heuristic are illustrated as dots.

A known disadvantage of EM for GMMs is that the number of skills  $|\mathcal{M}|$  has to be known a priori. In our approach, however,  $|\mathcal{M}|$  depends on the latent segmentation, and can therefore change every time we change the weighting of the segments. We circumvent this problem by using the Gaussian-means algorithm [23] to determine the number of skills, including an initial labeling for the EM Algorithm. The Gaussian-means algorithm is a bisecting k-means algorithm which uses a test based on the Anderson-Darling statistic to determine if the data assigned to a cluster is Gaussian or not. If the data is not Gaussian the cluster is split. The ProbS method is summarized in Algorithm 1.

---

#### Algorithm 1: Probabilistic Segmentation

---

**input** : The set of all possible segments  $\mathcal{S}$   
**output** : The mixture model  $\Theta^*$  explaining  $\mathcal{S}$

$K$  : current number of clusters

$L$  : current labeling of the segments

**while not converged do**

**E-Step** : – compute the weighting  $\alpha_{s_{i,j}}$  as described in Equation 9

**M-Step**: – compute  $\mathcal{W}$  according to Equation 11

– determine  $K$  and  $L$  by applying Gaussian-means on  $\mathcal{W}$

– update  $\Theta^*$  by using an EM-GMM on  $\mathcal{W}$  with  $K$  clusters and initial labeling  $L$

---

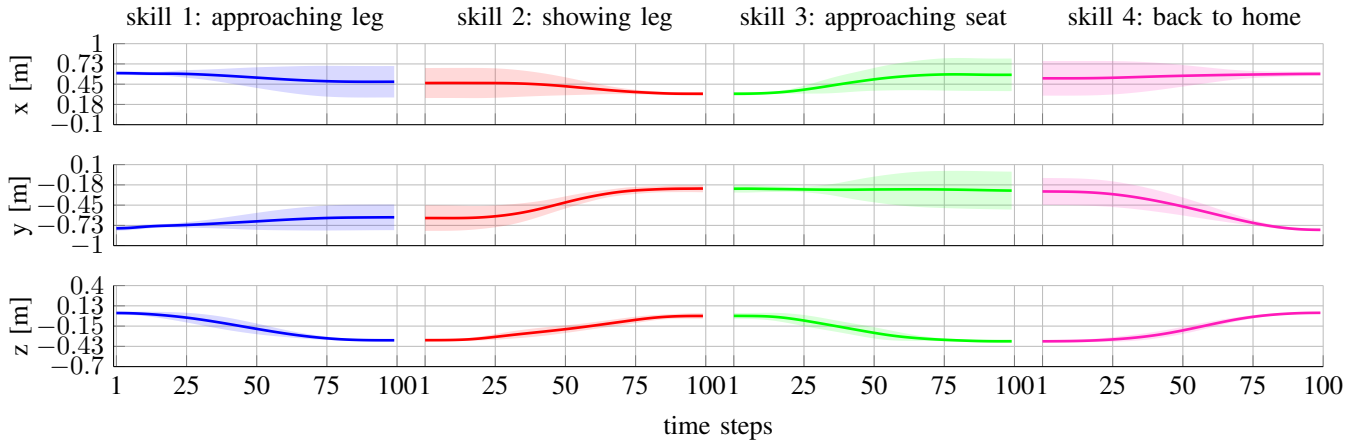


Fig. 5: The first three dimensions of the four skills learned by ProbS. The dark line shows the mean and the shaded area corresponds to two times the standard deviation. The skills are semantically meaningful, i.e., approaching a leg, showing the leg to a camera, approaching the seat and going back to the home position.

#### IV. EXPERIMENTS

We evaluated and compared ProbS on a chair assembly task. In order to show that useful skills were extracted, we replayed the skills on the real robot, while conditioning the skills to assemble the chair in and undemonstrated order. Figure 1 shows the experimental platform, a seven DoF KUKA lightweight arm equipped with a five finger DLR HIT Hand II as end effector.

The assembly was demonstrated by a human, as seen in Figure 4. The wrist of the human was tracked using the OptiTrack motion capture system. A total of six demonstrations were performed, where each demonstration consisted of four phases, i.e., approaching and picking up a chair leg, showing the leg tip to the camera, approaching and inserting the leg into the seat and finally retreating to the home position. Each data point is a seven dimensional vector containing the wrist position in the three dimensional space and the orientation encoded as a quaternion. The tracked positions are shown in

Figure 4.

ProbS successfully identified the true positive cutting points and learned a total of four skills, corresponding to the four steps in each demonstration. The learned skills are shown in Figure 5.

The initial over-segmentation was produced by a velocity based heuristic where the cutting points were positioned at the extrema of the velocity profile for each observation. The heuristic resulted in nine cuts per observation leading to a total of 768 possible segmentations or equivalently 216 segments. Even though the applied heuristic results in the same number of cuts per demonstration, this is not a requirement for ProbS.

In order to demonstrate the advantage of optimizing both, the segmentation and the skill library iteratively, we chose an Expectation-Maximization algorithm over Gaussian mixture models (EM-GMM) as a baseline, where the number of clusters as well as the initial labeling is determined by the

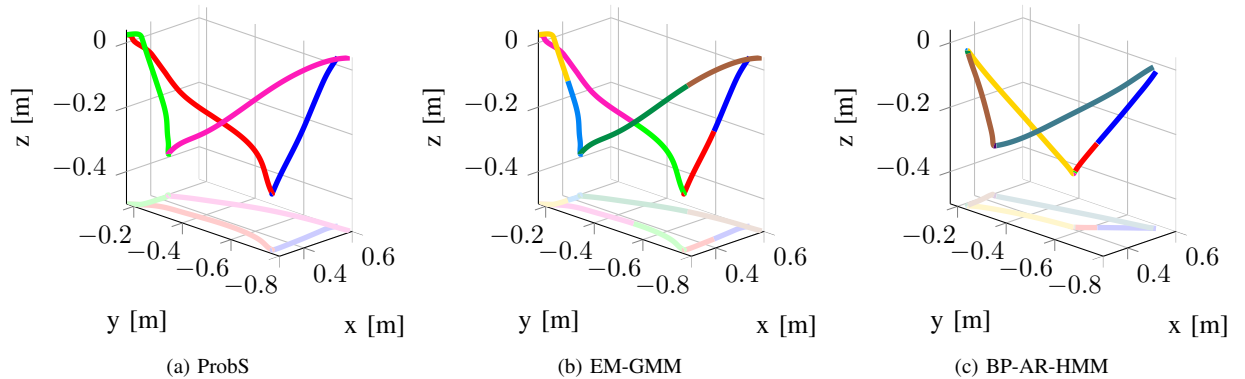


Fig. 6: The first demonstrations explained by the three different methods. Different colors within each plot illustrated different skills. ProbS explains the demonstration with four skills, EM-GMM identified eight skills in the demonstration and BP-AR-HMM found separated the demonstration into a total of 11 segments. Most of the segments span over a very short amount of time and are not visible in the plot.

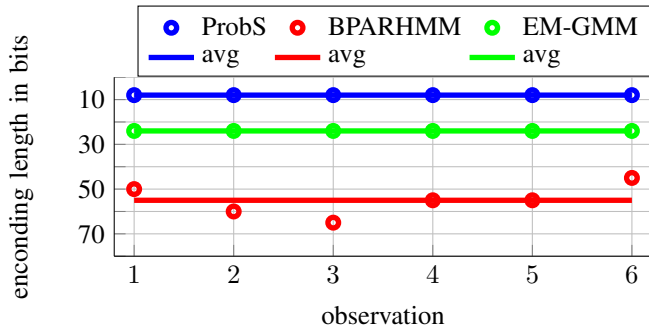


Fig. 7: The encoding length of the six demonstrations for each of the three methods. Less is better. ProbS and EM-GMM explain each observation with the same number of skills, that leads to a constant encoding length throughout the observations. BP-AR-HMM produces a larger variety of segments which leads to a higher encoding length.

Gaussian-means algorithm. In addition, we compared our method to the state-of-the-art, non-parametric segmentation method Beta Process Autoregressive Hidden Markov Model (BP-AR-HMM), as applied in [24] and [25].

#### A. Learned Skills

ProbS was able to identify four underlying skills. Each of the skills occurs exactly once in every demonstration and corresponds to the four steps of each demonstration. The first three dimensions are shown in Figure 5. It is clearly visible that the skills preserve a characteristic shape while the variance at certain time steps shows the adaptability of the skill at those points.

Given the found number of skills and the initial labeling of the Gaussian-means algorithm, the EM-GMM baseline identified a total of eight skills. Analogously to ProbS each found skill is present exactly once in each demonstration. The identified skills of the first demonstration are shown in Figure 6.

The BP-AR-HMM method does not rely on an initial segmentation, however, because of the expensive sampling process, the algorithm is computationally very expensive. We evaluated various hyper parameters and compared to the best found solution. However, the method failed to identify most of the re-occurring movements throughout the demonstrations and identified a total of 19 skills.

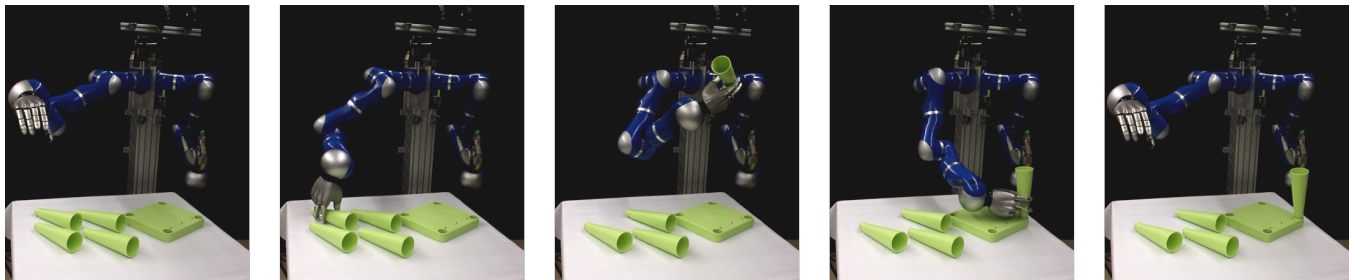
#### B. Encoding Length

As a metric to quantify the value of the segmentation we compute the number of bits necessary to encode each observation given the identified segments. The coding length for each observation as well as the average is shown in Figure 7.

Both ProbS and EM-GMM found skills which occur exactly once in every demonstration. This results a constant encoding length throughout the demonstrations. In the case of EM-GMM this results in 24 bits per observation. By reducing the number of active cuts and learning a small set of skills ProbS achieves an average encoding length of 8 bits. BP-AR-HMM found a total number of 19 segments. Each observation is explained by more segments than in either the EM-GMM or ProbS. This leads to a high average encoding length of 55 bits. This evaluation shows, that ProbS is able to find segments, that allow a compact representation of the observations.



(a) Demonstration of the chair assembly via the OptiTrack motion capture system.



(b) Execution of the learned skills.

Fig. 8: Demonstration and execution of the chair assembly task. The human was tracked during the chair assembly. The observed demonstrations were subsequently segmented using ProbS. The skills of the learned library were subsequently sequenced and executed on a robot platform to assemble the chair.

### C. Chair Assembly

Finally, we show the applicability of our method in real robot scenarios, by assembling an Ikea chair using the learned skill library. As shown in Figure 8, the robot is able to extract the necessary skills from the given demonstrations. The start and end point of each skill were conditioned to the corresponding point of interest, e.g. the "inserting" skill was conditioned to the hole position. Since each skill was learned from only six samples, the variance at some points was too low to successfully condition the corresponding ProMP. We therefore scaled the covariance matrix of each skill artificially. This step would be unnecessary if there had been sufficient demonstrations available. This experiment shows that ProbS is able to segment entire demonstrations to extract meaningful skills. These skills can be used and sequenced in order to solve observed as well as new tasks. For example, the chair was assembled with combinations of legs and holes which were not present in the demonstrations.

## V. CONCLUSION

We proposed a new algorithm for segmentation of unlabeled trajectories. The algorithm builds a skill library that is used to infer correct segmentations. The skill library as well as the inferred segmentations of the trajectories are iteratively optimized as there is a high dependency between both entities. Our algorithm takes advantage of heuristics that are used to over-segment the trajectories. In comparison to other state of the art methods such as non-parametric autoregressive HMMs, our algorithm has less hyper-parameters to fine tune and clear computational advantages. Finally, the returned skill library looked much more compact than the ones retrieved with related approaches.

In future work we will concentrate on reducing the reliance on the heuristics as well as learning high level control variables of each skill, such as possible conditioning points.

## ACKNOWLEDGMENTS

We would like to thank Emily Fox and Scott Niekum for their implementations and valuable feedback.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7-ICT-2013-10) under grant agreement 610878 (3rdHand) and from the DFG under grant agreement Learn-Robots (SPP1527).

## REFERENCES

- [1] K. Mülling, J. Kober, and J. Peters, "Learning Table Tennis with a Mixture of Motor Primitives," in *IEEE/RAS International Conference and Humanoid Robotics*, 2010, pp. 411–416.
- [2] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47, no. 2, pp. 79–91, 2004.
- [3] M. Brand and V. Kettner, "Discovery and segmentation of activities in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 844–851, 2000.
- [4] W. Takano and Y. Nakamura, "Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 425–431.
- [5] 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.
- [6] B. Williams, M. Toussaint, and A. J. Storkey, "Modelling motion primitives and their timing in biologically executed movements," in *Advances in Neural Information Processing Systems*, 2008, pp. 1609–1616.
- [7] A. d'Avella and M. C. Tresch, "Modularity in the motor system: decomposition of muscle patterns as combinations of time-varying synergies," *Advances in Neural Information Processing Systems*, vol. 1, pp. 141–148, 2002.
- [8] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [9] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, 2013, pp. 2616–2624.
- [10] S. Chiappa and J. Peters, "Movement extraction by detecting dynamics switches and repetitions," in *Advances in neural information processing systems*, 2010, pp. 388–396.
- [11] F. Meier, E. Theodorou, F. Stulp, and S. Schaal, "Movement segmentation using a primitive library," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3407–3412.
- [12] S. Niekum, S. Chitta, B. Marthi, S. Osentoski, and A. Barto, "Incremental semantically grounded learning from demonstration," in *Robotics: Science and Systems*, vol. 9, 2013.
- [13] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, 2011.
- [14] K. Yamane, Y. Yamaguchi, and Y. Nakamura, "Human motion database with a binary tree and node transition graphs," *Autonomous Robots*, vol. 30, no. 1, pp. 87–98, 2011.
- [15] A. Fod, M. J. Matarić, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous robots*, vol. 12, no. 1, pp. 39–54, 2002.
- [16] A. Nakazawa, S. Nakaoka, K. Ikeuchi, and K. Yokoi, "Imitating human dance motions through motion structure analysis," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2002, pp. 2539–2544.
- [17] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, "Segmenting motion capture data into distinct behaviors," in *Proceedings of the 2004 Graphics Interface Conference*, 2004, pp. 185–194.
- [18] A. Lemme, F. Reinhard, and J. J. Steil, "Semi-supervised bootstrapping of a movement primitive library from complex trajectories," in *Proceedings of the International Conference on Humanoid Robots*, 2014, pp. 726–732.
- [19] D. Endres, Y. Meirovitch, T. Flash, and M. A. Giese, "Segmenting sign language into motor primitives with bayesian binning," *Frontiers in Computational Neuroscience*, vol. 7, 2013.
- [20] B. Rohrer and N. Hogan, "Avoiding spurious submovement decompositions ii: a scattershot algorithm," *Biological Cybernetics*, vol. 94, no. 5, pp. 409–414, 2006.
- [21] O. Kroemer, H. van Hoof, G. Neumann, and J. Peters, "Learning to predict phases of manipulation tasks as hidden states," in *Proceedings of 2014 IEEE International Conference on Robotics and Automation*, 2014.
- [22] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer New York, 2006, vol. 1.
- [23] G. Hamerly and C. Elkan, "Learning the k in k-means," in *Advances in Neural Information Processing Systems*, 2003, pp. 281–288.
- [24] E. Fox, "Bayesian nonparametric learning of complex dynamical phenomena," Ph.D. Thesis, MIT, Cambridge, MA, 2009.
- [25] S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto, "Learning and generalization of complex tasks from unstructured demonstrations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5239–5246.