2014 14th IEEE-RAS International Conference on
Humanoid Robots (Humanoids)
November 18-20, 2014. Madrid, Spain

Learning Interaction for Collaborative Tasks with Probabilistic Movement Primitives

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

Abstract— **This paper proposes a probabilistic framework based on movement primitives for robots that work in collaboration with a human coworker. Since the human coworker can execute a variety of unforeseen tasks a requirement of our system is that the robot assistant must be able to adapt and learn new skills on-demand, without the need of an expert programmer. Thus, this paper leverages on the framework of imitation learning and its application to human-robot interaction using the concept of Interaction Primitives (IPs). We introduce the use of Probabilistic Movement Primitives (ProMPs) to devise an interaction method that both recognizes the action of a human and generates the appropriate movement primitive of the robot assistant. We evaluate our method on experiments using a lightweight arm interacting with a human partner and also using motion capture trajectories of two humans assembling a box. The advantages of ProMPs in relation to the original formulation for interaction are exposed and compared.**

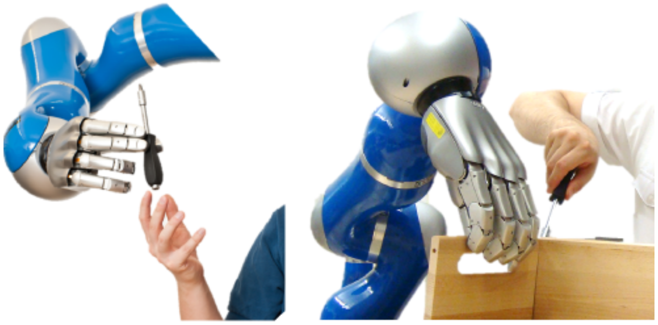


Fig. 1. Illustration of two collaborative tasks where a semi-autonomous robot helps a worker assembling a box. The robot must predict what is the action to execute, to hand over the screw driver or to hold the box. Its movement must also be coordinated relative to the location at which the human worker executes the task.

I. Introduction

While the traditional use of robots is to *replace* humans in dangerous and repetitive tasks we motivate this paper by semi-autonomous robots that *assist* humans. Semi-autonomous robots have the ability to physically interact with the human in order to achieve a task in a collaborative manner. The assembly of products in factories, the aiding of the elderly at home, the control of actuated prosthetics, and the shared control in tele-operated repetitive processes are just a few examples of application.

Only recently, physical human-robot interaction became possible due advances in robot design and safe, compliant control. As a consequence, algorithms for collaborative robots are still in the early stages of development. Assistance poses a variety of challenges related to the human presence. For example, Fig. 1 illustrates a robot assistant that helps a human to assemble a box. The robot must not only predict what is the most probable action to be executed based on the observations of the worker (to hand over a screw driver or to hold the box) but also the robot movement must be coordinated with the worker movement. Pre-programming a robot for all possible tasks that a worker may eventually need assistance with is unfeasible. A robot assistant must be able to learn the interaction and to adapt to a variety of unforeseen tasks without the need of an expert programmer.

Motivated by the described scenario, this work proposes the use of imitation learning [1] in the context of collaboration. Imitation learning has been widely used as a method to overcome the expensive programming of autonomous robots. Only recently, however, its application for physical interaction has been introduced under the concept of Interaction Primitives (IP) by Lee at al. in [2], defined as skills that allow robots to engage in collaborative activities with a human partner by Ben Amor et al. in [3].

Leveraging on the framework of [3], our approach is based on probabilistically modeling the interaction using a distribution of observed trajectories. We propose using Probabilistic Movement Primitives (ProMPs) [4] for modeling such a distribution. In a manufacturing scenario such a distribution of trajectories can be obtained by observing how two coworkers assemble a product, several times throughout the day, providing a rich data set for imitation learning. Such a collection of trajectories is used to create a prior model of the interaction in a lower dimensional weight space. The model is then used to recognize the intention of the observed agent and to generate the movement primitive of the unobserved agent given the same observations. The movement primitive of the unobserved agent can then be used to control a robot assistant.

The main contribution of this paper is the introduction of the Probabilistic Movement Primitives [4] in the context of imitation learning for human-robot interaction and action

[1]Intelligent Autonomous Systems Lab, Technische Universitaet Darmstadt, 64289 Darmstadt Germany. Correspondence should be addressed to maeda@ias.tu-darmstadt.de

[2]Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332-0280, USA.

[3]Max Planck Institute for Intelligent Systems Spemannstr. 38, 72076 Tuebingen, Germany

978-1-4799-7173-2/14/$31.00 ©2014 IEEE

recognition. We will show how Interaction ProMPs can be applied to address the three main problems previously illustrated in Fig. 1, that is: (a) learning a collaborative model by imitation learning and thus avoiding expert programming, (b) the ability to recognize a task by observing the worker, and (c) the coordination of the assistant movement in relation to the worker movement. We also show the advantages of ProMPs over the original DMP-based framework [3], and present an algorithm for aligning data using local optimization in order to avoid the issue of slope constraints typical of dynamic time warping.

II. Related Work

The data-driven analysis and classification of interactions between multiple persons has long been addressed within the computer vision community. In particular visual surveillance tasks, e.g., tracking of multiple pedestrians, require methods for identifying the occurrence and type of person-to-person interactions. In a seminal paper, Oliver et al. [5] show that hidden Markov models (HMMs), and more generally graphical models, are suited for representing the mutual dependencies of the behaviors between interacting agents. Graphical models have gained popularity in the field of human-robot interaction as they naturally include temporal information into the inference process and the Bayesian semantics provides a simple way to encode prior knowledge. In [6], Lee et al. use a hierarchical HMM to learn and represent responsive robot behaviors. In their approach, a high-level HMM identifies the current state of the interaction and triggers low-level HMMs which correspond to the robot's motor primitives. In order to ensure that the robot adapts to the movement of the human partner, virtual springs are attached between markers on the human body and corresponding positions on the robot. In a similar vein, Ben Amor et al. [7] use a path-map HMM to model interactions in cooperative tasks. In their approach, a backbone of shared hidden states correlates the actions of the interacting agents.

Tanaka et al. [8] use a Markov model to predict the positions of a worker in an assembly line. The space in which the worker moves is discretized into different regions. A Gaussian mixture model relates positions to procedures. Using this information a robot, then, delivers tools and parts to a human worker along the assembly line. Besides HMMs, other probabilistic graphical models have also been used to address interaction tasks. Koppula et al. [9] 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. Wang et al. [10] propose the intention-driven dynamics model, which models human intentions as latent states in graphical model. Intentions can be modeled as discrete variables, e.g., action labels, or continuous variables, e.g., an object's final position. The transitions between latent states and the mapping from latent states to observations are modeled via Gaussian Processes. As evidenced by these works, graphical models can be very powerful tools in classifying interactions. However, this often requires a substantial set of training data. In particular for humanoid motion generation with many degrees-of-freedom, it is often challenging to acquire sufficiently large and general data sets.

For more efficient learning and generalization, various authors investigated the projection of the original trajectories into a new, low-dimensional space where correlations between the agents are easier to unravel. Llorens et al. [11] show how such a low-dimensional interaction space can be used to implement an assistive robot arm. Similarly in [7], probabilistic principal component analysis is used to find a shared latent space. Dynamic Movement Primitives (DMPs) [12] allows for a low-dimensional, adaptive representation of a trajectory. The general idea is to encode a recorded trajectory as dynamical systems, which can be used to generate different variations of the original movement. In the context of interaction, Prada et al. [13] present a modified version of DMPs, that adapts the trajectory of one agent to a time-varying goal. By setting the goal to the wrist of another agent, the method can be used to generate handover motions.

Although graphical models and HMMs have been successfully used for action and intention recognition in a discretized symbolic level, the generation of trajectories for the continuous dynamic control of the robot is usually addressed by a different level of representation (e.g. a lower-level HMM [6] or DMPs). In relation to the previously cited works, here, we propose a framework based solely on a continuous movement representation that is used to both recognize actions and generate trajectories in the form of movement primitives; mainly leveraging on DMP-based Interaction Primitives [3] and Probabilistic Movement Primitives (ProMPs) [4]. By using ProMPs rather than DMPs our prosed method naturally correlates different agents directly in the same space in which observations are made, since observations of a task are usually given by their trajectories. This is an advantage in relation to the original framework of [3] since the representation of collaboration in the space of accelerations/forces due to the use of DMPs obfuscates the algorithm and increases its sensitivity to noise in the observations.

III. Proposed Method

This section briefly introduces Probabilistic Movement Primitives for a single degree of freedom as presented in [4] and proposes its extension for interaction and collaboration. Although not covered in this work, in its original proposition, the design of a feedback controller that tracks the distribution of trajectories is also part of ProMPs and the interested reader is referred to [4] for details; here we assume the existence of a human-safe standard feedback controller such as a low-gain PD controller. This section also exposes the main characteristics of the interaction framework based on DMPs in [3] and its relation to the approach of this paper. Finally, a simple local optimization algorithm is proposed for aligning several demonstrations provided by a human.

A. ProMPs for a Single DOF

For the purposes of the following derivations we generically refer to each joint or Cartesian coordinates of a human or robot simply as a degree of freedom (DOF) with position q and velocity $\dot{q}$. Starting with the case of a single DOF, we denote $\boldsymbol{y}(t) = [q(t)\ \dot{q}(t)]^T$ and a trajectory as a sequence $\boldsymbol{\tau} = \{\boldsymbol{y}(t)\}_{t=0,...T}$. We adopt linear regression with n Gaussian basis functions $\boldsymbol{\psi}$. The state vector $\boldsymbol{y}(t)$ can then be represented by a n-dimensional column vector of weights $\boldsymbol{w}$ as

$$\boldsymbol{y}(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}(t) \\ \dot{\boldsymbol{\psi}}(t) \end{bmatrix} \boldsymbol{w} + \boldsymbol{\epsilon}_y, \tag{1}$$

where $\boldsymbol{\Psi}_t = [\boldsymbol{\psi}(t),\ \dot{\boldsymbol{\psi}}(t)]^T$ is a 2×n dimensional time-dependent basis matrix and $\boldsymbol{\epsilon}_y \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_y)$ is zero-mean i.i.d. Gaussian noise. The probability of observing the whole trajectory is then

$$p(\boldsymbol{\tau}|\boldsymbol{w}) = \prod_{0}^{T} \mathcal{N}(\boldsymbol{y}(t)|\boldsymbol{\Psi}_t \boldsymbol{w}, \boldsymbol{\Sigma}_y). \tag{2}$$

Similar to DMPs the speed of the execution of the movement is decoupled from the speed of the original trajectory by using a phase variable $z(t)$. The phase variable replaces the time in order to control the location of the basis functions with $\boldsymbol{\psi}(z(t))$. For simplicity we will use $z(t) = t$ such that $\boldsymbol{\psi}(t) = \boldsymbol{\psi}(z(t))$ while remembering that any monotonically increasing function can be used [4].

Each trajectory is now represented by a low-dimensional vector $\boldsymbol{w}$ since the number of basis is usually much smaller than the number of time steps. Trajectory variations obtained by different demonstrations are captured by defining the distribution over the weights $p(\boldsymbol{w}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the learning parameter. The probability of the trajectory becomes

$$p(\boldsymbol{\tau}|\boldsymbol{\theta}) = \int p(\boldsymbol{\tau}|\boldsymbol{w}) p(\boldsymbol{w}|\boldsymbol{\theta}) d\boldsymbol{w}. \tag{3}$$

So far $\boldsymbol{\theta}$ captures the correlation among the weights within the trajectory and between demonstrations of the same DOF.

B. ProMPs for Collaboration

The key aspect for the realization of the interaction primitives is the introduction of a parameter $\boldsymbol{\theta}$ that captures the correlation of all DOFs of multiple agents. Assuming that the distribution of trajectories of different agents is normal, then $p(\boldsymbol{w}; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$. Under this assumption we redefine the vector of weights $\boldsymbol{w}$ to account for all degrees of freedom of multiple-agents. Following the definitions in [3] we will refer to the assisted human as the *observed agent*, and assume that he/she provides the observed DOFs of the model (e.g by motion capture). The robot will be referred to as the *controlled agent*.

For an observed agent with P DOFs and a controlled agent with Q DOFs, we construct a row weight vector by concatenating the trajectory weights

$$\bar{\boldsymbol{w}}_d = \{[\boldsymbol{w}_1^T, ...\boldsymbol{w}_p^T, ..., \boldsymbol{w}_P^T]^o,\ \ [\boldsymbol{w}_1^T, ...\boldsymbol{w}_q^T, ..., \boldsymbol{w}_Q^T]^c\} \tag{4}$$

where $\bar{\boldsymbol{w}}_d$ is the augmented weight vector corresponding to the d-th demonstration, $\boldsymbol{w}_p$ is the n-dimensional column vector of weights of the p-th DOF of the observed agent, and $\boldsymbol{w}_q$ is the vector of weights of the q-th DOF of the controlled agent. The mean and covariance are then computed by stacking all demonstration weights

$$\begin{aligned} \boldsymbol{\mu}_w &= \text{mean}([\bar{\boldsymbol{w}}_1, ..., \bar{\boldsymbol{w}}_d, , ..., \bar{\boldsymbol{w}}_D]^T), \\ \boldsymbol{\Sigma}_w &= \text{Cov}([\bar{\boldsymbol{w}}_1, ..., \bar{\boldsymbol{w}}_d, , ..., \bar{\boldsymbol{w}}_D]^T), \end{aligned} \tag{5}$$

where D is the number of demonstrations.

Gaussian conditioning can then be applied on-line as each new observation is made using recursive updates in the form

$$\begin{aligned} \boldsymbol{\mu}_w^+ &= \boldsymbol{\mu}_w^- + \boldsymbol{K}(\boldsymbol{y}^*(t) - \boldsymbol{H}_t^T \boldsymbol{\mu}_w^-) \\ \boldsymbol{\Sigma}_w^+ &= \boldsymbol{\Sigma}_w^- - \boldsymbol{K}(\boldsymbol{H}_t^T \boldsymbol{\Sigma}_w^-) \\ \boldsymbol{K} &= \boldsymbol{\Sigma}_w^- \boldsymbol{H}_t^T (\boldsymbol{\Sigma}_y^* + \boldsymbol{H}_t^T \boldsymbol{\Sigma}_w^+ \boldsymbol{H}_t)^{-1}, \end{aligned} \tag{6}$$

where $\boldsymbol{K}$ is the Kalman gain matrix, $\boldsymbol{y}^*(t)$ is the observed value at time t, $\boldsymbol{\Sigma}_y^*$ is the measurement noise, and the upperscripts $-$ and $+$ the values before and after the update. The observation matrix $\boldsymbol{H}_t$ is block diagonal and each diagonal entry contains the 2×n basis $[\boldsymbol{\psi}(t),\ \dot{\boldsymbol{\psi}}(t)]^T$ for each observed joint

$$\boldsymbol{H}_t = \begin{bmatrix} \boldsymbol{\Psi}_t & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{\Psi}_t \end{bmatrix} \tag{7}$$

In the collaboration case only measurements of the observed agent are provided. By maintaining consistency with definition (4) where the entries of the observed agent come before the controlled agent, the mean is then $\boldsymbol{\mu}_w = [\boldsymbol{\mu}_w^o\ \ \boldsymbol{\mu}_w^c]^T$ and the observation matrix $\boldsymbol{H}_t$ is partitioned as

$$\boldsymbol{H}_t = \left[\begin{array}{cc:cc} (\boldsymbol{\Psi}_t^o)_{(1,1)} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{\Psi}_t^o)_{(P,P)} & \boldsymbol{0} & \boldsymbol{0} \\ \hdashline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}_{(1,1)}^c & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}_{(Q,Q)}^c \end{array}\right] \tag{8}$$

where each zero entry is of 2×n dimension. Note that if only positions of the observed agent are provided $(\boldsymbol{\Psi}_t^o)_{(p,p)} = [\boldsymbol{\psi}(t),\ \boldsymbol{0}(t)]^T$.

In general, since (6) is a full state linear estimator, any partial combination of observations (for example when $\boldsymbol{y}^*$ only contains positions, or velocities, or a mixture of both) provides the optimal estimate of states $\boldsymbol{\mu}_w^+$ and their uncertainty $\boldsymbol{\Sigma}_w^+$.

C. Action Recognition for Primitive Activation

Here we use the ProMP framework in a multi-task scenario where each task is one encoded by one interaction primitive.

Consider a specific task $s \in \{1, .., K\}$ and assume that for each task an Interaction ProMP has been generated as it was proposed in section III-B. Using the recursive notation of (6), the upper script $(\cdot)^-$ refers to the parameters of the Interaction ProMP updated up to the previous observation, that is $\boldsymbol{\theta}_s^- = \{\boldsymbol{\mu}_w^-, \boldsymbol{\Sigma}_w^-\}_s$. The probability of the observation at a subsequent time t given the parameters $\boldsymbol{\theta}_s^-$ of one of the tasks is

$$p(\boldsymbol{y}^*(t); \boldsymbol{\theta}_s^-) = \int p(\boldsymbol{y}^*(t)|\boldsymbol{\Psi}_t \boldsymbol{w}, \boldsymbol{\Sigma}_y^*) p(\boldsymbol{w}|\boldsymbol{\theta}_s^-) d\boldsymbol{w} \quad (9)$$

$$= \mathcal{N}(\boldsymbol{y}^*(t)|\boldsymbol{\Psi}_t \boldsymbol{\mu}_w^-, \boldsymbol{\Psi}_t \boldsymbol{\Sigma}_w^- \boldsymbol{\Psi}_t + \boldsymbol{\Sigma}_y^*). \quad (10)$$

The task s can now be recognized by applying Bayes rule

$$p(s|\boldsymbol{y}^*(t)) = \frac{p(\boldsymbol{y}^*(t)|\boldsymbol{\theta}_s^-)p(s)}{\sum_{k=1}^K p(\boldsymbol{y}^*(t)|\boldsymbol{\theta}_k^-)p(k)}, \quad (11)$$

where $p(s)$ is the initial probability of the task (e.g. $p(s) = 1/K$ for uniform distribution). We will evaluate Eqs. (9)-(11) using real collaboration data in the experimental section of this paper.

D. Relation to Interaction DMPs

It is now straightforward to relate our proposed method with the previous interaction primitives based on DMPs [3]. The principal difference is that in the framework of interaction DMPs the weights are mapped from the forcing function $f(t)$ as opposed to the positions $q(t)$. Using the linear basis-function model

$$f(t) = \boldsymbol{\psi}(t)^T \boldsymbol{w}, \quad (12)$$

where $\boldsymbol{\psi}(t)$ are the normalized Gaussian basis functions. Similarly to the ProMP case a distribution of weights $p(\boldsymbol{w})$ is learned based on several demonstrations of a task.

For each DOF, the forcing function adds a nonlinear behavior on the movement which complements a linear and stable spring-damper system

$$\ddot{q} = [\alpha_y(\beta_y(g - q) - \dot{q}/\tau) + f(t)]\tau^2, \quad (13)$$

where g is the goal attractor, α_y, β_y are user-defined parameters that characterize the spring-damper behavior and τ controls the speed of execution. For details on DMPs please refer to [12] and references therein.

When using imitation learning a demonstration is executed and measurements are usually given in the form of positions, which must be differentiated twice such that the forcing function can be computed

$$f(t) = \ddot{q}/\tau^2 - \alpha_y(\beta_y(g - q) - \dot{q}/\tau). \quad (14)$$

Referring back to (6) the Gaussian conditioning is now based on the observation of forces or accelerations, that is $y^*(t) = f(\ddot{q}, (\cdot), t)^*$. As our evaluations will show, the fact that forces are usually computed using second derivatives of the position can be restrictive for applications with asynchronous or sparse measurements as the observed accelerations needed for conditioning are hard to obtain in this case. In contrast, in the ProMP framework, it is possible to directly condition on the observed quantities, i.e., the position of the agent.

E. Time Warping with Local Optimization

One issue of imitation learning for trajectories is that multiple demonstrations provided by humans are usually, sometimes severely, warped in time. Demonstrations must be unwarped or time-aligned before the distribution of the weights can be computed. Here we propose aligning trajectories by taking one of the demonstrations as a reference $\boldsymbol{y}_r$ and using local optimization of the time warping function with

$$\boldsymbol{t}_w^{j+1} = v_0^j + \boldsymbol{g}(\boldsymbol{v}^j)\boldsymbol{t}_w^j, \quad (15)$$

where $\boldsymbol{t}_w^j$ represents a vector containing the warped time of demonstration $\boldsymbol{y}_w$ at the j-th iteration of the optimization.

We propose $\boldsymbol{g}$ as a smooth, linear Gaussian-basis-model with P weights $\boldsymbol{v}^j = [v_1^j, ..., v_P^j]$ as the parameters to be optimized. The extra parameter v_0^j is used to shift the time which is useful when the reference and warped trajectories are, in fact, identical but start at different instants. The optimization is initialized with $v_0^j = 0$ and $\boldsymbol{t}_w^j = \boldsymbol{t}_r$ for $j = 1$. The parameters $\boldsymbol{v}^j$ are optimized with gradient descent to decrease the absolute cumulative distance between the reference and warped trajectories

$$\boldsymbol{v} = \arg\min_v \sum_{k=0}^K |\boldsymbol{y}_r(\boldsymbol{t}_r(k)) - \boldsymbol{y}_w(v_0^j + \boldsymbol{g}(\boldsymbol{v}^j)\boldsymbol{t}_w^j)|. \quad (16)$$

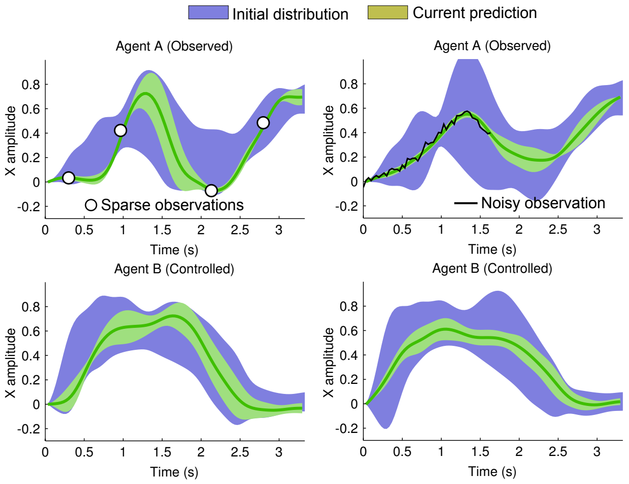
While Dynamic Time Warping (DTW) [14] is widely used for such problems, our local method forces alignment without "jumping" the indexes of the warped time vector which is an usual outcome of DTW and renders unrealistic and non-smooth trajectories. While this problem is usually minimized by imposing a slope constraint [14], the use of a smooth function $\boldsymbol{g}$ not only avoids the tunning of this parameter but also preserves the overall shape of the trajectory.

IV. Experiments

This section presents results on a simple simulated scenario to compare the differences between the original work of Interaction DMPs with Interaction ProMPs. Next, we evaluate the accuracy of Interaction ProMPs for generating reference trajectories for an anthropomorphic robot arm conditioned on the movement of a human. Finally, we will show experimental results with Interaction ProMPs used in a collaborative scenario of a box assembly to both recognize and predict the action of two collaborators.

A. Comparison with Interaction DMPs

In a typical interaction task the observations of a coworker might arrive asynchronously, at irregular periods of time, for example, when the measurement signal is prone to interruption (a typical case is occlusion in motion capture systems). Fig. 2 (a) illustrates a simple case where both observed and controlled agents have a single joint each. The training data was created by sketching two sets of trajectories on a PC screen using a computer mouse. We than use these two sets as proxies of the observed and controlled agents resulting on the initial distribution of trajectories (in blue). The upper



(a) (b)

Fig. 2. Two scenarios where the Interaction ProMPs are advantageous over Interaction DMPs. (a) Sparse and asynchronous observations. (b) Noisy stream of observed data ($\sigma^2 = 0.04$). The patches represent the ± 2σ deviations from the mean.

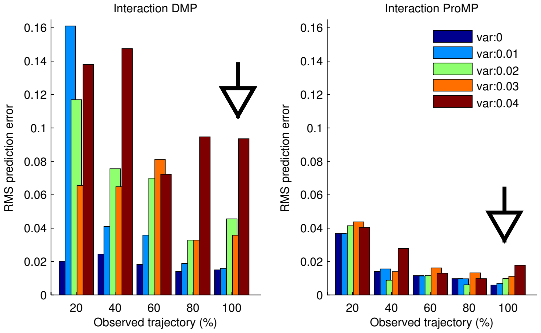
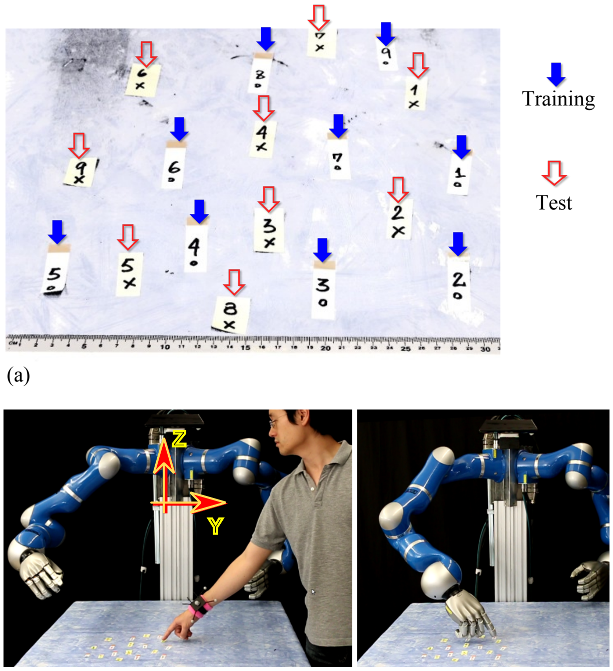


Fig. 3. Root-mean-square prediction error of the movement of collaborator B as a function of the number of observed samples of the trajectory of collaborator A. The different bars indicate the amount of noise added to the observation of the position of collaborator A.

plot shows the prediction (in green) of the observed agent after observing four measurements. Note that following (4) the predicted mean $\boldsymbol{\mu}_w^+$ has the dimension of the augmented weight vector, that is, if each single-DOF agent trajectory is encoded by n basis functions $\boldsymbol{\mu}_w^+$ is a column vector of size $2n$. The bottom figure depicts the predicted distribution of the controlled agent. Note that the same experiment can not be reproduced with Interaction DMPs as the second derivative on such sparse measurement is hard to compute and introduce innacuracies on the representation of the true force.

In Fig. 2 (b) the ProMP is being conditioned on a constant synchronous stream of noisy position measurements. The plot shows the case where the true trajectory is corrupted with a Gaussian noise with variance $\sigma^2 = 0.04$. Interaction DMPs suffer from noisy position measurements as the observation must be differentiated twice to compute the forcing function. While low-pass filters alleviate this problem, the introduction of phase lag is an issue that can be potentially avoided with ProMPs.



(a)

(b)

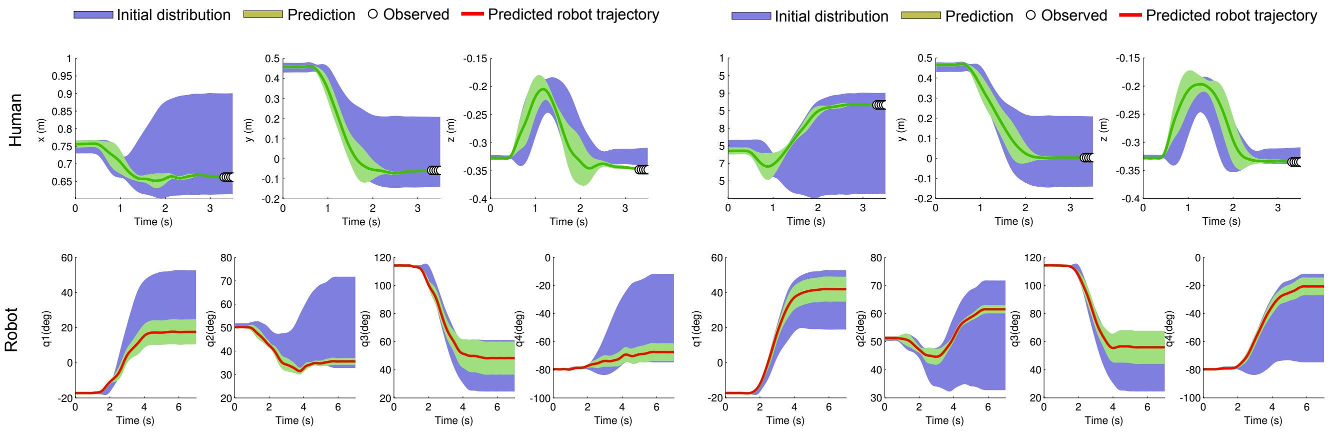
Fig. 4. An interactive task where the robot has to point at the same position previously pointed by the human. The robot, however, has no exteroceptive sensors and its predicted trajectory is based solely on the correlation with the observed human movement. (a) The nine positions used to create the Interaction ProMP (dot markers) and the extra nine positions used to test the method (cross markers). (b) An example where the human points at the test position #1 and the robot points to the same position.

Fig. 3 compares the prediction error over the whole trajectory of Interaction DMPs and ProMPs given the same noisy observed data. With DMPs the error is greatly influenced by the amount of noise while ProMPs show much less sensitivity. For the case where the full trajectory of collaborator A is observed (indicated by the arrow) the prediction error increased by a factor of five times using the Interaction DMPs when noise ranged from a clean signal to a signal of noise variance 0.04. In contrast, the error deteriorates by a factor of two with Interaction ProMPs.

B. Robot Control with Interaction ProMPs

We evaluated the ability of Interaction ProMPs in generating the appropriate movement primitive for controlling a robot based on observations of a human partner. The experiment consisted in measuring the $[x, y, z]$ trajectory coordinates of the wrist of an observed agent via motion capture[1] while pointing at a certain position on a table placed in front of our robot (a 7-DOF anthropomorphic arm with a 5-finger hand). Then, the robot was moved in gravity compensation mode to point with its index finger at the same position on the table while its joint positions were being recorded (kinesthetic teaching). This pair of

[1] All human positions were measured in relation to the world reference frame located at the torso of the robot (as shown in Fig. 4(b)).



(a)

(b)

Fig. 5. The upper row shows the human Cartesian coordinates of the wrist. The bottom row shows the first four joints of the 7-DOF robotic arm. (a) Conditioned results of the test position #6. (b) Conditioned results of the test position #8.

trajectories formed by the Cartesian positions of the wrist and the joint positions of the arm where then mapped into the weight space and concatenated as in Eq. (4). In total, nine different positions were pointed to collect training data, sparsely covering an approximate circular area of diameter 30 cm. The pointed positions are shown in Fig. 4(a) by the dots.

After creating the Interaction ProMPs, as described in Section III-B, we defined extra nine marked positions shown in Fig. 4(a) by the crosses. The human then pointed at one of the crosses while motion capture was used to measure the trajectory of the wrist. These observations were then used to condition the Interaction ProMP to predict trajectories for each joint of the robot, whose mean values where used as reference for a standard trajectory tracking inverse dynamics feedback controller with low gains.

Fig. 4(b) shows one example of the interactive task where the human pointed to the position marked by the cross #1, which was not part of the training; the robot was capable of pointing to the same position. Note that the robot was not provided with any exteroceptive feedback, such as cameras, to reveal the location of the pointed position. Although the robot was not directly "aware" of the position of the pointed cross, the interaction primitive provides the robot the capability to predict what movement to make based solely on the observed trajectories of the partner.

Figure 5 shows two examples on the conditioned interaction primitives when the human pointed at positions #6 in (a) and #8 in (b) (refer back to Fig. 4(a) for the physical position of the crosses). The first row in each subplot shows the $[x, y, z]$ coordinates of the wrist. The second row shows the first four joints of the robot, starting from the shoulder joint. Since we are only interested in the final pointing position, the interaction primitive was conditioned on the final measurements of the wrist position. As positions #6 and #8 were physically distant from each other, the difference between their predicted trajectories were quite large in relation to each other, roughly covering the whole span of the prior distribution (in blue) for some certain DOFs of the arm.

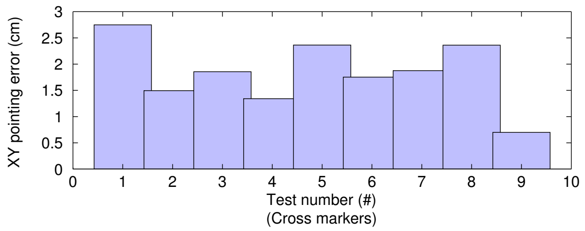


Fig. 6. Accuracy of the pointed positions by the robot when using the test positions given by the cross markers. The error was computed by taking the actual pointed position and the true position of the markers on the table.

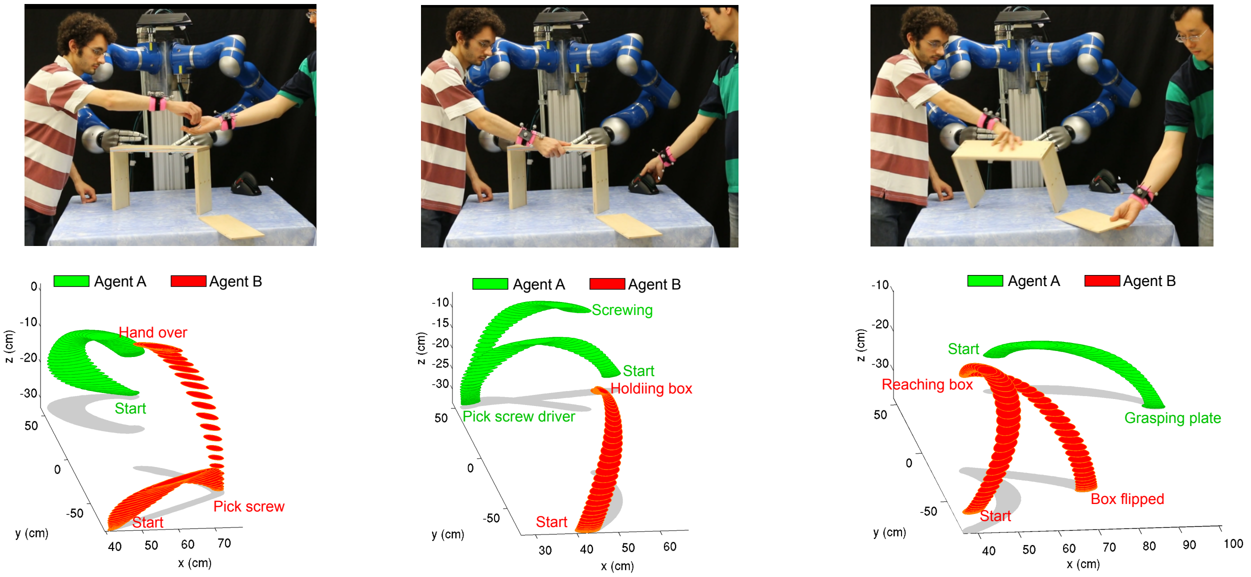
Figure 6 shows the distance error on the plane of the table between the position pointed by the robot and its true position. The robot was able to reasonably point even at locations at the limits of the training data such as position #1, #7, and #8 (see Fig. 4). The maximum error was of 3 cm, or 10% in relation to the total area covered by the training points (approximately a circle of diameter 30 cm). The experiments show that the physical movement of the robot is clearly conditioned by the position indicated by the human (see the accompanying video[2]).

Note that this not a precision positioning experiment; the markers on the wrist were not fixed in a rigid, repeatable manner, neither the finger of the robot could be positioned with milimetric precision during the kinesthetic teaching phase. The framework of Interaction ProMPs allows, however, to seamlessly integrate additional sensing to increase accuracy in precision tasks. This is naturally achieved adjusting the observation vector $\boldsymbol{y}^*$ in (6) and the zero entries in (8) to include new sensory information such as the reference position of a hole in which the robot must insert a peg.

C. Action Recognition for Primitive Activation

While in the previous experiments interaction primitives were evaluated for the case of a single task, here we show

[2] also available from http://youtu.be/2Ok6KQQQDNQ



(a) Hand over (b) Fastening (c) Plate insertion

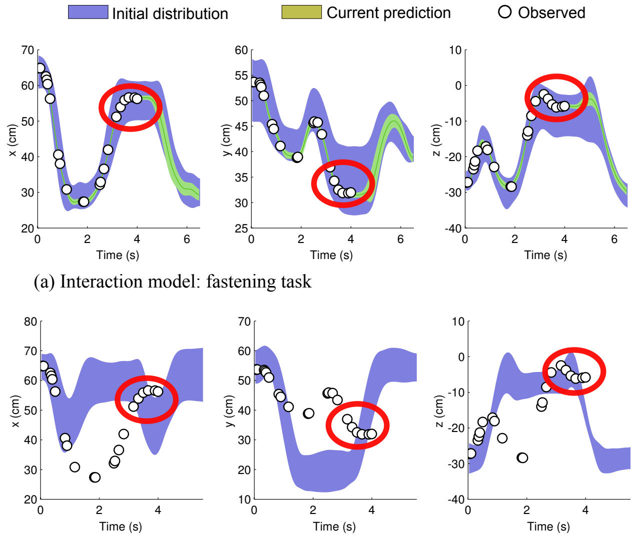
Fig. 7. Three collaborative tasks involved when assembling a box by two co-workers. From left to right, the photos show the hand over of a screw, the fastening of the screw where one agent grasps the screw driver while the other holds the box steadily, and the insertion of a plate, which requires one agent to flip the box such that the slot becomes accessible to the other agent. The distribution of aligned demonstrations for each task are shown under their respective photos. The plot shows the covariance in the x-y plane at each corresponding z height.

how Interaction ProMPs can be used for recognizing the action of the observed agent and to select the appropriate desired movement primitive of the controlled agent. This capability allows the robot to maintain a library of several tasks encoded as Interaction ProMPs and to activate the appropriate primitive based on the observation of the current task.

As shown in the photos of Fig. 7, we collected collaborative data in the form of the Cartesian coordinate positions of the wrists of two humans assembling a box. As in the previous experiment of section IV-B, all measurements were taken in relation to the torso of the robot. The collaborator on the right plays the role of the observed agent while the collaborator at the left plays the role of the controlled agent. The controlled agent will be referred to as the *predicted agent* since he/she can not be controlled.

In the ”hand-over” task shown in Fig. 7(a), the observed agent stretches his hand as a gesture to request a screw. The predicted agent then grasps a screw sitting on the table and hand it over to the collaborator. In the ”fastening” task shown in Fig. 7(b), the observed agent grasps an electrical screwdriver. The predicted agent reacts by holding the box firmly while the observed agent fastens the screw. In the ”plate insertion” task shown in Fig. 7(c), the observed agent grasps the bottom plate of the box. The predicted agent then flips the box such that the slots to which the plate slides in are directed towards the observed agent.

Each task is repeated 40 times. All trajectories are aligned using the method described in Section III-E. The aligned trajectories are shown in Fig. 7 under their respective photos as three-dimensional plots for each of the tasks where the covariance in x-y directions are shown at the corresponding heights (Z direction) of the movement. Interaction ProMPs are created for each task using the distribution of aligned trajectories.



(a) Interaction model: fastening task

(b) Interaction model: hand over task

Fig. 8. Action recognition based on conditioning the movement primitives of the observed agent. In this example the observations of the fastening task also overlaps with the primitives of the hand over task.

We evaluated action recognition using Eqs. (9)-(11) on the three presented tasks for box assembly. Fig. 8 shows one evaluation as an example. Note from the figure that the majority of observations indicate that the fastening task is taking place. The last five observations (surrounded by the ellipse), however, fits both tasks and could be a potential source of ambiguity in task recognition. Even in those

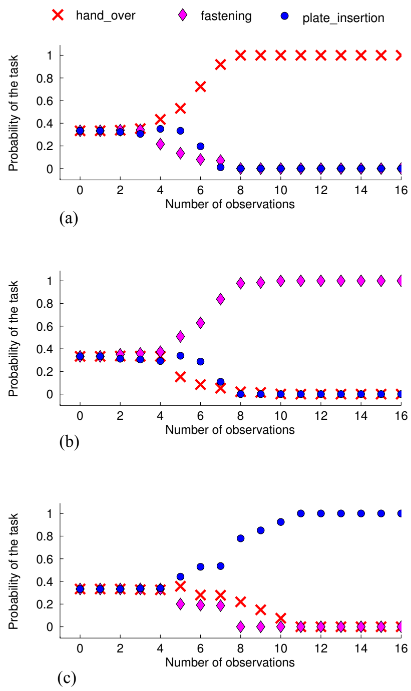


Fig. 9. Action recognition given three different Interaction ProMPs, one for each task involved in assembling the box. The three Interaction ProMPs are conditioned on the same observations of the observed agent. Probabilities of tasks are shown as a function of the number of observations along the trajectory of the observed agent. (a) Recognition of the hand over task. (b) Recognition of the fastening task. (c) Recognition of the plate insertion task.

cases, ProMPs can clearly distinguish among tasks as shown by the plots in Fig. 9 where the probabilities of the task are given as a function of the number of observations. Subplots (a), (b) and (c) show the task recognition for the fastening, hand over and plant insertion tasks, respectively. In general, we observed that 3-5 observations are required to achieve a 100% certainty for each task. (The last part of the accompanying video shows our method controlling the robot assistant to assembly a box with recognition of two different handover tasks).

V. Conclusion

This paper introduced a method for collaboration suited for new applications using semi-autonomous robots whose movements must be coordinated with the movements of a human partner. By leveraging on the original framework of Interaction Primitives [3] we proposed the use of ProMPs for the realization of primitives that capture the correlation between trajectories of multiple agents. This work compared the main differences between DMPs and ProMPs for interaction and advocates the later for applications where measurements are noisy and/or prone to interruption. Using a 7-DOF lightweight arm we evaluated the capability of Interaction ProMPs in generating the appropriate primitive for controlling the robot in an interactive task involving a human partner. We also proposed a method for task recognition that naturally fits the ProMP framework.

Our current work addresses the use of mixture-models to automatically generate different Interaction ProMPs without a priori hand labeling of multiple tasks. We are also investigating tasks in which some of the involved DOFs do not correlate linearly and also when certain tasks do not induce correlation. The later is especially true for tasks where the movement of the agents are not related by causality.

VI. Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programmes (FP7-ICT-2013-10) under grant agreement 610878 (3rdHand) and (FP7-ICT-2009-6) under grant agreement 270327 (ComPLACS). The authors would like to acknowledge Filipe Veiga, Tucker Hermans and Serena Ivaldi for their assistance during the preparation of this manuscript.

References

[1] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[2] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in humanhumanoid interaction," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1684–1704, 2010.

[3] 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.

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

[5] 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.

[6] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in human-humanoid interaction," *Int. Journal of Robotics Research.*, vol. 29, no. 13, pp. 1684–1704, Nov. 2010.

[7] 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.

[8] 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.

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

[10] Z. Wang, K. Mülling, M. P. Deisenroth, H. B. 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] 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.

[12] 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.

[13] M. Prada, A. Remazeilles, A. Koene, and S. Endo, "Dynamic movement primitives for human-robot interaction: comparison with human behavioral observation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1168–1175.

[14] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43–49, 1978.