Learning Multiple Collaborative Tasks with a Mixture of Interaction Primitives

Marco Ewerton[1], Gerhard Neumann[1], Rudolf Lioutikov[1], Heni Ben Amor[2], Jan Peters[1,3] and Guilherme Maeda[1]

Abstract— **Robots that interact with humans must learn to not only adapt to different human partners but also to new interactions. Such a form of learning can be achieved by demonstrations and imitation. A recently introduced method to learn interactions from demonstrations is the framework of *Interaction Primitives*. While this framework is limited to represent and generalize a single interaction pattern, in practice, interactions between a human and a robot can consist of many different patterns. To overcome this limitation this paper proposes a *Mixture of Interaction Primitives* to learn multiple interaction patterns from unlabeled demonstrations. Specifically the proposed method uses Gaussian Mixture Models of Interaction Primitives to model nonlinear correlations between the movements of the different agents. We validate our algorithm with two experiments involving interactive tasks between a human and a lightweight robotic arm. In the first, we compare our proposed method with conventional Interaction Primitives in a toy problem scenario where the robot and the human are not linearly correlated. In the second, we present a proof-of-concept experiment where the robot assists a human in assembling a box.**

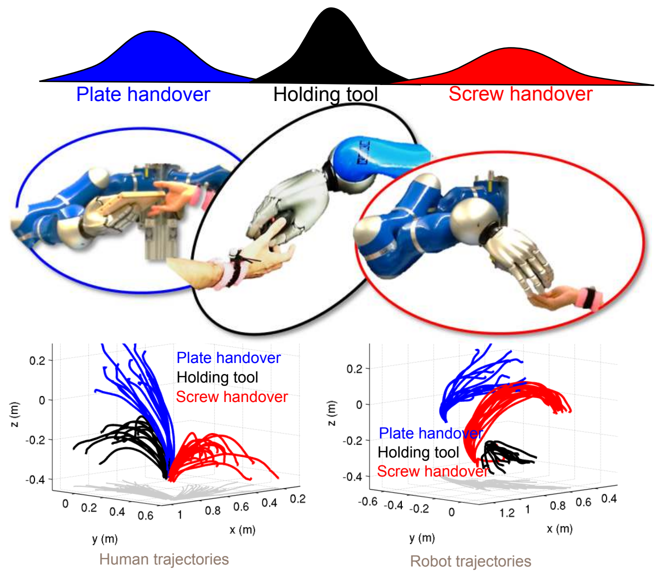


Fig. 1. Illustration of a task consisting of multiple interaction patterns, where each can be represented as an Interaction Primitive. In this work, we want to learn multiple interaction patterns from an unlabeled data set of interaction trajectories.

I. Introduction

Robots that can assist us in the industry, in the household, in hospitals, etc. can be of great benefit to the society. The variety of tasks in which a human may need assistance is, however, practically unlimited. Thus, it is very hard (if not impossible) to program a robot in the traditional way to assist humans in scenarios that have not been exactly prespecified.

Learning from demonstrations is therefore a promising idea. Based on this idea, Interaction Primitive (IP) is a framework that has been recently proposed to alleviate the problem of programming a robot for physical collaboration and assistive tasks [1], [2]. At the core, IPs are primitives that capture the correlation between the movements of two agents—usually a human and a robot. Then, by observing one of the agents, say the human, it is possible to infer the controls for the robot such that collaboration can be achieved.

A main limitation of IPs is the assumption that the movements of the human and the movements of the robot assistant are linearly correlated. This assumption is reflected in the underlying Gaussian distribution that is used to model the demonstrations. While this assumption holds for tasks that cover a small region of the workspace (a high-five task in [1] or handover of objects in [2]), it limits the use of IPs in two aspects. First, as illustrated in Fig. 1, a task such as the assembly of a toolbox consists of several interaction patterns that differ significantly from each other and therefore can not be captured by a single Gaussian. Moreover, even within a single interaction pattern, the correlation between the two agents may be nonlinear, for example, if the movements of the human are measured in the Cartesian space, while the movements of the robot are measured in joint space.

Manually labeling each subtask (e.g. “plate handover", “screw handover", “holding screw driver") is a way to model interactions with multiple subtasks. Ideally, however, robots should be able to identify different subtasks by themselves. Moreover, it may not be clear to a human how to separate a number of demonstrated interactions in different, linearly correlated groups. Thus, a method to learn multiple interaction patterns from unlabeled demonstrations is necessary. The main contribution of this paper is the development of such a method. In particular, this work uses Gaussian Mixture Models (GMMs) to create a Mixture of Interaction

[1]Intelligent Autonomous Systems Lab, Department of Computer Science, Technische Universität Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany {ewerton, neumann, lioutikov, peters, maeda}@ias.tu-darmstadt.de

[2]Institute of Robotics and Intelligent Machines, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332-0280, USA hbenamor@cc.gatech.edu

[3]Max Planck Institute for Intelligent Systems, Spemannstr. 38, 72076 Tuebingen, Germany jan.peters@tuebingen.mpg.de

Probabilistic Movement Primitives [2].

The remainder of this paper is organized as follows: Section II presents related work. In Section III, Probabilistic Movement Primitives (ProMPs) and Interaction ProMPs are briefly introduced, followed by the proposition of the main contribution of this paper: a Mixture of Interaction ProMPs based on Gaussian Mixture Models (GMMs). Section IV evaluates the proposed method first on a toy problem that is useful to clarify the characteristics of the method and then on a practical application of a collaborative toolbox assembly. Section V presents conclusions and ideas for future work.

II. Related Work

Physical human-robot interaction poses the problem of both action recognition and movement control. Interaction dynamics need to be specified in a way that allows for robust reproduction of the collaborative task under different external disturbances, and a common approach is based on direct force sensing or emulation. Rozo et al. [3] propose a framework for haptic collaboration between a human and a robot manipulator. Given a set of kinesthetic demonstrations, their method learns a mapping between measured forces and the impedance parameters used for actuating the robot, e.g., the stiffness of virtual springs governing the collaborative task. In another force-based approach, Lawitzky et al. [4] propose learning physical assistance in a collaborative transportation task. In the early learning phase, the robot uses the measured force values to follow the human guidance during the task. Recorded force and motion patterns are then used to learn a Hidden Markov Model (HMM) which can predict the human's next action and over time the robot learns to take over a more active role in the interaction. Kulvicius et al. [5] also address a transportation task where the two agents are modeled as two point particles coupled by a spring. The forces applied by the other agent tell the robot how to adapt its own trajectory.

Our work differs significantly from the cited works in the sense that our method does not use nor emulate force signals, but instead learns the correlation between the trajectories of two agents. Correlating trajectories not only simplifies the problem in terms of hardware and planning/control but also allows us to correlate multi-agent movements that do not generate force during the interaction, for example, the simple gesture of asking and receiving an object.

Graphical models have also been used to describe interaction dynamics. In the computer vision community, HMMs have been widely adopted to model interaction dynamics from input video streams [6], [7]. As a result, graphical models have also gained considerable attention in the field of human-robot interaction. In [8], Hawkins and colleagues use a Bayes network to improve the fluency in a joint assembly task. The Bayes network learns to infer the current state of the interaction, as well as task constraints and the anticipated timing of human actions. Tanaka et al. [9] use a Markov model to predict the positions of a worker in an assembly line. Wang et al. [10] propose the Intention-Driven Dynamics Model (IDDM) as a probabilistic graphical model with observations, latent states and intentions where the transitions between latent states and the mapping from latent states to observations are modeled as Gaussian Processes. Koppula et al. [11] use a conditional random field with sub-activities, human poses, object affordances and object locations over time. Inference on the graphical model allows a robot to anticipate human activity and choose a corresponding, preprogrammed robot response. Lee et al. [12] learn a hierarchical HMM which triggers action primitives in response to observed behaviors of a human partner.

While very successful for classifying actions, graphical models, however, may not be the best option when it comes to generating motions. In [13], for example, the use of a HMM with discrete states, although very successful in action classification, introduces artifacts into the motion generation part that hinders motion generalization. Therefore, a clear problem in physical human-robot interaction is that while graphical models may be suitable in the action recognition domain, motion generation at the continuous level must also be taken into account. Llorens et al. [14] present a hybrid design for a robot to be used on the shoulder. In their work, Petri Nets accounts for discrete control transitions while, at the motion level, Partial Least Squares Regression has been used to find the best action of the robot at future time steps.

Perhaps the principal distinction of our method is the use of Interaction Primitives (IPs), introduced by Ben Amor et al. [1], initially based on Dynamical Movement Primitives [15] and later extended to Probabilistic Movement Primitives [16] with action recognition in the work of Maeda et al. [2]. As shown in [2], Interaction Primitives can be used to not only recognize the action of an agent, but also to coordinate the actions of a collaborator at the movement level; thus overcoming in a single framework both layers of discrete action recognition and continuous movement control. Differently from [2], where different interaction patterns must be hand-labeled, our contribution is the unsupervised learning of a Mixture of Interaction Primitives.

III. Mixture of Interaction Primitives

In this section, we will briefly discuss the Interaction Primitive framework based on Probabilistic Movement Primitives [2], [16], followed by the presentation of the proposed method, based on Gaussian Mixture Models.

A. Probabilistic Movement Primitives

A Probabilistic Movement Primitive (ProMP) [16] is a movement representation based on a distribution over trajectories. The probabilistic formulation of a movement primitive allows operations from probability theory to seamlessly combine primitives, specify via points, and correlate joints via conditioning. Given a number of demonstrations, ProMPs are designed to capture the variance of the positions q and velocities $\dot{q}$ as well as the covariance between different joints.

For simplicity, let us first consider only the positions q for one degree of freedom (DOF). The position q_t at time step t can be approximated by a linear combination of basis functions,

$$q_t = \boldsymbol{\psi}_t^T \boldsymbol{w} + \epsilon, \tag{1}$$

where ϵ is Gaussian noise. The vector $\boldsymbol{\psi}_t$ contains the N basis functions ψ_i, $i \in \{1,\ 2,\ 3, ...,\ N\}$, evaluated at time step t where we will use the standard normalized Gaussian basis functions.

The weight vector $\boldsymbol{w}$ is a compact representation of a trajectory[1]. Having recorded a number of trajectories of q, we can infer a probability distribution over the weights $\boldsymbol{w}$. Typically, a single Gaussian distribution is used to represent $p(\boldsymbol{w})$. While a single $\boldsymbol{w}$ represents a single trajectory, we can obtain a distribution $p(q_{1:T})$ over trajectories $q_{1:T}$ by integrating $\boldsymbol{w}$ out,

$$p(q_{1:T}) = \int p(q_{1:T}|\boldsymbol{w})p(\boldsymbol{w})\mathrm{d}\boldsymbol{w}. \tag{2}$$

If $p(\boldsymbol{w})$ is a Gaussian, $p(q_{1:T})$ is also Gaussian. The distribution $p(q_{1:T})$ is called a Probabilistic Movement Primitive (ProMP).

B. Interaction ProMP

An Interaction ProMP builds upon the ProMP formulation, with the fundamental difference that we will use a distribution over the trajectories of all agents involved in the interaction. Hence, $\boldsymbol{q}$ is multidimensional and contains the positions in joint angles or Cartesian coordinates of all agents. In this paper, we are interested in the interaction between two agents, here defined as the observed agent (human) and the controlled agent (robot). Thus, the vector $\boldsymbol{q}$ is now given as $\boldsymbol{q} = [(\boldsymbol{q}^o)^T,\ (\boldsymbol{q}^c)^T]^T$, where $(\cdot)^o$ and $(\cdot)^c$ refer to the observed and controlled agent, respectively.

Let us suppose we have observed a sequence of positions $\boldsymbol{q}_t^o$ at m specific time steps t, $m \leq T$. We will denote this sequence by D. Given those observations, we want to infer the most likely remaining trajectory of both the human and the robot.

Defining $\bar{\boldsymbol{w}} = [\boldsymbol{w}_o^T,\ \boldsymbol{w}_c^T]^T$ as an augmented vector that contains the weights of the human and of the robot for one demonstration, we write the conditional probability over trajectories $\boldsymbol{q}_{1:T}$ given the observations D of the human as

$$p(\boldsymbol{q}_{1:T}|D) = \int p(\boldsymbol{q}_{1:T}|\bar{\boldsymbol{w}})p(\bar{\boldsymbol{w}}|D)\mathrm{d}\bar{\boldsymbol{w}}. \tag{3}$$

We compute a normal distribution from n demonstrations by stacking several weight vectors $[\bar{\boldsymbol{w}}_1,\ ...,\ \bar{\boldsymbol{w}}_n]^T$, one for each demonstration, such that $\bar{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$. A posterior distribution can be obtained after observing D with

$$\begin{aligned} \boldsymbol{\mu}_w^{new} &= \boldsymbol{\mu}_w + \boldsymbol{K}(D - \boldsymbol{H}_t^T\boldsymbol{\mu}_w), \\ \boldsymbol{\Sigma}_w^{new} &= \boldsymbol{\Sigma}_w - \boldsymbol{K}(\boldsymbol{H}_t^T\boldsymbol{\Sigma}_w), \end{aligned} \tag{4}$$

[1]In order to cope with the different speeds of execution during demonstration, the trajectories must be time-aligned before parameterization. The interested reader is referred to [2] for details.

where $\boldsymbol{K} = \boldsymbol{\Sigma}_w\boldsymbol{H}_t^T(\boldsymbol{\Sigma}_D + \boldsymbol{H}_t^T\boldsymbol{\Sigma}_w\boldsymbol{H}_t)^{-1}$, $\boldsymbol{\Sigma}_D$ is the observation noise, and

$$\boldsymbol{H}_t = \left[\begin{array}{cc:cc} (\boldsymbol{\psi}_t^o)_{(1,1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\boldsymbol{\psi}_t^o)_{(P,P)} & \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{0}_{(1,1)}^c & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}_{(Q,Q)}^c \end{array}\right] \tag{5}$$

is the observation matrix where the unobserved states of the robot are filled with zero bases. Here, the human and the robot are assumed to have P and Q DOFs, respectively.

Now, by combining (1), (3) and (4), we can compute the probability distribution over the trajectories $\boldsymbol{q}_{1:T}$ given the observation D. For a detailed implementation the interested reader is referred to [2].

C. Mixture of Interaction ProMPs

The goal of our method is to learn several interaction patterns given the weight vectors that parameterize our unlabeled training trajectories. For this purpose, we learn a GMM in the weight space, using the Expectation-Maximization algorithm (EM) [17].

Assume a training set with n vectors $\bar{\boldsymbol{w}}$ representing the concatenated vectors of human-robot weights as defined in section III-B. In order to implement EM for a GMM with a number K of Gaussian mixture components, we need to implement the Expectation step and the Maximization step and iterate over those steps until convergence of the probability distribution over the weights, $p(\bar{\boldsymbol{w}}; \alpha_{1:K}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K})$, where $\alpha_{1:K} = \{\alpha_1, \alpha_2, \cdots, \alpha_K\}$, $\boldsymbol{\mu}_{1:K} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_K\}$ and $\boldsymbol{\Sigma}_{1:K} = \{\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \cdots, \boldsymbol{\Sigma}_K\}$. Here, $\alpha_k = p(k)$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the prior probability, the mean and the covariance matrix of mixture component k, respectively. We initialize the parameters $\alpha_{1:K}$, $\boldsymbol{\mu}_{1:K}$ and $\boldsymbol{\Sigma}_{1:K}$ using k-means clustering before starting the Expectation-Maximization loop. The number K of Gaussian mixture components is found by leave-one-out cross-validation.

The mixture model can be formalized as

$$p(\bar{\boldsymbol{w}}) = \sum_{k=1}^{K} p(k)p(\bar{\boldsymbol{w}}|k) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\bar{\boldsymbol{w}}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{6}$$

Expectation step: Compute the *responsibilities* r_{ik}, where r_{ik} is the probability of cluster k given weight vector $\bar{\boldsymbol{w}}_i$,

$$r_{ik} = p(k|\bar{\boldsymbol{w}}_i) = \frac{\mathcal{N}(\bar{\boldsymbol{w}}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\alpha_k}{\sum_{l=1}^{K} \alpha_l \mathcal{N}(\bar{\boldsymbol{w}}_i; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}. \tag{7}$$

Maximization step: Update the parameters α_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ of each cluster k, using

$$n_k = \sum_{i=1}^{n} r_{ik},\ \ \alpha_k = \frac{n_k}{n}, \tag{8}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n} r_{ik}\bar{\boldsymbol{w}}_i}{n_k}, \tag{9}$$

$$\mathbf{\Sigma}_k = \frac{1}{n_k}\left(\sum_{i=1}^{n} r_{ik}(\bar{\boldsymbol{w}}_i - \boldsymbol{\mu}_k)(\bar{\boldsymbol{w}}_i - \boldsymbol{\mu}_k)^T\right). \tag{10}$$

Finally, we want to use our model to infer the trajectories of the controlled agent given observations from the observed agents. We need to find the posterior probability distribution over trajectories $\mathbf{q}_{1:T}$ given the observations D, as in Section III-B.

In order to compute this posterior using our GMM prior, first we find the most probable cluster k^* given the observation D, using the Bayes' theorem. The posterior over the clusters k given the observation D is given by

$$p(k|D) \propto p(D|k)p(k), \tag{11}$$

where

$$p(D|k) = \int p(D|\bar{\boldsymbol{w}})p(\bar{\boldsymbol{w}}|k)\mathrm{d}\bar{\boldsymbol{w}}$$

and

$$p(\bar{\boldsymbol{w}}|k) = \mathcal{N}(\bar{\boldsymbol{w}}; \boldsymbol{\mu}_k, \mathbf{\Sigma}_k).$$

Thus the most probable cluster k^* given the observation D is

$$k^* = \arg\max_k p(k|D). \tag{12}$$

The output of the proposed algorithm is the posterior probability distribution over trajectories $\mathbf{q}_{1:T}$, conditioning cluster k^* to the observation D,

$$p\left(\mathbf{q}_{1:T}|D\right) = \int p\left(\mathbf{q}_{1:T}|\bar{\boldsymbol{w}}\right) p\left(\bar{\boldsymbol{w}}|k^*, D\right) \mathrm{d}\bar{\boldsymbol{w}}. \tag{13}$$

Algorithms 1 and 2 provide a compact description of the proposed methods for training and inference, respectively.

Algorithm 1 Training

1) Parameterize demonstrated trajectories:
Find vector of weights $\bar{\boldsymbol{w}}$ for each trajectory, such that $\mathbf{q}_t \approx \boldsymbol{\psi}_t^T \bar{\boldsymbol{w}}$.

2) Find GMM in parameter space, using EM:
Initialize GMM parameters $\alpha_{1:K}$, $\boldsymbol{\mu}_{1:K}$ and $\mathbf{\Sigma}_{1:K}$ with k-means clustering.

repeat

E step

$$r_{ik} = p(k|\bar{\boldsymbol{w}}_i) = \frac{\mathcal{N}(\bar{\boldsymbol{w}}_i; \boldsymbol{\mu}_k, \mathbf{\Sigma}_k)\alpha_k}{\sum_{l=1}^{K} \alpha_l \mathcal{N}(\bar{\boldsymbol{w}}_i; \boldsymbol{\mu}_l, \mathbf{\Sigma}_l)}$$

M step

$$n_k = \sum_{i=1}^{n} r_{ik}, \ \ \alpha_k = \frac{n_k}{n}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n} r_{ik}\bar{\boldsymbol{w}}_i}{n_k}$$

$$\mathbf{\Sigma}_k = \frac{1}{n_k}\left(\sum_{i=1}^{n} r_{ik}(\bar{\boldsymbol{w}}_i - \boldsymbol{\mu}_k)(\bar{\boldsymbol{w}}_i - \boldsymbol{\mu}_k)^T\right)$$

until $p(\bar{\boldsymbol{w}}; \alpha_{1:K}, \boldsymbol{\mu}_{1:K}, \mathbf{\Sigma}_{1:K})$ converges

Algorithm 2 Inference

1) Find most probable cluster given observation:

$$p(k|D) \propto p(D|k)p(k)$$

$$k^* = \arg\max_k p(k|D)$$

2) Condition on observation, using cluster k^*:

$$p(\mathbf{q}_{1:T}|D) = \int p(\mathbf{q}_{1:T}|\bar{\boldsymbol{w}})p(\bar{\boldsymbol{w}}|k^*, D)\mathrm{d}\bar{\boldsymbol{w}}$$

IV. Experiments

This section presents experimental results in two different scenarios using a 7-DOF KUKA lightweight arm with a 5-finger hand[2].

The goal of the first scenario is to expose the issue of the original Interaction Primitives [1], [2] when dealing with trajectories that have a clear multimodal distribution. In the second scenario we propose a real application of our method where the robot assistant acts as a third hand of a worker assembling a toolbox (please, refer to the accompanying video[3]).

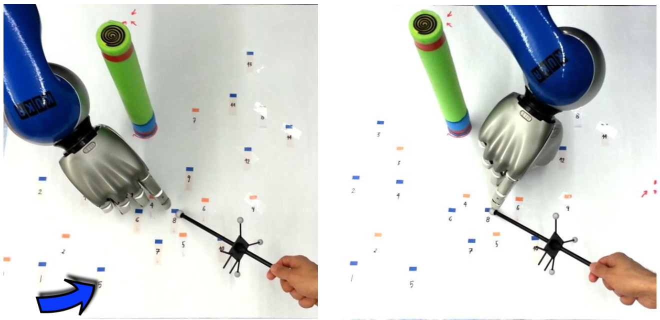
A. Nonlinear Correlations between the Human and the Robot on a Single Task

To expose the capability of our method for dealing with multimodal distributions, we propose a toy problem where a human specifies a position on a table and the robot must point at the same position. The robot is not provided any form of exteroceptive sensors; the only way it is capable of generating the appropriate pointing trajectory is by correlating its movement with the trajectories of the human. As shown in Fig. 2, however, we placed a pole in front of the robot such that the robot can only achieve the position specified by the human by moving either to the right or to the left of the pole. This scenario forces the robot to assume quite different configurations, depending on which side of the pole its arm is moving around.

During demonstrations, the robot was moved by kinesthetic teaching to point at the same positions indicated by the human (tracked by motion capture) without touching the pole. For certain positions, as the one indicated by the arrow in Fig. 2(a), only one demonstration was possible. For other positions, both right and left demonstrations could be provided as shown in Fig. 2(a) and 2(b). The demonstrations, totaling 28 pairs of human-robot trajectories, resulted in a multimodal distribution of right and left trajectory patterns moving around the pole.

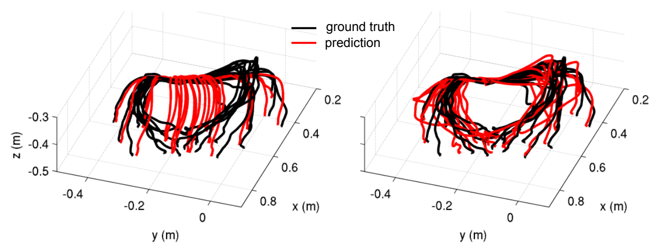
[2]Regarding the control of the robot, the design of a stochastic controller capable of reproducing the distribution of trajectories is also part of ProMPs and the interested reader is referred to [16] for details. Here we use a compliant, human-safe standard inverse-dynamics based feedback controller.

[3]Also available at http://youtu.be/9XwqW_V0bDw



(a) (b)

Fig. 2. Experimental setup of a toy problem used to illustrate the properties of the Mixture of Interaction Primitives. The robot is driven by kinesthetic teaching to point at the positions specified by the human (pointed with the wand). Certain pointed positions can be achieved by either moving the arm to the right (a) or to left (b) of the pole placed on the table. Other positions, such as the one indicated by the arrow, can only be achieved by one interaction pattern.



(a) (b)

Fig. 3. Results of the predictions of the robot trajectories in Cartesian space. Both subplots show the same ground truth trajectories generated by driving the robot in kinesthetic teaching. The predictions are generated by leave-one-out cross-validation on the whole data set comprised of 28 demonstrations. (a) Prediction using the conventional Interaction ProMPs with a single Gaussian. (b) Prediction using the proposed method with a mixture of Gaussians.

In this scenario, modeling the whole distribution over the parameters of the trajectories with one single Gaussian (as in the original Interaction Primitive formulation) is not capable of generalizing the movements of the robot to other positions in a way that resembles the training, as the original framework is limited by assuming a single pattern. This limitation is clearly shown in Fig. 3(a), where several trajectories generated by a single cluster GMM (as in the original Interaction Primitive) cross over the middle of the demonstrated trajectories, which, in fact, represents the mean of the single Gaussian distribution.

Fig. 3(b) shows the predictions using the proposed method with a mixture of Gaussians. By modeling the distribution over the parameters of the trajectories using GMMs as described in section III-C, a much better performance could be achieved. The GMM assumption that the parameters are only locally linear correlated seemed to represent the data much more accurately. As shown in Fig. 4, this improvement is quantified in terms of the Root Mean Square (RMS) error of the prediction of the trajectory in relation to the ground truth using leave-one-out cross-validation over the whole data set. The same figure also shows that there is a sharp decrease in the RMS error up to six clusters, especially when taking into account the variance among the 28 tests. Beyond seven clusters it is observed that the prediction error fluctuates around 4 cm. The experiments previously shown in Fig. 3(b) were done with eight clusters.

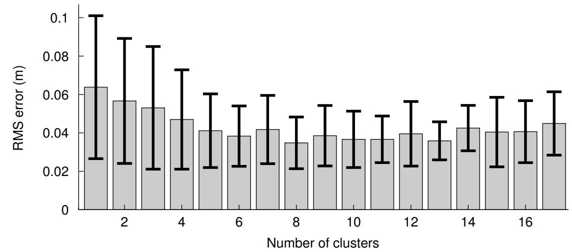


Fig. 4. Root Mean Square error with models using up to 17 Gaussians.

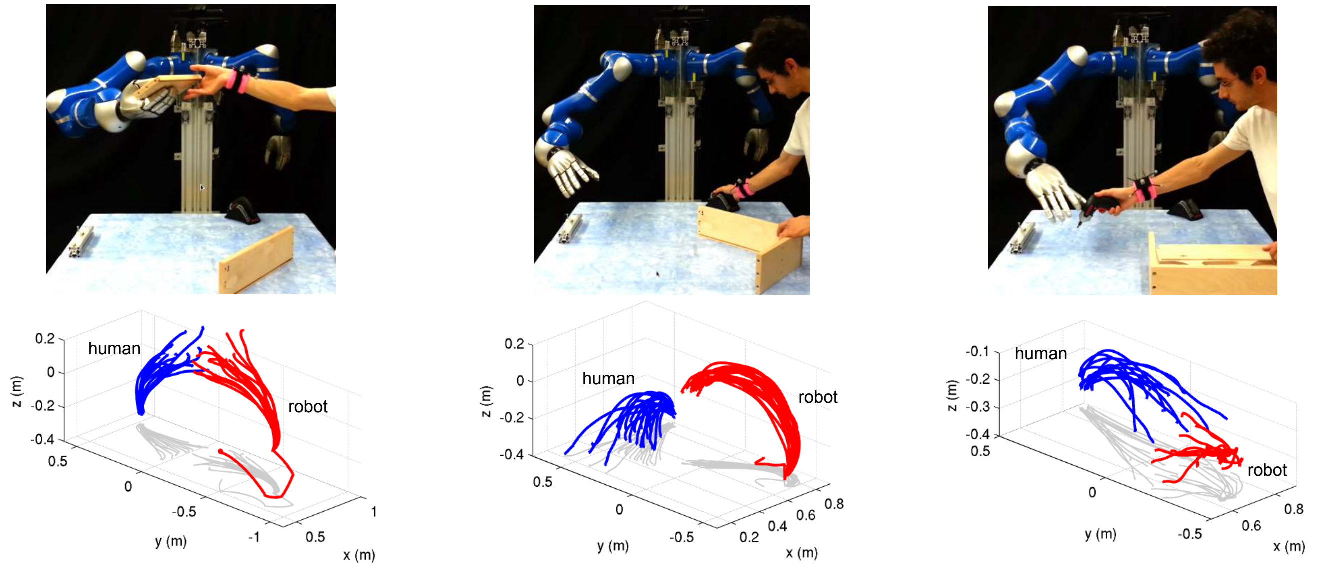
B. Assembling a Box with a Robot Assistant

In this experiment, we recorded a number of demonstrations of different interaction patterns between a human and the robot cooperating to assemble a box. We used the same robot described in the previous experiment. During demonstrations, the human wore a bracelet with markers whose trajectories in Cartesian coordinates were recorded by motion capture. Similarly to the first scenario, the robot was moved in gravity compensation mode by another human during the training phase and the trajectories of the robot in joint space were recorded.

There are three interaction patterns. Each interaction pattern was demonstrated several times to reveal the variance of the movements. In one of them, the human extends his/her hand to receive a plate. The robot fetches a plate from a stand and gives it to the human. In a second interaction, the human fetches the screwdriver, the robot grasps and gives a screw to the human as a pre-emptive collaborator would do. The third type of interaction consists of giving/receiving a screwdriver. Each interaction of plate handover, screw handover and holding the screwdriver was demonstrated 15, 20, and 13 times, respectively. The pairs of trajectories of each interaction are shown in Fig. 5[4].

As described in section III, all training data are fed to the algorithm resulting in 48 human-robot pairs of unlabeled demonstrations as shown in the upper row of Fig. 7. The presented method parameterizes the trajectories and performs clustering in the parameter space in order to encode the mixture of primitives. In the upper row of Fig. 7, each mixture is represented by a different color. The human is represented by the (x, y, z) Cartesian coordinates while the robot is represented by the seven joints of the arm. The figure shows the first four joints of the robot (starting from the base).

[4]Due to the experimental setup, for the sub-tasks of plate and screw handover we added an initial hand-coded trajectory that runs before the kinesthetic teaching effectively starts. These trajectories are used to make the robot grasp and remove the plate or screw from their respective stands. This is reflected in the figure as the deterministic part at the beginning of the trajectory of the robot. This initial trajectory, however, has no effect on the proposed method itself.



(a) Handing over a plate

(b) Handing over a screw

(c) Holding the screw driver

Fig. 5. Demonstrations of the three different interactions and their respective trajectories. For the case of plate and screw handover the beginning of the robot trajectory shows a deterministic part that accounts for the fact that the robot has to remove objects from their respective stands, which is not part of the kinesthetic teaching and does not affect the algorithm in any sense.

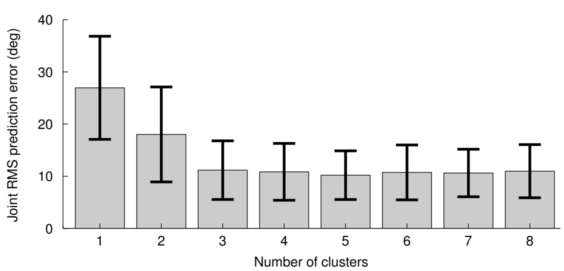


Fig. 6. Root Mean Square error of the joint trajectories (averaged over all tests) using a leave-one-out cross-validation as a function of the number of clusters (mixture components). The plateau after three clusters seems to be consistent with the training data set, since it consists of three distinct interaction patterns.

Figure 6 shows the RMS prediction error averaged over all tests as the number of mixture components increase. The prediction is obtained by leave-one-out cross-validation over the whole set of 48 demonstrations. As one would expect, since the unlabeled data set contains three distinct interaction patterns, the improvement is clearly visible up to three mixture components. No significant improvement is obtained afterwards, thus the GMM with three mixture components was selected for experiments.

In the inference/execution phase, the algorithm first computes the most probable Interaction Primitive mixture component based on the observation of the position of the wrist of the human with (12). Using the same observation, we then condition the most probable Interaction Primitive, which allows computing a posterior distribution over trajectories for all seven joints of the robot arm as in (13). Finally, the mean of each joint posterior distribution is fed to a standard inverse dynamics feedback tracking controller.

The lower row of Fig. 7 depicts the posterior distribution for one test example where a three-cluster GMM was trained with the other 47 trajectories. The GMM prior is shown in gray where the patches of different clusters overlap. The observation consists only of the final position of the wrist, shown as asterisks in the figure. The black lines are the ground truth trajectories of each degree of freedom. The posterior, in red, is represented by its mean and by the region inside $\pm$ two standard deviations. The mean of this posterior is the most probable trajectory for each degree of freedom given the observed end position of the wrist of the human.

We assembled the toolbox, consisting of seven parts and 12 screws, two times. The experiments demanded more than 40 executions of the Interaction Primitives. The selection of the right mixture component was 100% correct. (Please refer to the accompanying video).

We evaluated the precision of the interactions by computing the final position of the hand of the robot with forward kinematics. The forward kinematics was fed with the conditioned robot trajectories predicted by leave-one-out cross validation. The interactions of plate handover and holding screwdriver resulted in mean error with two standard deviations (mean error $\pm 2\sigma$) of 3.2 ± 2.6 cm and 2.1 ± 2.3 cm, respectively. We did not evaluate the precision of the handover of the screw, as the position at which the robot hands the screw is not correlated with the human (please refer to the accompanying video). As an example, Fig. 8 shows the robot executing the plate handover at three different positions based on the location of the wrist marker. Note that the postures of the arm are very different, although they are all captured by the same Interaction Primitive.

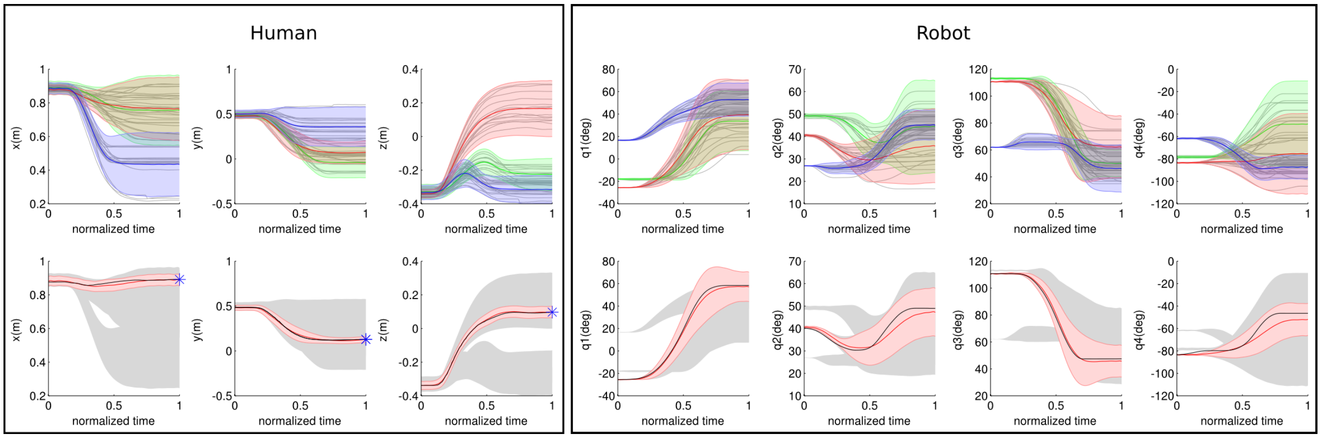


Fig. 7. **Upper row**: Mixture components represented by their mean trajectories and the region inside two standard deviations ($\mu \pm 2\sigma$). Each mixture component is represented by a different color and corresponds to a different interaction pattern. The light gray trajectories are the training trajectories. Obs.: The plots show only the part of the trajectories generated by kinesthetic teaching. **Lower row**: Posterior probability distribution (red) given observation depicted by the blue asterisks. The GMM prior is shown in gray.

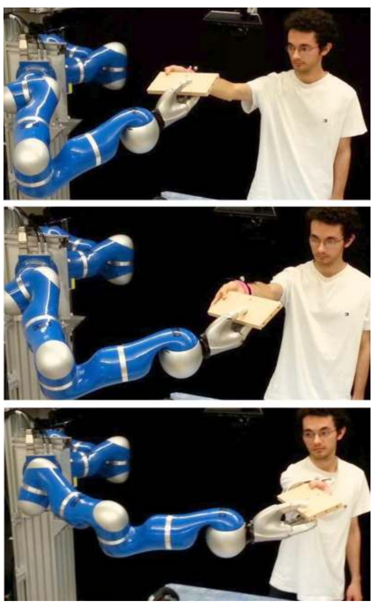


Fig. 8. Handover of a plate. Conditioning on three different positions of the wrist (using motion capture) of a human coworker.

V. Conclusions

In this paper we presented a Mixture of Interaction Primitives where Gaussian Mixture Models are used to model multiple interaction patterns from unlabeled data. The multimodal prior probability distribution is obtained over parameterized demonstration trajectories of two agents working in collaboration. During the execution, the algorithm selects the mixture component with the highest probability given the observation of the human, which is then conditioned to infer the appropriate robot reaction. The proposed method is able to learn and recognize multiple human-robot collaboration tasks from an arbitrary number of demonstrations consisting of unlabeled interaction patterns, what was not possible with the previous Interaction Primitive framework.

In the context of human-robot interaction we are currently addressing the estimation of the phase of the execution of the primitive for switching tasks in real time. Also, we are addressing the use of the stochastic feedback controller provided by the original ProMP work in [16]. Although this work focused on human-robot trajectories, we are currently considering extensions of our work where the human is replaced by other variables of interest. For example, the same framework can be used to correlate joint and end-effector trajectories of the same robot to learn nonlinear forward/inverse kinematic models. Similarly the Mixture of Interaction Primitives can be used to correlate the interaction between motor commands and joint trajectories to learn inverse dynamics models.

VI. Acknowledgments

The research leading to these results has received funding from the project BIMROB of the "Forum für interdisziplinäre Forschung" (FiF) of the TU Darmstadt, from the European Community's Seventh Framework Programme (FP7-ICT-2013-10) under grant agreement 610878 (3rdHand) and from the European Community's Seventh Framework Programme (FP7-ICT-2009-6) under grant agreement 270327 (ComPLACS).

References

[1] H. Ben Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *Proceedings of 2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[2] G. Maeda, M. Ewerton, R. Lioutikov, H. Ben Amor, J. Peters, and G. Neumann, "Learning interaction for collaborative tasks with probabilistic movement primitives," in *Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)*, 2014.

[3] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning collaborative impedance-based robot behaviors," in *AAAI Conference on Artificial Intelligence*, Bellevue, Washington, USA, 2013.

[4] M. Lawitzky, J. Medina, D. Lee, and S. Hirche, "Feedback motion planning and learning from demonstration in physical robotic assistance: differences and synergies," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 3646–3652.

[5] T. Kulvicius, M. Biehl, M. J. Aein, M. Tamosiunaite, and F. Wörgötter, "Interaction learning for dynamic movement primitives used in cooperative robotic tasks," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1450–1459, 2013.

[6] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, ser. CVPR '97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 994–.

[7] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 831–843, Aug 2000.

[8] K. Hawkins, N. Vo, S. Bansal, and A. F. Bobic, "Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration," in *Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)*, 2013.

[9] Y. Tanaka, J. Kinugawa, Y. Sugahara, and K. Kosuge, "Motion planning with worker's trajectory prediction for assembly task partner robot," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 1525–1532.

[10] Z. Wang, K. Mülling, M. P. Deisenroth, H. Ben Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic movement modeling for intention inference in human–robot interaction," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 841–858, 2013.

[11] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response." in *Robotics: Science and Systems*, 2013.

[12] D. Lee, C. Ott, Y. Nakamura, and G. Hirzinger, "Physical human robot interaction in imitation learning," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3439–3440.

[13] H. Ben Amor, D. Vogt, M. Ewerton, E. Berger, B. Jung, and J. Peters, "Learning responsive robot behavior by imitation," in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 3257–3264.

[14] B. Llorens-Bonilla and H. H. Asada, "A robot on the shoulder: Coordinated human-wearable robot control using coloured petri nets and partial least squares predictions," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation*, 2014.

[15] 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.

[16] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 2616–2624.

[17] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.